scientific reports

OPEN

Asymmetric coordinate attention spectral-spatial feature fusion network for hyperspectral image classification

Shuli Cheng, Liejun Wang ✉ & Anyu Du

In recent years, the hyperspectral classification algorithm based on deep learning has received widespread attention, but the existing network models have higher model complexity and require more time consumption. In order to further improve the accuracy of hyperspectral image classification and reduce model complexity, this paper proposes an asymmetric coordinate attention spectral-spatial feature fusion network (ACAS2F2N) to capture distinguishing hyperspectral features. Specifically, adaptive asymmetric iterative attention was proposed to obtain the discriminative spectral-spatial features. Different from the common feature fusion method, this feature fusion method can adapt to most skip connection tasks. In addition, there is no manual parameter setting. Coordinate attention is used to obtain accurate coordinate information and channel relationship. The strip pooling module was introduced to increase the network's receptive field and avoid irrelevant information brought by conventional convolution kernels. The proposed algorithm is tested on the mainstream hyperspectral datasets (IP, KSC, and Botswana), experimental results show that the proposed ACAS2F2N can achieve state-of-the-art performance with lower time complexity.

Hyperspectral image data has rich spectral-spatial information, and hyperspectral image classification tasks have important research significance, such as crop detection, geological prospecting and other fields[1]. However, the inherent characteristics of hyperspectral data (spatial pixel non-uniformity, spectral noise and frequency band correlation) affect the performance of hyperspectral image classification to a certain extent. Make full use of hyperspectral-spatial data to improve the performance of hyperspectral image classification, which has become a research hotspot in hyperspectral data analysis and processing[2–5]. At the same time, the processing methods of hyperspectral image classification tasks include traditional methods[6,7] and deep learning methods[8,9].

In recent years, deep learning-based methods have become the mainstream algorithm for hyperspectral image classification. Specifically, in terms of spectral-spatial feature extraction, Roy et al.[2] proposed a residual squeeze-and-excitation to extract joint spectral-spatial features. Tang et al.[3] introduced octave convolution to capture features with strong discriminative ability. Tang et al.[4] developed an improved residual network, which has a spectral-spatial kernel function and receptive field of different scales (A2S2K-ResNet). Xu et al.[5] studied the fully convolutional network based on dense conditional random fields and atrous convolution to deal with hyperspectral classification. Roy et al.[8] explored hybrid spectral CNN (HybridSN) by combining 3D CNN and 2D CNN. Guo et al.[9] proposed feature-grouped network, the method includes spatial residual block and spectral residual block. He et al.[10] used transfer learning to obtain hyperspectral image representation. Gao et al.[11] explored the multi-scale residual network based on mixed convolution to obtain fusion features. Lu et al.[12] presented a multi-scale residual network based on channel and spatial attention to perform hyperspectral image classification. Li et al.[13] proposed a double-branch dual-attention (DBDA), which includes spectral attention and spatial attention. Zhong et al.[14] explored the spectral–spatial residual network (SSRN) based on 3D convolutional layers. Wang et al.[15] studied fast dense spectral–spatial convolution (FDSSC), which is a densely-connected structure to learn hyperspectral features. Ge et al. merged 2D CNN and 3D CNN, and proposed a hyperspectral classification method based on multi-branch feature fusion[16]. Lee et al.[17] proposed a deeper contextual CNN (ContextNet), which can enrich the spectral-spatial features of hyperspectral. Paoletti et al.[18] studied the deep pyramidal residual network (DPyResNet), which focuses on the depth of the network block and the generation of more feature maps.

College of Information Science and Engineering, Xinjiang University, Urumqi 830046, China. ✉email: wljxju@xju.edu.cn

The construction of the network model has become the main technology of hyperspectral image classification, and it is also the current research focus of hyperspectral classification. In terms of graph convolutional networks, Mou et al.[19] proposed a nonlocal graph convolutional network (GCN). Wan et al.[20,21] explored multi-scale dynamic GCN based on superpixel segmentation of simple linear iterative clustering (SLIC). Yang et al. proposed GCN hyperspectral classification by sampling and aggregating, referred to as GraphSAGE[22]. Liu et al.[23] studied GCN hyperspectral classification based on label consistency and multi-scale convolutional networks. Ding et al.[24] proposed a globally consistent GCN based on SLIC. Hong et al. fuse CNN features and GCN features to perform hyperspectral classification[25]. Sha et al. proposed Graph Attention Network based on KNN and attention layer[26]. Zhao et al. explored the spectral-spatial GAT to reserve the discriminative features of hyperspectral[27]. In short, GCN plays an important role in the classification of hyperspectral images.

