Abstract—Convolutional neural network (CNN) is widely used in hyperspectral image (HSI) classification owing to their advantages of spatial-spectral features capture capability and learning depth features as well as their structural flexibility. Nevertheless, the shape of the convolution kernel is fixed, a limitation that leads to shape fixation when modeling different features in CNN, especially in the edge regions between classes. A multiscale adaptive convolution (MSAC) model is proposed in this article to overcome this shortcoming. Combining superpixels with the traditional convolutional kernels to form adaptive kernels automatically adjusts the receptive field, suppresses edge noise, and enhances feature learning for different classes. On the basis of adaptive convolution, adaptive convolution units (AConvUs) are constructed. The hierarchical residual structure is constructed by superimposing multiple AConvUs to learn the spatial spectral features of different receptive fields of the HSI, reduce the gradient disappearance and enhance the robustness. The proposed multiscale convolution adjusts the shape of the convolution kernel according to the spatial distribution of different superpixels in HSI. Finally, the MSAC classification framework is constructed by the decision fusion of multiscale adaptive convolution at different superpixel scales, which helps to extract the complementary information of the HSI. The MSAC method’s experimental performance on several HSI datasets, including Indian Pines, University of Pavia, Salinas and Gaofen-5, to verify the validity and practically.

Index Terms—Adaptive convolution kernel (AConv), convolution neural network (CNN), hierarchical residual network, hyperspectral image (HSI) classification, superpixel segmentation.

Manuscript received 13 March 2022; revised 4 May 2022 and 21 May 2022; accepted 15 June 2022. Date of publication 21 June 2022; date of current version 4 July 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61977022, in part by the Science Foundation for Distinguished Young Scholars of Hunan Province under Grant 2020J2017, in part by the Key Research and Development Program of Hunan Province under Grant 2019SK2012, in part by the Foundation of Department of Water Resources of Hunan Province under Grant XSKJ2021000-12 and Grant XSKJ2021000-13, in part by the Natural Science Foundation of Hunan Province under Grant 2022JJ40226, and in part by the Foundation of Education Bureau of Hunan Province under Grant 21B0590, Grant 21B0595 and Grant 20B062.

I. INTRODUCTION

HYPERSONTICAL images (HSIs) is a 3-D that contains spatial features and continuous spectral information. Each pixel contains thousands of continuous bands, providing rich spectral signatures [1]. Therefore, HSI data containing a large amount of information have been successfully applied in environment monitoring [2], [3], medical treatment [4], [5], agricultural evaluation [6], and geological exploration [7]. These applications are premised on the precise classification of each pixel in the HSI.

At present, various classification methods have been designed for HSI data processing. Most HSI data processing methods focus on discriminant spatial, spectral, or joint spectral-spatial feature extraction. Most of the previous approaches consider the separation of spectra in higher dimensions. Consequently, determining the establishment of mapping functions and finding separable hyperplanes become the goal of the investigation. Extreme learning machine [8] and support vector machine (SVM) [9] are widely used pixel-by-pixel HSI classifiers for HSI classification. Yet, the final classification maps obtained by pixel-by-pixel methods are subject to interference noise as well as misclassified regions. For addressing this problem, researchers have proposed kernel methods to enhance distinguish ability using kernel support vector machine[10] and multiple kernel learning [11]. However, kernel methods generally concern only the design of classifiers and ignore feature representation and feature learning. Therefore, conventional feature extraction method [12], such as principal component analysis (PCA) [13] and manifold learning [14]–[16], can make full use of spectral information to reveal the inherent spectral features of HSIs. In addition, other methods such as superpixel segmentation [17] and morphological segmentation [18], investigate the spatial structures of HSIs to facilitate different types of spectral-spatial feature learning. Explicit modeling of the spatial structure of an HSI could enable the better use of its spatial information [19], [20]. Due to the difficulty of the manually extracted features and the inadequate parameter settings, the above methods fail to learn deep features with robustness from HSIs.

As an example, deep learning such as convolutional neural networks (CNNs) [21], long short-term memory [22], and gated recurrent unit (GRU) [23], can automatically learn deep features in HSIs from training samples with satisfactory results compared to conventional methods. CNNs are widely used as tools to extract deep spectral-spatial features from HSIs because of...
their superior performance. Besides, researchers have proposed various variants from 1D-CNN [24] to 3D-CNN [25], thus improving the capability of spectral-spatial features. For example, Li et al. [26] proposed a dual-flow CNN approach for deep feature fusion of HSI and extracting spectral, local spatial, and global spatial features of HSI with a limited number of training samples. In addition, Wang et al. [27] proposed an adaptive end-to-end spectral-spatial multiscale network to address that the final features extracted by common deep learning-based methods are always at a single scale for HSI classification. However, the above experiments also show that simply adding additional branches is insufficient for the obtained HSI classification. Thus, Zhong et al. [28] introduced efficient residual blocks and dense connections to extract discriminative features for HSI classification to avoid this phenomenon. Meanwhile, other researchers are exploring CNN architectures for spectral and spatial attention networks to improve HSI classification.

Nevertheless, most deep learning-based HSI classification maps are usually excessively smooth for the classification of cross-class edge features. To solve this problem, several strategies are introduced to avoid the limitation of fixed kernels. In the local area of HSI, neighboring pixels that are near the central image element and are of the same feature type are considered to provide useful information. In contrast, the neighboring image elements that are not of the same feature types as the central image element are considered to provide additional information. In the field of computer vision, a deformable convolutional network has been proposed to add offset to the sampling position of the convolutional kernel so that the kernel can transform the position instead of sampling the feature map at a fixed position [29]. Based on this, Zhu et al. [30] further proposed a deformed HSI classification network for HSI by applying a 2-D offset to the input HSI feature map and introducing deformed convolution sampling locations. Paoletti and Haut [31] proposed an adaptive convolution approach that is based on a combination of a deformation kernel and a deformation convolution layer to match its receptive field with the input data. Although previous methods proposed irregular CNNs with deformable kernels to effectively characterize the spatial structure of HSI, more research is needed on how to fully utilize the spectral-spatial structure of HSI through CNNs.

This article proposes a new multiscale adaptive convolution structure for HSI classification, which can adaptively adjust its kernel shape using different superpixels of HSI. First, linear discriminant analysis (LDA) [32] automatically learns low-dimensional spectral features from a limited number of training samples. Then, simple linear iterative clustering (SLIC) [33] obtains spatial features of HSIs from low-dimensional spectral features. Next, we utilize the proposed multiscale convolution combined with spectral and superpixel adaptive convolution operations to enhance the robustness of the network structure. Based on the adaptive convolution, the adaptive convolution unit (AConvU) is constructed to obtain the spatial-spectral features of HSIs. Several AConvU hierarchical residual structures are proposed at a single superpixel scale to extend the receptive field of the adaptive convolution operation and to mine the depth features. Finally, the multiple superpixels AConvU hierarchical residual structures are passed through the Softmax function to obtain HSI probability maps at different scales and voted to form the final classification maps.

The following are the main contributors to this article.

1) An adaptive convolution with an adaptive convolution kernel shape is proposed, and the superpixel adjusts the convolution kernel shape to fully exploit the spatial structure of HSI with the advantages of fewer parameters and enhanced robustness.

2) It is combined with adaptive convolution to construct an AConvU that can model different local HSIs. At the same time, the AConvU constructs a hierarchical residual structure to enlarge the spatial-spectrum association characteristics of the receptive field.

3) Superpixels of different scales form adaptive convolution and hierarchical residual structures of different scales to mine HSI details of different degrees and complement each other to improve the classification effect.