Summary of current research: For hyperspectral classification tasks, researchers have explored many classic algorithms, and these studies have further improved the accuracy of hyperspectral image classification. The relevant research summary is described as follows:

(1) *Network model* Residual networks and densely connected networks are the main network types for current hyperspectral classification. The fusion of 2D CNN and 3D CNN is also a trend in hyperspectral research. Inserting the network module into the backbone network (such as the residual network) may improve the performance of hyperspectral classification. However, it will increase model parameters and consume time complexity.

(2) *Multi-feature fusion* Hyperspectral image data has rich spectral-spatial features, so multi-feature fusion is the main method of hyperspectral classification. Related technologies include: multi-scale fusion, multi-branch fusion, spectral-spatial fusion, etc. The adaptability and effectiveness of feature fusion still need to be considered, which can avoid manual parameter adjustment and improve the accuracy of hyperspectral classification.

(3) *Attention mechanism* The attention mechanism can effectively enhance hyperspectral features and select discriminative spectral-spatial features. In computer vision, most of the existing attention mechanisms have high complexity, and there is less exploration of low-rank attention networks. This is also very important to expand the receptive field of the network and choose a suitable attention model.

The main contributions of this paper. Compared with most existing hyperspectral classification baselines, in order to further improve the accuracy of hyperspectral image classification and reduce model complexity, this paper proposes a new network model to deal with hyperspectral classification tasks. The algorithm is called ACAS2F2N. The main contributions of this paper are summarized as follows:

(1) This paper proposes an asymmetric coordinate attention spectral-spatial feature fusion network (ACAS2F2N) to complete the hyperspectral classification task. ACAS2F2N is an asymmetric learning model, and it is an end-to-end feature learning method. The proposed algorithm can improve the performance of hyperspectral image classification and has lower model complexity.

(2) The adaptive iterative attention feature fusion method is adopted to extract the discriminative spectral-spatial features. Different from the common feature fusion method, this feature fusion method can adapt to most skip connection tasks. In addition, there is no manual parameter setting.

(3) Coordinate attention is used to obtain accurate coordinate information and channel relationship. The strip pooling module was introduced to increase the network's receptive field and avoid irrelevant information brought by conventional convolution kernels.

(4) The proposed algorithm is tested on the mainstream hyperspectral datasets (IP, KSC, and Botswana), experimental results show that the proposed ACAS2F2N can achieve state-of-the-art performance with lower time complexity.

The rest of the paper are organized as follows. "Related work" section describes the proposed ACAS2F2N network architecture. "ACAS2F2N network architecture" section shows the experimental results. "Results" section elaborates the conclusion.

Related work

In hyperspectral classification tasks, generative adversarial network[28–30], long short-term memory[31], network architecture search[32], and capsule network[33–35] are all used in hyperspectral classification. In addition, Hao et al. proposed a hyperspectral classification algorithm based on recurrent neural network and geometry-aware loss, referred to as Geo-DRNN[36]. In the attention mechanism, Xue et al. proposed a second-order covariance pooling network based on attention, which reduces model complexity while ensuring classification accuracy[37]. Gao et al. combined 2D CNN and 3D CNN to construct a deep covariance attention network[38]. Gao et al. proposed densely connected multiscale attention network by combining multi-scale technology and densely connected structure[39]. Hang et al. explored attention-aided CNN based on the spectral attention module and the spatial attention module, and completed hyperspectral classification based on feature fusion[40]. Li et al.[41] proposed a multi-attention fusion network by fusing the band attention module and the spatial attention module. Xi et al. extracted multi-stream hyperspectral features, integrated corresponding features, and introduced a fully connected layer to perform hyperspectral classification[42]. Xi et al. explored hybrid residual attention. In the feature extraction process, 3D CNN features, 2D CNN features, and 1D CNN features are acquired to make corresponding decisions[43]. The attention mechanism is also a key technology to improve the performance of hyperspectral classification.