4) A strategy of multiscale adaptive deformable kernels is proposed for the HSI data classification task. Moreover, the superiority of multiscale adaptive convolution (MSAC) over other advanced algorithms in three widely used public HSI scenarios, respectively, and the validity of the method is verified using actual data from the Gaofen-5 (GF-5) satellite.

The rest of this article is organized as follows. Section II describes the related work, including that on the superpixel segmentation algorithm as well as the Res2Net architecture. Section III introduces the proposed MSAC method. Section IV evaluates the classification performance of the proposed method with other comparing the experimental results on three real hyperspectral datasets and presents the practical application of the MSAC is introduced. The proposed method is analyzed by experiments conducted with GF-5 satellite data. Finally, Section V concludes the article.

II. RELATED WORK

A. Dimension Reduction

Dimensionality reduction methods are generally divided into unsupervised and supervised methods, among which, unsupervised methods are dominated by PCA and supervised by LDA. Unlike PCA, which maximizes the variance of HSI data, LDA mainly projects the HSI data to the low-dimensional space, and the variance within classes is minimized and the variance between classes is maximized after the projection; in other words, similar data are close together and different classes are dispersed.

We assume that the dataset \(D = \{x_1, y_1\}, \{x_2, y_2\}, \{x_3, y_3\}, \ldots; \{x_m, y_m\}\) is denoted as the \(n\)th \(d\)-dimensional vector. \(x_j\) \((j = 1, 2, 3, \ldots, k)\) is the set of samples of the \(j\)th class, and \(y_m \in C_1, C_2, \ldots, C_k\) is \(k\)th class. First, calculate the interclass scatter matrix \(S_b\) and the intraclass scatter matrix \(S_w\) as follows:

\[
S_b = \sum_{j=1}^{k} N_j (\mu_j - \mu)(\mu_j - \mu)^T
\]

\[
S_w = \sum_{j=1}^{k} \sum_{x \in X_j} (x - \mu_j)(x - \mu_j)^T.
\]
Among them, \( \mu_j = \frac{1}{N} \sum_{x \in X_j} x \), \( N_j \) denotes the number of samples of the \( j \)-th class. The optimization is achieved by constructing the objective matrix \( J \)

\[
J = \frac{W^T S_i W}{W^T S W} = \prod_{i=1}^{d} \frac{w_i^T S_i w_i}{w_i^T S w_i}.
\]

(3)

Then, the target matrix is eigendecomposed to find the eigenvectors corresponding to the \( d \) largest eigenvalues and form the projection matrix \( W \).

**B. Superpixel Segmentation**

Recently, superpixel segmentation algorithms have been applied to image classification research [34]. Among them, SLIC have been successfully used in HSI classification task, to explore the spatial texture information of HSIs. The SLIC algorithm uses the oversegmentation method to identify superpixels, and the main idea is to use the K-means algorithm to locally cluster and segment superpixels effectively. To be specific, the distance from the center of each cluster to the pixel is calculated by a \( 2V \times 2V \) block, where \( V = \sqrt{N/C} \), \( N \) denotes all pixels number and \( C \) is the total number of superpixels.

Generally speaking, the SLIC algorithm is implemented by the following steps: The first step is to screen \( C \) initial cluster centers from the HSI. The second step is to divide each pixel point into the nearest cluster center and construct different clusters. The clustering process is iterated until the cluster center positions are stable. The definition of distance in SLIC is as follows:

\[
D_{\text{SLIC}} = D_{\text{spectral}} + \frac{\lambda}{\rho} D_{\text{spatial}}
\]

(4)

where \( D_{\text{spectral}} \) is the distance between pixel \( i \) and pixel \( j \) in the spectral dimension, which is used to ensure the homogeneity inside the superpixel, and is described as follows:

\[
D_{\text{spectral}} = \sqrt{\sum_{d=1}^{D} (p_{i,d} - p_{j,d})^2}
\]

(5)

where \( p_{i,d} \) is the band \( d \) value of pixel \( i \). Moreover, \( D_{\text{spatial}} \) denotes the distance between pixel \( i \) and pixel \( j \) in spatial position, and is defined as follows:

\[
D_{\text{spatial}} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}
\]

(6)

where \((x_i, y_i)\) is the location of pixel \( i \), and \( \lambda \) and \( \rho \) are the scale parameters.

**C. Res2Net Module**

The traditional CNNs acquires multilayer features by overlaying a varying amount of convolutional layers, but those features have a relatively fixed size receptive fields. In contrast, Res2Net [35] introduces levels as a fundamental factor in addition to the existing of depth, width, and cardinality dimension. In Res2Net, a residual-like hierarchical connectivity structure in a single residual module varies the receptive fields at a more subtle level as a way to extract details and global features.

![Fig. 1. Res2Net module (the level dimension \( l = 4 \).)](image)

Fig. 1 shows a hierarchical residual unit with four levels. First, after \( 1 \times 1 \) convolution, the feature map is divided into \( l \) feature map subsets, which are represented by \( X_i \), where \( i \in \{1, 2, \ldots, l\} \). Unlike the input feature map, each subset of features \( X_i \) with the same spatial size and the channel number as \( 1/l \). Except for \( X_1 \), each \( X_i \) is denoted by \( K_i(\cdot) \) after a corresponding \( 3 \times 3 \) convolution. \( Y_i \) denotes the output of \( K_i(\cdot) \). The feature subset \( X_i \) is summed with the output of \( K_{i-1}(\cdot) \) and fed into \( K_i(\cdot) \). \( X_i \) omits the \( 3 \times 3 \) convolution and can reduce the parameters at the same times as increasing \( l \). Thus, \( Y_i \) is denoted as follows:

\[
Y_i = \begin{cases} 
  X_i, & i = 1; \\
  K_i(X_i), & i = 2; \\
  K_i(X_i + Y_{i-1}), & 2 < i \leq l 
\end{cases}
\]

(7)

where each \( 3 \times 3 \) convolutional operator \( K_i(\cdot) \) extracts feature information from all feature partitions \( X_i \), \( j \leq i \). Each time a feature segmentation \( X_i \) through the \( 3 \times 3 \) convolutional operator, the output result has a larger receptive field than \( X_j \). The output of the Res2Net module contains different numbers and different combinations of received field sizes to avoid the combinatorial explosion effect. The Res2Net module adopts a multilevel approach, which facilitates the extraction of global and local information. Connecting all the tiles and passing them by \( 1 \times 1 \) convolution, which can fuse information better from different levels.

**III. PROPOSED APPROACH**

The proposed MSAC method is a typical end-to-end HSI classification framework, as shown in Fig. 2, and the proposed method is summarized as Algorithm 1. The original HSI is represented as \( X \in \mathbb{R}^{H \times W \times B} \) and the classification label for each pixel is represented as \( Y \in \mathbb{R}^{H \times W \times C} \), where the heights, widths, bands, and class number of HSI are denoted by \( H, W, B \), and \( C \), respectively. MSAC differs from the conventional HSI classification framework for deep learning in two aspects: 1) the use of the adaptive convolution kernel (AConv) based on superpixels; and 2) the adaptive convolutional embedded hierarchical residual structure. Each component is made up of specific steps, as shown below.
A. Superpixel Generation

To reduce the complexity of HSI spatial feature acquisition, LDA is used to preprocess the HSI by dimensionality reduction. Then, the SLIC method partitions the reduced-dimensional HSI into several superpixels that have similar spectral features. Hence, the HSI is divided into $Z$ superpixels by the LDA and SLIC methods, where $Z = \lfloor \frac{H \times W}{\lambda} \rfloor$ and the splitting scale used to the size of superpixels is $\lambda (1 \leq \lambda)$.