The construction of the network model is still the main technology of hyperspectral classification. Specifically, Sun et al. proposed[44] a fully convolutional segmentation network, and its core structure is still a residual block. Pan et al.[45] presented a semantic segmentation network to reduce the adjustment of manual parameters of the model. Transfer learning is also one of the research contents of hyperspectral feature extraction[46,47]. GhostNet module and channel attention are selected to improve the accuracy of hyperspectral classification[48]. A nonlocal module and a fully convolutional network were chosen to improve the effectiveness of hyperspectral classification[49]. Li et al. proposed a multi-layer fusion dense network based on the 3D dense module to extract spectral-spatial features[50]. Xie et al. proposed a multiscale densely-connected fusion network based on the multi-scale fusion and dense module for hyperspectral image classification[51]. Meng et al. developed a mixed link network based on addition, concatenation and dense modules[52]. Wei et al. proposed an unsupervised feature learning method, and the loss function is based on classification loss and clustering loss[53]. Zhang et al. fuse low-level residual blocks, medium-level residual blocks and high-level residual blocks to build a deep feature integration network[54]. Li et al. explored a two-stream convolutional network to obtain global and local features[55]. Yuan et al. proposed a proxy-based deep learning framework to learn Hyperspectral features[56]. Zheng et al. mixed 2D CNN and 3D CNN, and introduced covariance pooling to reduce the channel dimension[57]. Jiang et al. carried out related research on hyperspectral classification in spatial consistence[58]. Mu et al. separated low-rank components and sparse components, and implemented a two-branch network[59]. The convolutional layer and residual module still play an important role in hyperspectral classification[60–64]. In addition, support vector machines[65], self-learning[66], multi-view feature[67] are used to obtain discriminative features to perform hyperspectral classification. Some methods based on RNN and LSTM are also widely used in the field of histopathology[68–70]. The goal of these methods is to obtain highly discriminative spectral-spatial fusion features. In short, hyperspectral classification has become a research hotspot in the field of remote sensing and related research is of great significance.

ACAS2F2N network architecture

Research motivation.

At present, there are still some problems in hyperspectral classification that need to be dealt with: (1) hyperspectral images have rich spectral and spatial information. The spatial and spectral information of hyperspectral images are not fully utilized; (2) most existing hyperspectral classifications have high model complexity. In other words, the network model has more parameters.

Although many algorithms based on attention mechanism have been proposed in the field of computer vision, if the attention mechanism is directly inserted into the backbone network, the time complexity of the algorithm will inevitably increase. In addition, directly fusing two independent networks will also increase algorithm time complexity and model complexity. Based on the above considerations, the goal of this paper is to independently build a hyperspectral classification network. The proposed network can achieve better hyperspectral classification performance with lower model complexity.

Algorithm advantages.

In order to further improve the accuracy of hyperspectral image classification and reduce the time loss of hyperspectral classification, this paper proposes asymmetric coordinate attention spectral-spatial feature fusion network (ACAS2F2N). The proposed algorithm has the following major advantages:

(1) Different from most existing algorithms, this paper does not use residual network and dense connection module to obtain hyperspectral image features, so the proposed algorithm does not increase the model complexity.

(2) Coordinate attention is used to extract discriminative spectral-spatial features. Different from existing hyperspectral classification algorithms, coordinate attention can accurately obtain coordinate information and channel relationships. In addition, coordinate attention is also proposed for the first time in hyperspectral processing.

(3) Strip pooling increases the network receptive field and optimizes the feature map. Strip kernel function can avoid the introduction of irrelevant information.

(4) Asymmetric iterative attention feature fusion can obtain multi-scale information. This feature fusion method conveniently uses skip connection information, so it has better scalability. In addition, there is no manual parameter setting.