B. Adaptive Convolution

In this section, we introduce the proposed adaptive convolution, in which the shape of the kernel can be controlled by superpixels.

1) Regular Convolution Operation: Generally speaking, these are the steps of the 2-D convolution operation in Fig. 3. In convolutional 2-D convolution, the output feature map $y$ on each spatial location $q_0 = (x, y)$, we have

$$y_j(q_0) = \sum_{q_n \in \mathbb{R}} w_j(q_n) \odot X(q_0 + q_n) \quad (8)$$

where $q_n$ denotes the spatial position in $\mathbb{R}$, $w_j(q_n)$ expresses the pixel vector of the $j$th regular kernel at position $q_n$, $X(q_0 + q_n)$ is the input feature of pixel at position $q_0 + q_n$, $\odot$ indicates inner product, and the output feature value of the $j$th channel at position $q_0$ is denoted as $y_j(q_0)$. Note that bias and activation functions are excluded for convenience of description.

2) Adaptive Convolution Operation: HSI suffers from the problem of a limited number of training samples and the deficiency of a large number of parameters in a deep network. Therefore, the network’s resistance to overfitting can be improved by reducing the settings of the network parameters. Inspired by the deformed convolution in [35]–[37], superpixel-based adaptive convolution is proposed. In this case, in the adaptive convolution illustrated in Fig. 4, the input feature map is divided into 2-D bands along the spectral dimension, and each band is convolved with a 2-D superpixel to yield a 2-D feature band as follows:

$$y_j(q_0) = \sum_{q_n \in \mathbb{R}} A_j(q_0, q_n; S) \odot X(q_0 + q_n) \quad (9)$$

where $A_j(q_0, q_n; S)$ indicates that the $j$th adaptive kernel at sampling location $q_0$ and $y_j$ is the $j$th channel of the output feature map. The adaptive kernel $A$ is obtained from the local of
the superpixel map $S$ at the current location $q_0$ using the same sampling grid and the shape is determined by the anisotropic kernel. The adaptive kernel can be separated into kernels

$$A_j(q_0, q_n; S) = a(q_0, q_n; S)w_j(q_n)$$

(10)

where $w_j$ denotes the $j$th isotropic kernel, while $a(q_0, q_n; S)$ is the anisotropic kernel

$$a(q_0, q_n; S) = \exp\left(-\frac{\|S(q_0 + q_n) - S(q_0)\|^2}{\sigma^2}\right)$$

(11)

where $\sigma$ is the sensitivity coefficient. The superpixel map $S$ at the current location $q_0$ is calculated to obtain the anisotropic kernel $a$. The isotropic kernel $w_j$ is shared in the 2-D convolution process, independent of the sampling location. It is worth nothing that $a$ is 2-D, while $w_j$ is multidimensional, and needs to be multiplied point by point for each slice along the spectral dimension.

In (11), the sensitivity value $\sigma$ is the self-updating parameter, which is to be updated by the back-propagation (BP) [38] during training process. In the adaptive convolution layer, the error is first propagated to adaptive kernel $A_j$ by the chain rule, and then, to the anisotropy kernel $a$ and the sensitivity value $\sigma$. The differential equation for calculating $\sigma$ is as follows:

$$\frac{\partial A_j(q_0, q_n; S)}{\partial \sigma} = \frac{\partial A_j(q_0, q_n; S)}{\partial a(q_0, q_n; S)} \frac{\partial a(q_0, q_n; S)}{\partial \sigma}$$

$$= w_j(q_n) \frac{2\theta}{\sigma^3} \exp\left(-\frac{\theta}{\sigma^2}\right)$$

(12)

where $\theta = \|S(q_0 + q_n) - S(q_0)\|^2$. The $\sigma$ is then converted and updated to

$$\sigma \leftarrow \sigma - r \times \frac{\partial A_j(q_0, q_n; S)}{\partial \sigma} \otimes \frac{\partial A_j(q_0, q_n; S)}{\partial \sigma}$$

(13)

where $\gamma$ serves as the loss function of the proposed method and the learning rate is $r$. $\sigma$ is updated by gradient descent. We use the tool in TensorFlow [39] to compute and update the differential equations of trainable parameters. In the bandwise adaptive convolution, the adaptive kernel $A_j$ is 2-D. The number of input and output feature bands is the same and equal to the number of 2-D adaptive kernels. Since adaptive convolution only extracts the spectral information of the individual band of the HSI input, but not the spectral information. Therefore, a combination of the band convolution layer and a 2-D convolution layer of size $1 \times 1$ is considered.

### C. Adaptive Convolution Unit

Based on the adaptive convolution, a hidden unit that contains a sequence of layers is proposed to extract spatial-spectral features and named AConvU. There are two inputs to an AConvU as illustrated in Fig. 5: 1) The superpixel map; and 2) the input feature map generated by the previous layer. Batch normalization (BN) [40] was used to minimize covariance drift using the feature map of the previous layer of AConvU as input. We take $J_l \in \mathbb{R}^{H \times W \times U}$ as the input feature mapping of the $l$th AConv, and $U$ denote the bands of the input feature mapping. The normalized data $\tilde{J}_l$ can be obtained from

$$\tilde{J}_l = \frac{J_l - E(J_l)}{\text{Var}(J_l)}$$

(14)

where $E(\cdot)$ and $\text{Var}(\cdot)$ denote the expectation function and variance function, respectively. A nonlinear transform is then to be applied to each pixel in $\tilde{J}_l$ using $1 \times 1$ size convolution to extract the HSI spectral features. At each position $q_0$ on the output feature map $Z_l \in \mathbb{R}^{H \times W \times M}$, $M$ denotes the output bands number, there is

$$Z_{l,j}(q_0) = h(w_{l,j} \otimes \tilde{J}_l(q_0) + c_{l,j})$$

(15)

where $w_{l,j}$ and $c_{l,j}$ is the $j$th 1-D kernel in the $l$th AConvU, $c_{l,j}$ is the $j$th bias in the $l$th AConvU, $h(\cdot)$ denotes the activation function, and $Z_{l,j}$ is used as the $j$th band of the output feature map. The spectral features of each pixel are extracted separately using $1 \times 1$ convolution, while the spatial information of HSI is extracted by adaptive convolution of the feature map $J_l$. For each location $q_0$ on the output feature map $O_l \in \mathbb{R}^{H \times W \times V}$

$$O_{l,j}(q_0) = h\left(\sum_{q_0 \in \mathbb{R}} A_{l,j}(q_0, q_n; S) \otimes Z_{l,j}(q_0 + q_n) + p_{l,j}\right)$$

(16)

where $A_{l,j}$ is the $j$th adaptive kernel in the AConvU, $p_{l,j}$ are the bias in the AConvU, and $Z_{l,j}$ and $O_{l,j}$ represents the $j$th channel of the input and output feature maps, respectively.

### D. Architecture of MSAC

Inspired by Res2Net [35], we propose the residual connection of different AConvUs in our network. Fig. 2 is described as a three-scale multilevel hierarchical residual structure. Each scale contains three levels, in which $\otimes$ represents separation operation and $+$ denotes catenation operation. Let the input of the hierarchical residual structure be $X$, and the output be denoted as $Y$. Each AConvU $X_i$, where $i \in 1, 2, \cdots, l$, and $l$ is the number of AConvU. The spatial size and dimensions of $X_i$ are the same as the input $X$. Every $X_i$ enters into the corresponding AConvU operations, denoted by $H_i(\cdot)$ for the AConvU operation, while $Y_i$ denotes the output of $H_i(\cdot)$. For the convenience of obtaining hierarchical, the output of $H_i(\cdot)$ is added to the feature subset $X_i$, which is then fed into $H_i(\cdot)$. 