ACAS2F2N overall architecture. The overall architecture of the proposed ACAS2F2N is shown in Fig. 1. The goal of ACAS2F2N is to complete the classification of hyperspectral images and reduce the time consumption of the model. In Fig. 1, the basic steps of ACAS2F2N are as follows: (1) obtain the feature map (A). Specifically, for a given input hyperspectral data, the size of the hyperspectral image is $C \times M \times N$, the neighborhood feature map ($\boldsymbol{A}$) is obtained based on neighborhood extraction. The size of feature map ($\boldsymbol{A}$) is $C \times M \times N$; (2) coordinate attention is used to accurately capture the position information and channel relationship in the feature map (A). Specifically, the spectral-spatial information in the hyperspectral data is captured based on coordinate attention. After coordinate attention, ACAS2F2N can obtain the feature map ($\boldsymbol{B}$). The size of the feature map ($\boldsymbol{B}$) is $C \times M \times N$; (3) establish long-term connections, eliminate irrelevant information and increase the receptive field. The strip pooling module converts feature map ($\boldsymbol{B}$) to feature map ($\boldsymbol{C}$). Different from the traditional convolution kernel function, strip pooling adopts a narrow-band kernel to avoid the negative influence of irrelevant information. In addition, the strip pooling module has the ability to establish long-term connections and increase the receptive field of ACAS2F2N; (4) adaptive asymmetric iterative attention feature fusion module (A2IAFFM). The main function of A2IAFFM is to adaptively fuse feature map ($\boldsymbol{B}$) and feature map