![Fig. 5. Adaptive convolution unit.](image_url)
Thus, $Y_i$ can generally be expressed as

$$Y_i = \begin{cases} H_i \left( X_i \right), & i = 1; \\ H_i \left( X_i + Y_{i-1} \right), & 2 < i \leq l. \end{cases} \quad (17)$$

Within the hierarchical residual structure, information from different subsets $X_j (j \leq i)$ is eventually available from each AConvU operation $H_i(\cdot)$ so that feature $X_i$ has a larger receptive field compared to than $X_j$. In addition, the tails of hierarchical residual structure are used to represent multilevel features by connecting feature mappings of different receptive fields. Large AConvU numbers are able to learn features with richer receptive fields than smaller AConvU numbers. Multilevel spectral and spatial features are extracted by a specific AConv. As shown in Fig. 2, different superpixels combine multiple hierarchical residual structures to feature probability maps of different scales of HSIs, which can fully explore the spatial characteristics and spectrum features of HSI.

The feature map of the hierarchical residual structure is subjected to the Softmax function to obtain the probabilities of different landcover classes. Then, voting rules are applied to the feature probabilities of different scales, to improve the classification accuracy of the final classification result.

IV. EXPERIMENTS AND ANALYSIS

In this section, we detail the experiments performed on HSI with deep-learning architectures; they are presented and compared with other traditional approaches as well as advanced ones.

A. Dataset Descriptions

To test the classification performance of the proposed method, three widely used HSI datasets are used in the experiments: Indian Pines (IP), University of Pavia (UP), and Salinas Valley (SV). Detailed descriptions of the three datasets are given below as follows.

1) **Airborne visible infrared imaging spectrometer (AVIRIS) IP Dataset**: The IP dataset was acquired from images taken by the AVIRIS sensor in Northwestern Indiana. The dataset featured 220 spectral bands (400–2500 nm) and a resolution of 20 m per pixel. The entire dataset is represented as a data cube with height and width of 145, and band number of 200. After removing the interference bands such as moisture absorption and noise, the spectral band number is saved as 200. The ground truth contains 10 249 labeled pixels for 16 landcover classes. Fig. 6(a)–(c) show the false-color maps of IP dataset with the color combinations of the corresponding categories, and the correspondence between the 16 different colors of landcover classifications, respectively. In the experiments of this article, we randomly select 10, 1, and 89% of the samples from each class as the training set, validation set, and test set, respectively. The detailed sample numbers for each feature category are listed in Table I.

2) **ROSIS UP Dataset**: The second HSI dataset which is an urban image of a Northern Italy city taken by the Reflective Optics System Imaging Spectrometer (ROSIS) in 2002. The entire image contains nine classes with a total of 610 × 340 pixels. After removing the 12 noise bands, the image contains 115 spectral bands ranging from 430 to 860 nm. Fig. 7(a)–(c) shows the false-color synthesis of the images of the UP dataset with the corresponding reference data and the legend. The 10, 1, and 89% samples were randomly selected from each class as the training set, validation set, and test set, respectively, as listed in Table II.

3) **AVIRIS SV Dataset**: The third dataset is SV, collected by the AVIRIS sensor shot in Salinas Valley, California. The SV dataset contains 204 spectral bands with 512 × 217 pixels in size after removing moisture absorption and other noise bands. Fig. 8(a)–(c) illustrates the false-color composite of the Salinas images, the corresponding reference data, and the correspondence between the 16 different landcover classes with different colors. A

---

**Algorithm 1**: MSAC for HSI Classification.

**Inputs**:
1. The raw HSI $X$;
2. The training sample label $L_{tr}$;
3. The AConvU $U$;
4. The superpixel segmentation map $M$;
5. The superpixel scale $S$.

**Output**: Predicted labels for test pixel $Y$.

1. The LDA algorithm incorporates the training sample labels $L_{tr}$ to reduce the dimensionality of the original HSI $X$.
2. The SLIC algorithm segments the image after the LDA dimension reduction, and obtains the superpixel segmentation map $M$ according to different superpixel scales $S$.

For $s$ in $S$

   For $u$ in $U$

   3. $X_{BN}$ is obtained by BN of the input feature map.
   4. $X_{1 \times 1}$ is obtained through $1 \times 1$ convolution on $X_{BN}$, and the spectral features of HSI are extracted.
   5. The sigmoid function is used for $X_{1 \times 1}$ to obtain $X_{sigmoid}$.
   6. $X_{sigmoid}$ is obtained through the dropout to avoid algorithm overfitting.
   7. Superpixel map $M_s$ and $X_{sigmoid}$ use the AConv operation to obtain the adaptive feature map $X_{AConv}$.
   8. $X_{Asigmoid}$ is obtained from $X_{AConv}$ through the sigmoid function.
   9. The dropout is used to obtained the feature map $X_{dropout}$.

End For

10. Feature maps at different scales are used to obtain feature landcover probability maps $Y_s$ through the Softmax function.

End For

11. The voting rules extract the maximum probability of each type of feature in the multiscale probability map to obtain the final classification map $Y$. 

---
Fig. 6. IP dataset. (a) False-color image of IP dataset. (b) Ground truth categorization map. (c) Legend.

| Class No. | Class Name  | Train | Val  | Test |
|-----------|-------------|-------|------|------|
| 1         | Alfalfa     | 5     | 2    | 39   |
| 2         | Corn_N      | 143   | 1271 |      |
| 3         | Corn_M      | 83    | 739  |      |
| 4         | Corn        | 23    | 212  |      |
| 5         | Grass_P     | 49    | 430  |      |
| 6         | Grass_T     | 74    | 649  |      |
| 7         | Grass_M     | 2     | 24   |      |
| 8         | Hay_W       | 48    | 426  |      |
| 9         | Oats        | 17    |      |      |
| 10        | Soybean_N   | 96    | 867  |      |
| 11        | Soybean_M   | 246   | 2185 |      |
| 12        | Soybean_C   | 61    | 526  |      |
| 13        | Wheat       | 21    | 182  |      |
| 14        | Woods       | 129   | 1124 |      |
| 15        | Buildings   | 38    | 345  |      |
| 16        | Stone       | 9     | 2    | 82   |

Total Number: 1029, 102, 9118

Table I
LANDCOVER CLASSES AND SAMPLES OF THE IP DATASET

Fig. 7. UP dataset. (a) False-color image of UP dataset. (b) Ground truth categorization map. (c) Legend.

| Class No. | Class Name | Train | Val  | Test | Val  |
|-----------|------------|-------|------|------|------|
| 1         | Asphalt    | 664   | 66   | 5901 |      |
| 2         | Meadows    | 1865  | 187  | 16597|      |
| 3         | Gravel     | 210   | 21   | 1868 |      |
| 4         | Trees      | 307   | 31   | 2726 |      |
| 5         | Metal sheets| 135  | 14   | 1196 |      |
| 6         | Bare soil  | 503   | 50   | 4476 |      |
| 7         | Bitumen    | 133   | 11   | 1186 |      |
| 8         | Bricks     | 369   | 37   | 3276 |      |
| 9         | Shadows    | 95    | 10   | 842  |      |

Total Number: 4281, 427, 38068

Table II
LANDCOVER CLASSES AND SAMPLES OF THE UP DATASET

random sample of 5%, 1%, and 94% from each class is selected for training, validation, and testing. The detailed numbers are listed in Table III.

B. Experimental Setting

In the experiment, MSAC is conducted via TensorFlow with the Adam optimizer [41]. The hyperparameter selection in our MASC, including segmentation scale $S$, learning rate $r$, epoch $N$, kernel size $K$, AConvU $U$, dimensional $d$, and dropout $\eta$, are shown in Table IV.

Fig. 8. SV dataset. (a) False-color image of SV dataset. (b) Ground truth categorization map. (c) Legend.

Table III
LANDCOVER CLASSES AND SAMPLES OF SV DATASET

| Class No. | Class Name | Train | Val  | Test |
|-----------|------------|-------|------|------|
| 1         | Weeds_1    | 101   | 20   | 1888 |      |
| 2         | Weeds_2    | 187   | 36   | 3503 |      |
| 3         | Fallow     | 99    | 20   | 1857 |      |
| 4         | Fallow_P   | 70    | 14   | 1310 |      |
| 5         | Fallow_S   | 134   | 26   | 2518 |      |
| 6         | Stubble    | 198   | 40   | 3721 |      |
| 7         | Celery     | 179   | 36   | 3364 |      |
| 8         | Grapes     | 564   | 113  | 10594|      |
| 9         | Soil       | 311   | 62   | 5830 |      |
| 10        | Corn       | 164   | 33   | 3081 |      |
| 11        | Lettuce_4wk| 54    | 11   | 1003 |      |
| 12        | Lettuce_5wk| 97    | 19   | 1811 |      |
| 13        | Lettuce_6wk| 46    | 9    | 861  |      |
| 14        | Lettuce_7wk| 54    | 11   | 1005 |      |
| 15        | Vineyard_U | 364   | 73   | 6831 |      |
| 16        | Vineyard_T | 91    | 18   | 1698 |      |

Total Number: 2713, 541, 50875
TABLE IV
HYPERPARAMETER SETTINGS FOR DIFFERENT DATASETS

| Datasets | S          | r   | N   | K   | U   | d   | η   |
|----------|------------|-----|-----|-----|-----|-----|-----|
| IP       | [500,750,1000] | 0.01 | 300 | 7×7 | 4   | 32  | 0.1 |
| UP       | [500,750,1000] | 0.01 | 400 | 3×3 | 4   | 32  | 0.1 |
| SV       | [500,750,1000] | 0.01 | 700 | 5×5 | 4   | 20  | 0.1 |

Fig. 9. Sensitivity of different size adaptive kernels on different datasets.

To quantitatively and qualitatively compare the performance of these nine methods on three real HSI datasets, each class accuracy, overall accuracy (OA), average accuracy (AA), and kappa coefficient (Kappa) are employed to evaluate the performance of the proposed MSAC. All experiments were repeated ten times using randomly selected training samples to obtain the mean and standard variances of OA, AA, and Kappa. The best numerical markers in the article are bolded.

C. Parameter Sensitivity Analysis

To verify the sensitivity of the proposed method to the different parameters, therefore, in this section, the effect of various parameters on the performance of MSAC is discussed.

1) Influence of Different Sizes of AConv: Different parameters have different effects on the accuracy of the MSAC architecture. First, the performance of the proposed MSAC with different sizes of AConv is discussed. In all the HSI datasets, the kernel size K input to the MSAC varies from 3 × 3 to 7 × 7, while the number of AConvUs U is fixed at 3. As can be seen from Fig. 9, the accuracy of the MSAC on the IP, UP, and SV datasets increases with the size of the adaptive kernel. The reason for this phenomenon may be the increase in adaptive kernel size, where the receptive field can combine the irregular structure of the superpixels to better utilize the spatial information of the HSI. The adaptive kernel size of IP dataset was set to 7 × 7, while those of the UP and SV datasets were set to 3 × 3 and 5 × 5, respectively, due to the stable OA fluctuation.

2) Influence of Different Numbers of AConvUs: Most believe the gradient disappearance problem exists and that deep network does not necessarily boost performance. In this part, the number of AConvUs U is set from 3 to 5, and the performance ability of the MSAC under different numbers of AConvUs is tested. OA is taken as the evaluation indexes. Therefore, an appropriate number of AConvUs boosts its stability and capability. As shown in Fig. 10, for the IP dataset, MSAC performs best when the number of AConvUs is 3. For the UP dataset, setting the number of AConvUs to 4 is the best choice. For the SV dataset, the spatial distribution structure of the SV dataset is simple and can easily cause overfitting, which may lead to the decrease of OA with the increase of the number of AConvUs. As SV is composed of large-scale landcovers, the increase in the number of AConvUs will adversely affect the performance improvement of the MSAC method, which is also reflected in UP. However, the characteristics of AConvU are detrimental to HSIs involving large-scale land features. Therefore, an overly large AConvU number may result in decreased performance on UP and SV. As the number of AConvUs increases, the running increases accordingly. Therefore, the number of AConvUs configured in IP, UP, and SV is 4.

3) Influence of Different Segmentation Scales: For the selection of superpixel scale range, we do the following experiments on three datasets. In the first one, pick scale1 in the range [500, 550, 600] at 50 intervals in the small scale range. Then, in the medium scale range, we choose the range of scale2 in 250 intervals as [500, 750, 1000]. Finally, scale3 is selected as [500, 1500, 2500] in a large scale range with an interval of 1000.

The mean values of OA, AA, and Kappa are used to test the effect of different scale ranges on the method. As shown in Fig. 11, on IP dataset, it can be seen that all three evaluation indicators show an increasing trend when the scale range is gradually increased. However, on the other two datasets UP, SV, OA, AA, and Kappa performed poorly both at the small scale range scale1 and at the large scale range scale3. One of the reasons maybe due to the superpixel segmentation approximation in the small scale range scale1 and consistent classification results, leading to no improvement in classification ability. Although the large scale range scale3 has some improvement relative to the low scale, it does not help the effective overall effect due to the large interval of classification scales and inconsistent position of respective attention for HSI classification results. Finally, it can be observed from three subfigures that when the scale range is [500, 750, 1000], all three evaluation metrics have a large improvement to discover the detailed information of HSI better, and the classification results between different scales can complement the detailed feature better. Therefore, scale2 was applied to all datasets in the experiments.

4) Influence of Learning Rate and Epoch: The learning rate r and epoch N control the step of gradient descent in the MSAC model, which can also affect the classification performance. The candidate set of learning rates of the three datasets is {0.01, 0.001, 0.0001}. The epoch ranges of IP, UP, and SV datasets are fixed as {100, 200, 300}, {200, 300, 400}, and 500-700, respectively. The epoch parameters above were set as the iteration number of single-scale superpixel adaptive convolution execution, in the MSAC was always fixed to the same epoch, and we tested the effects of parameters on the OA. Fig. 12 reports OA with different learning rates and epochs on the three datasets. From this figure, It is clear that the larger the epoch, the more stable classification accuracy, and the smaller the epoch, the lower the classification accuracy. In addition, the two parameters have an obvious impact on the IP and UP datasets, but less of an impact on the SV dataset. Based on the experimental results
Fig. 10. Performance of MSAC with different number of AConvU on different datasets.

Fig. 11. The effect of different superpixel scale ranges on different datasets. (a), (b), and (c) denote the performance on the IP, UP, and SV datasets, respectively. scale1 = [500, 550, 600], scale2 = [500, 750, 1000], and scale3 = [500, 1500, 2500].

Fig. 12. Performance of MSAC on different datasets with different epoch $N$ and learning rate $r$.

for the three datasets, $r$ and $N$ were set as {0.01, 300}, {0.01, 400}, and {0.01, 700}.

D. Comparisons With Other Methods

Throughout this section, we quantitatively and qualitatively evaluate the classification performance of the MSAC by comparing the proposed method with advanced methods including machine learning and representative deep-learning methods in [42]. In particular, the methods considered include GRU [23], 1D-CNN [24], 3D-CNN [25], spectral-spatial residual network (SSRN) [28], content-guide CNN (CGCNN) [37], 2D-CNN [42], SVM with radial basis function [43], multilayer perceptron (MLP) [44], and recurrent neural network (RNN) [45]. To reflect the fairness of the experiment, we used the hyperparameters recommended in their original paper for all comparison methods.

1) Results on IP Dataset: The classification accuracy and result plots under different methods are given in Table V and Fig. 13. For the IP dataset, our method achieves the best classification performance on OA, AA, and Kappa indices, which marks an advance in model and theoretical design. Although the SSRN model achieves good results, the classification accuracy in Grass_M, Buildings, and Stone is not satisfactory, while the method in this article has good adaptability to classification of HSI detailed features. We observe that 2D-CNN shows better classification advantages than 1D-CNN and 3D-CNN, and achieves better accuracy in a few classes. Overall, the proposed MSAC method consistently outperforms the traditional 1D-CNN, 2D-CNN, and 3D-CNN by a large margin. Although, the accuracy levels of the SSRN and CGCNN are higher than...
TABLE V
OBJECTIVE INDEXES OBTAINED ON THE IP DATASET BY SVM, SSRN, MLP, 1D-CNN, 2D-CNN, 3D-CNN, RNN, GRU, CGCNN, AND MSAC METHODS

| Class | SVM | SSRN | MLP | 1D-CNN | 2D-CNN | 3D-CNN | RNN | GRU | CGCNN | MSAC |
|-------|-----|------|-----|--------|--------|--------|-----|-----|-------|------|
| 1     | 0.70 | 0.70 | 0.70 | 0.70   | 0.70   | 0.70   | 0.70 | 0.70 | 0.70  | 0.70 |
| 2     | 0.69 | 0.69 | 0.69 | 0.69   | 0.69   | 0.69   | 0.69 | 0.69 | 0.69  | 0.69 |
| 3     | 0.68 | 0.68 | 0.68 | 0.68   | 0.68   | 0.68   | 0.68 | 0.68 | 0.68  | 0.68 |
| 4     | 0.67 | 0.67 | 0.67 | 0.67   | 0.67   | 0.67   | 0.67 | 0.67 | 0.67  | 0.67 |
| 5     | 0.66 | 0.66 | 0.66 | 0.66   | 0.66   | 0.66   | 0.66 | 0.66 | 0.66  | 0.66 |
| 6     | 0.65 | 0.65 | 0.65 | 0.65   | 0.65   | 0.65   | 0.65 | 0.65 | 0.65  | 0.65 |

TABLE VI
OBJECTIVE INDEXES OBTAINED ON UP DATASET BY SVM, SSRN, MLP, 1D-CNN, 2D-CNN, 3D-CNN, RNN, GRU, CGCNN, AND MSAC METHODS

those of other comparison methods, they are lower than that of the MSAC. The other classification models (i.e., SVM, MLP, and GRU) achieve similar accuracy levels, while RNN provides the lowest OA, AA, and Kappa values.

2) Results on UP Dataset: We can observe that Table VI gives the evaluation results of the different methods on the UP dataset. Consistent with the results on the IP dataset, the results in Table VI show that the MSAC method proposed in the article is a dominant position and significantly outperforms most of the compared methods, which again validates the performance advantage of the MSAC proposed in this article. In particular, the OA, AA, and Kappa of the method in this article are higher...
compared to convolution-based model, i.e., 1D-CNN, 2D-CNN, and 3D-CNN. This is due to the fact that the spectral bands of “Asphalt,” “Bricks,” and “Shadows” are most similar, and the adaptive convolution and extended receptive fields facilitate better capturing the spatial features of HSI and clearly distinguish them, while their simplicity prevents the overfitting problems. As can be seen in Fig. 14, 2D-CNN provides significantly better OA, AA, and Kappa values than 1D-CNN and 3D-CNN due to the poor generalization ability of 1D-CNN and the overfitting and complexity issues of 3D-CNN. Furthermore, the OA of MSAC is higher compared to SSRN, while lower than the AA and Kappa coefficient of SSRN.

3) Results on SV Dataset: As mentioned earlier, the total number of training and testing samples for the SV dataset is shown in Table III. The results in Table VII and Fig. 15 reveal that the proposed method in this article presents a constant performance gain in all the measurements (i.e., OA, AA, and Kappa) compared to SSRN, CGCNN, and all other methods. When the training samples of Lettuce_6wk and Vineyard_U are only 5%, the proposed methods are all optimal. In particular, when the category is Vineyard_U, the accuracy increases by 1.91% and 24.92%, respectively, compared with SSRN and CGCNN. For the recurrent model, the classification results of RNN are worse than those of GRU. And as seen from the performance of SVM,

| Class No. | Accuracy of classification obtained by different comparison methods (in %) |
|----------|------------------------------------------------------------------------|
| SVM      | SSRN                     | MLP | 1D-CNN | 2D-CNN | 3D-CNN | RNN | GRU | CGCNN | MSAC |
| 1        | 99.95±0.06               | 100.0±0.00 | 98.95±0.72 | 99.76±0.19 | 99.98±0.02 | 100.0±0.00 | 99.00±0.24 | 99.35±0.24 | 100.0±0.00 | 99.98±0.04 |
| 2        | 99.92±0.15               | 99.93±0.13 | 99.75±0.11 | 99.66±0.02 | 99.97±0.05 | 99.99±0.07 | 99.87±0.11 | 99.92±0.07 | 99.39±1.10 | 99.58±1.12 |
| 3        | 96.58±1.33               | 99.96±0.09 | 99.06±0.41 | 99.70±0.17 | 99.69±0.33 | 99.98±0.02 | 98.50±0.61 | 99.15±0.48 | 65.92±4.43 | 100.0±0.00 |
| 4        | 98.13±1.07               | 97.47±0.77 | 98.58±0.30 | 99.29±0.21 | 99.71±0.15 | 99.73±0.52 | 99.11±0.51 | 99.30±0.18 | 99.86±0.14 | 99.97±0.05 |
| 5        | 98.71±0.74               | 99.90±0.11 | 99.82±0.35 | 98.69±0.72 | 99.35±0.60 | 99.19±0.67 | 97.78±0.57 | 98.55±0.49 | 96.76±2.63 | 99.90±0.10 |
| 6        | 99.97±0.03               | 100.0±0.00 | 98.82±0.10 | 98.85±0.11 | 100.0±0.00 | 99.99±0.00 | 99.73±0.13 | 99.78±0.12 | 99.90±0.06 | 99.99±0.03 |
| 7        | 99.83±0.47               | 99.99±0.01 | 98.67±0.15 | 98.77±0.08 | 99.86±0.13 | 99.96±0.04 | 99.57±0.13 | 99.69±0.19 | 99.94±0.03 | 99.95±0.06 |
| 8        | 81.33±1.33               | 97.30±1.00 | 85.36±2.06 | 87.66±1.53 | 94.03±0.89 | 98.13±0.60 | 87.69±2.49 | 88.18±2.04 | 93.69±4.80 | 98.32±1.02 |
| 9        | 99.34±0.10               | 99.93±0.06 | 99.73±0.10 | 99.65±0.41 | 99.98±0.01 | 99.99±0.01 | 99.65±0.33 | 99.52±0.50 | 99.29±0.95 | 99.99±0.19 |
| 10       | 81.53±1.24               | 98.86±0.09 | 94.53±1.01 | 96.66±0.74 | 97.47±0.54 | 99.44±0.30 | 95.10±0.98 | 96.33±0.96 | 96.37±1.81 | 99.91±0.13 |
| 11       | 97.29±1.72               | 99.64±0.83 | 95.59±2.09 | 97.77±0.79 | 99.02±0.04 | 99.70±0.13 | 95.66±1.75 | 96.81±1.81 | 98.85±0.78 | 99.98±0.06 |
| 12       | 97.83±0.72               | 99.47±0.09 | 99.43±2.29 | 99.93±0.04 | 100.0±0.00 | 100.0±0.00 | 99.35±0.48 | 99.44±0.33 | 99.90±0.12 | 99.98±0.03 |
| 13       | 97.19±1.19               | 99.47±0.05 | 98.28±0.83 | 99.31±0.56 | 99.88±0.20 | 99.80±0.34 | 98.60±0.49 | 97.89±0.82 | 95.76±3.28 | 100.0±0.00 |
| 14       | 97.16±1.21               | 99.87±0.24 | 94.55±2.34 | 95.49±1.78 | 98.86±0.67 | 97.70±0.31 | 94.94±1.59 | 96.64±1.16 | 99.07±0.44 | 99.89±0.16 |
| 15       | 79.53±1.11               | 97.93±2.05 | 72.47±3.03 | 74.95±3.01 | 91.66±0.65 | 96.72±1.05 | 70.78±4.35 | 75.53±3.67 | 74.92±15.9 | 99.84±0.16 |
| 16       | 99.11±0.67               | 99.87±0.36 | 98.35±0.54 | 98.45±0.79 | 99.02±0.51 | 99.13±0.40 | 98.32±0.42 | 98.80±0.35 | 98.92±0.73 | 99.92±0.10 |

OA(%)  | 92.56±0.28               | 92.07±0.36 | 92.55±0.24 | 93.91±0.22 | 97.33±0.20 | 99.04±0.19 | 92.61±0.25 | 93.57±0.17 | 93.40±1.8 | 99.79±0.12 |
AA(%)   | 96.16±0.26               | 95.86±0.18 | 95.90±0.26 | 96.81±0.18 | 98.65±0.09 | 99.47±0.12 | 95.86±0.25 | 96.57±0.20 | 94.91±0.79 | 99.83±0.10 |
Kappa   | 100                      | 91.70±0.32 | 98.96±0.40 | 97.70±0.26 | 93.21±0.25 | 97.03±0.22 | 98.93±0.22 | 91.76±0.28 | 92.84±0.19 | 92.63±1.79 | 99.71±0.16 |
E. Classification Results With Different Training and Test Sets

Comparing the effect of different numbers of training samples on the performance of the nine methods on different HSI datasets are demonstrated in Fig. 16. Experiments were carried out on each of the IP, UP, and SV datasets, and the proportion of training samples was 2, 4, 6, 8, and 10%. As seen from the radar plot, the performance of all classification methods tends to increase upward as the number of training samples increase. Although on the IP dataset, the performance of this method is lower than that of CGCNN when the training sample is 2%. However, as the number of training samples increases, MSAC can achieve better performance in comparison to other methods. Compared with the IP and SV datasets, the MSAC consistently outperforms all methods on the UP dataset. However, 3D-CNN shows competitiveness in SV datasets, especially with the increase of training samples.

F. Analysis of Computation Cost

All experiments in this article were conducted on a PC equipped with an Intel i7-8700 CPU and a GeForce GTX1060 GPU. Table VIII tabulates the training and test-run times of...
Table VIII

| Datasets | Times | SVM | SSRN | MLP | 1D-CNN | 2D-CNN | 3D-CNN | RNN | GRU | CGCNN | MSAC |
|----------|-------|-----|------|-----|--------|--------|--------|-----|-----|-------|------|
| IP       | Training | 0.19 | 361.22 | 54.39 | 167.18 | 216.84 | 216.84 | 1759.96 | 115.68 | 459.91 | 378.44 |
|          | Testing  | 3.53 | 4.87  | 0.42  | 1.07   | 1.34   | 3.88   | 32.35 | 1.96 | 1.52  | 7.49  |
| UP       | Training | 0.36 | 1321.5 | 126.26 | 202.2 | 215.29 | 379.81 | 3904.85 | 295.66 | 2872.55 | 1029.31 |
|          | Testing  | 23.39 | 11.48 | 0.87  | 1.68   | 1.92   | 3.23   | 69   | 4.16 | 6.14  | 17.63 |
| SV       | Training | 0.35 | 936   | 161.96 | 268.78 | 723.54 | 251.89 | 9197.85 | 573.27 | 1057.91 | 1122.67 |
|          | Testing  | 22.1  | 27.81 | 1.11  | 2.22   | 3.42   | 4.06   | 180.77 | 8.08 | 6.22  | 12.79 |

Fig. 17. Ablation study on three HSI datasets. Error bar represents the standard deviation. (a) and (b) are the effects of different parts and the proposed method when the training sample is 10%, then (c) is the effect of different parts and the proposed method when the training sample is 5%. (a) IP dataset. (b) UP dataset. (c) SV dataset.

MSAC with the nine comparison methods. The IP, UP, and SV datasets all accounted for 1% of the training sample numbers. The above table shows that a deeper classification model requires more training time for more complex datasets than SVM. This is because the BP algorithm used for training must iterate thousands of epochs to convergence, while only one forward operator is tested. This causes the algorithm to consume more training time than other comparison algorithms. However, MSAC benefits from this long-term training to fully exploit the spatial-spectral information of HSI to obtain a better feature representation capability. SSRN, CGCNN, and the proposed MSAC all adopt a similar residual structure, so the training time is the same in the three datasets. Although the test time of the MSAC is longer than other methods, it significantly improves the classification accuracy.

G. Ablation Study

In order to validate the role of our model in a certain aspect. We further compared the different ablation studies. We evaluate which aspects benefit from the performance of our approach by iteratively removing each of these aspects. The combination of superpixel segmentation and traditional convolution is used to give full play to the spatial characteristics of HSIs. In order to avoid overfitting, the hierarchical residual structure of the deep network is adopted. We aim to verify the effectiveness of superpixel segmentation and hierarchical residual structure in improving the classification performance, in line with the training samples selected for the previous experiments, 10% of IP and UP datasets are selected, while 5% of SV dataset is selected, thus comparing the various evaluation results (OA, AA, Kappa, and standard deviation results) of the model without superpixel segmentation, the model without hierarchical residual structure, and the MSAC model. The compared results are shown in Fig. 17. It can be seen that both the superpixel segmentation and the hierarchical residual structure are obviously beneficial to improve the classification performance of MSAC on UP and SV datasets relative to IP dataset. The standard deviations of OA, AA, and Kappa of MSAC model were lower than those of the model without superpixel segmentation and without hierarchical residual structure, except for IP dataset, indicating that superpixel segmentation and hierarchical residual structure not only improved the classification accuracy but also enhanced the model stability. However, the model without superpixel segmentation has about the same classification effect compared with MSAC on IP dataset, which may be due to the fact that the hierarchical residual results can better extract the deep features of IP dataset. Overall, each part of MSAC is extremely helpful for the robustness and generalization of model to fully explore the spatial features and deep information of HSI.

H. Algorithms Applied to Practice and Analysis

Unlike the three conventional datasets mentioned above, we verify the practicality of the proposed MSAC method in practical situations with the HSI data acquired by the GF-5 satellite. GF-5 is the world’s first full-spectrum hyperspectral satellite to achieve integrated observation of the atmosphere and land, with a spatial resolution of 30 m and six payloads, which can acquire visible to short-wave optical infrared spectra (400–2500 nm) [46]. The HSI data of the Dongting Lake region acquired by GF-5 on January 22, 2019 were selected as the experimental dataset. The visible NIR spectral bands in the GF-5 raw data were filtered and other preprocessing operations were performed to obtain data containing 310 spectral bands with a size of
Fig. 18. Classification maps obtained by the different tested methods with ten samples per class for the GF-5 satellite dataset (OAs are in parentheses).

456 × 352. The reference data contains six classes with a total of 4816 labeled samples.

In order to verify the superiority of the model and the fairness of the comparison experiment, in this experiment, we randomly select ten samples from each class as training samples, and a total of 60 samples are used as the training set. In the same way, 60 samples are selected as the validation set, and the remaining samples are used as the test set. All the comparison methods that take the same parameters are the original. The classification results of the comparison method and the proposed method in Dongting Lake region are illustrated in Fig. 18.

MSAC improves the classification accuracy OA by 8.38% compared to SVM, which proves that the proposed method has a significant advantage over the traditional machine-learning method. Among the CNN-based models, SSRN shows better classification results than traditional 1D-CNN, 2D-CNN, and 3D-CNN, where 3D-CNN performs poorly. In contrast, MSAC achieves the best classification performance with a 2.74% and 5.81% increase in classification performance compared with recent advance algorithms of SSRN and CGCNN, respectively, which verifies the effectiveness of convolutional operations and superpixel maps to form adaptive convolutions and avoid transitional smoothing of classification maps. Compared with other methods, this not only demonstrates the effect of multiple superpixel segmentation scales on the convolution kernel, but likewise shows that the hierarchical residual structure at different superpixel segmentation scales is decisive for adapting to different spatial features of different classes of objects.

V. CONCLUSION

Due to the inherent defects of convolution kernels with fixed shapes, most HSI classification methods based on CNNs easily misclassify pixels in cross-class edges. In this article, an adaptive convolution-based HSI classification network is proposed. The convolution kernel is adaptively adjusted according to the HSI superpixel space features, and this adaptive convolution can suppress irrelevant noise information, extract boundary information across class regions, and reduce oversmoothing of classification maps. The hierarchical residual structure formed by multiple AConvUs expands feature learning in the receptive field, reduces gradient disappearance, and enhances robustness. A single-scale adaptive convolution algorithm based on adaptive convolution and adaptive convolution is proposed to extract spectral space features with more stable parameters. As a result, superpixels at different scales can form an adaptive convolutional network structure that exhibits stronger performance and can better extract the details of HSI classification maps. In the future works, we will investigate new adaptive convolution schemes to enhance HSI data modeling in deep structures and improve the flexibility of existing neural models.

REFERENCES

[1] M. Zucco, V. Caricato, A. Egidi, and M. Pisani, “A hyperspectral camera in the UVA band,” IEEE Trans. Instrum. Meas., vol. 64, no. 6, pp. 1425–1430, Jun. 2015.

[2] S. Ali, T. Glass, B. Parr, J. Potgieter, and F. Alam, “Low cost sensor with IoT Lorawan connectivity and machine learning-based calibration for air pollution monitoring,” IEEE Trans. Instrum. Meas., vol. 70, 2021, Art. no. 5500511.

[3] K. Gu, Y. Zhang, and J. Qiao, “Random forest ensemble for river turbidity measurement from space remote sensing data,” IEEE Trans. Instrum. Meas., vol. 69, no. 11, pp. 9028–9036, Nov. 2020.

[4] X. Wei, W. Li, M. Zhang, and Q. Li, “Medical hyperspectral image classification based on end-to-end fusion deep neural network,” IEEE Trans. Instrum. Meas., vol. 68, no. 11, pp. 4481–4492, Nov. 2019.

[5] Q. Hao et al., “Fusing multiple deep models for in vivo human brain hyperspectral image classification to identify glioblastoma tumor,” IEEE Trans. Instrum. Meas., vol. 70, 2021, Art. no. 4007314.

[6] K. Zhou et al., “Assessment of spectral variation between rice canopy components using spectral feature analysis of near-ground hyperspectral imaging data,” in Proc. 8th Workshop Hyperspectral Image Signal Process.: Evol. Remote Sens., 2016, pp. 1–4.

[7] S. Jakob, R. Zimmermann, and R. Glaouguen, “Processing of drone-borne hyperspectral data for geological applications,” in Proc. 8th Workshop Hyperspectral Image Signal Process.: Evol. Remote Sens., 2016, pp. 1–5.

[8] W. Li, C. Chen, H. Su, and Q. Du, “Local binary patterns and extreme learning machine for hyperspectral imagery classification,” IEEE Trans. Geosci. Remote Sens., vol. 53, no. 7, pp. 3681–3693, Jul. 2015.
et al. received the M.S. degree

\[ \text{IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.} \]

IEEE J. Sel. Topics Appl. Earth Observ. Remote

Cybern., to be published (doi: 10.1109/TCYB.2021.3080304).

Multiscale adaptive convolution for hyperspectral image classification

Qi Ren (Student Member, IEEE) received the B.S. degree in computer science and technology from the Shandong University, Datong, China, in 2020. He is currently working toward the M.S. degree in information and communication engineering from the Hunan Institute of Science and Technology, Yueyang, China.

His research interests include image processing, hyperspectral image classification, and manifold learning.

Bing Tu (Member, IEEE) received the M.S. degree in control science and engineering from the Guilin University of Technology, Guilin, China, in 2009, and the Ph.D. degree in mechatronic engineering from the Beijing University of Technology, Beijing, China, in 2013.

From 2015 to 2016, he was a Visiting Researcher with the Department of Computer Science and Engineering, University of Nevada, Reno, NV, USA, which is supported by the China Scholarship Council. Since 2018, he has been an Associate Professor with the School of Information Science and Engineering, Hunan Institute of Science and Technology, Yueyang, China. His research interests include sparse representation, pattern recognition, and analysis in remote sensing.
Qianming Li (Member, IEEE) received the M.S. degree in urban planning and the Ph.D. degree in civil architecture and planning design, from Central South University, Changsha, China, in 2016 and 2021, respectively. Since 2021, she has been a Lecturer with the School of Information Science and Engineering, Hunan Institute of Science and Technology, Yueyang, China. Her research interests include image processing, regional spatial structure, and urban analysis in remote sensing.

Wangquan He (Student Member, IEEE) received the B.S. degree in communication engineering, in 2019, from the Hunan Institute of Science and Technology, Yueyang, China, where he is currently working toward the M.S. degree in information and communication engineering. His research interests include pattern recognition, hyperspectral image classification, remote sensing image scene classification, and noisy label detection.

Yishu Peng (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from Northeastern University, Shenyang, China, in 2009, 2011, and 2017 respectively. From 2017 to 2019, he was with the School of Mechanical Engineering, Hunan Institute of Science and Technology, Yueyang, China, where he is currently with the School of Information Science and Technology. His research interests include image processing, object detection, and target tracing.