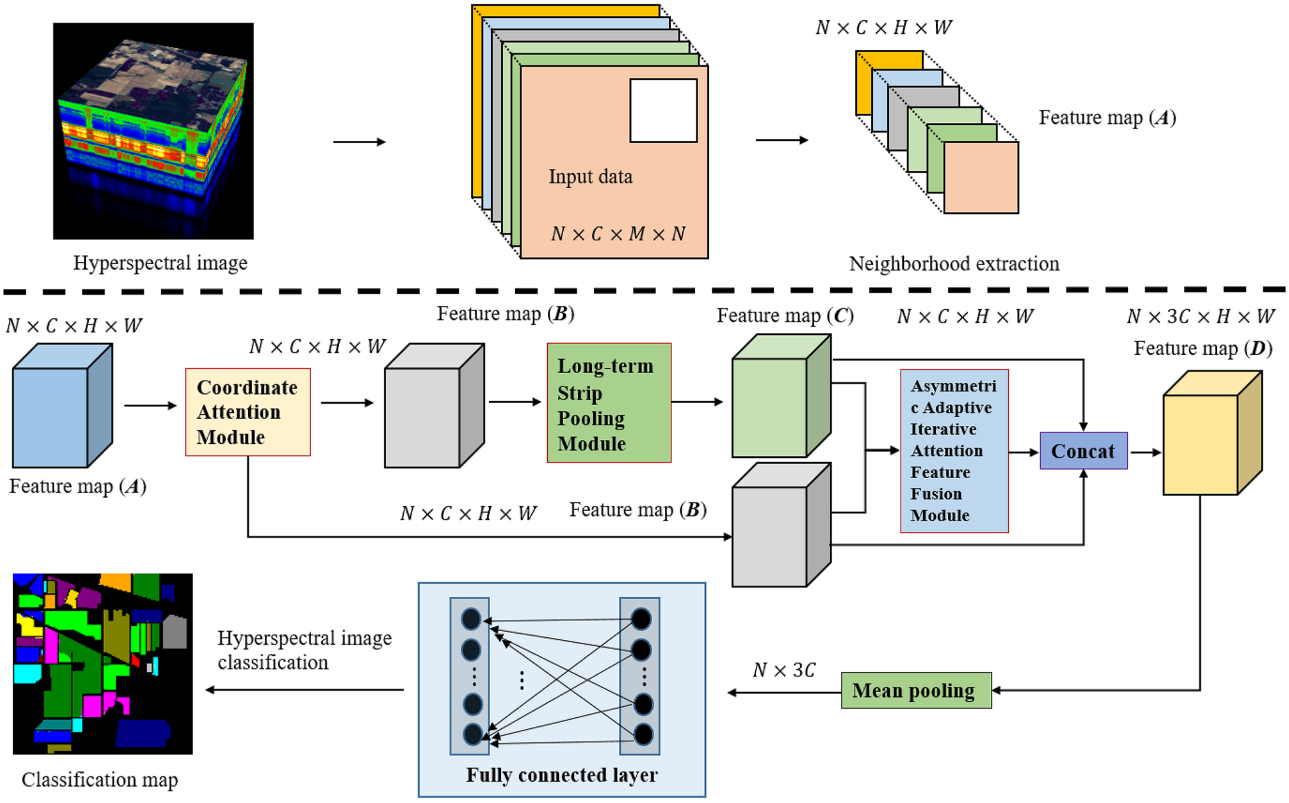

Hyperspectral image
Input data
$N \times C \times M \times N$
$N \times C \times H \times W$
Feature map (*A*)
Neighborhood extraction
$N \times C \times H \times W$
Feature map (*A*)
Coordinate Attention Module
$N \times C \times H \times W$
Feature map (*B*)
Long-term Strip Pooling Module
Feature map (*C*)
$N \times C \times H \times W$
Asymmetric Adaptive Iterative Attention Feature Fusion Module
Concat
$N \times 3C \times H \times W$
Feature map (*D*)
$N \times C \times H \times W$
Feature map (*B*)
Classification map
Hyperspectral image classification
Fully connected layer
$N \times 3C$
Mean pooling

Figure 1. The overall architecture of the proposed ACAS2F2N.

(*C*). The feature map acquisition process is an asymmetric way, so it is named A2IAFFM. Through A2IAFFM, ACAS2F2N can obtain spectral-spatial features with discriminative capabilities. In addition, compared to most existing feature fusion modules, A2IAFFM does not require additional manual parameter settings while acquiring multi-scale information. After A2IAFFM, the feature map (*D*) is obtained, and the size of the feature map is $C \times M \times N$; (5) to complete hyperspectral image classification based on mean pooling and fully connected layer. The size of the mean pooling output is $N \times 3C$. Finally, the hyperspectral classification task is completed through the fully connected layer and cross entropy loss.

The working principle of ACAS2F2N is as follows: (1) obtain feature map (A); (2) coordinate attention is used to obtain accurate coordinate information and channel relationship; (3) strip pooling module was introduced to increase the network's receptive field and avoid irrelevant information brought by conventional convolution kernels; (4) the adaptive iterative attention feature fusion method is adopted to extract the discriminative spectral-spatial features; (5) complete hyperspectral image classification based on mean pooling and fully connected layers.

In Fig. 1, the definition of symbols is described as follows: N is the batch size, C is the channel size, and C is the hyperspectral band size. $M \times N$ is the size of the hyperspectral image spatial domain, M is the height of the hyperspectral image spatial domain, and N is the width of the hyperspectral image spatial domain. $H \times W$ is the size of the feature map, H is the height of the feature map, and W is the width of the feature map.

Coordinate attention module. Existing hyperspectral image classification algorithms, location information (hyperspectral spatial domain) and channel relationship (hyperspectral image frequency band) have received less attention. Hyperspectral images have rich spectral-spatial information, so coordinate attention module[71] is used in this paper to obtain accurate position information and channel relationships of hyperspectral images.

Channel attention pays less attention to location information, and spatial attention pays less attention to channel relationships. The mixed-domain attention mechanism can consider location information and channel information at the same time. The mixed-domain attention mechanism is widely used in computer vision, most of the existing attention mechanism models have high computational complexity[72,73]. In addition, related studies have shown that low-rank attention and lightweight attention are less studied in computer vision[74,75].

In order to obtain the precise location information of the hyperspectral, we use coordinate attention to extract the feature map (*B*). The coordinate attention module mainly captures position information, and the structure of the coordinate attention module is shown in Fig. 2.

In Fig. 2, the coordinate attention module only encodes H and W. In hyperspectral image, given position (i,j), the pixel value on channel c is $x_c(i,j)$.

The output of W mean pooling is defined as follows:

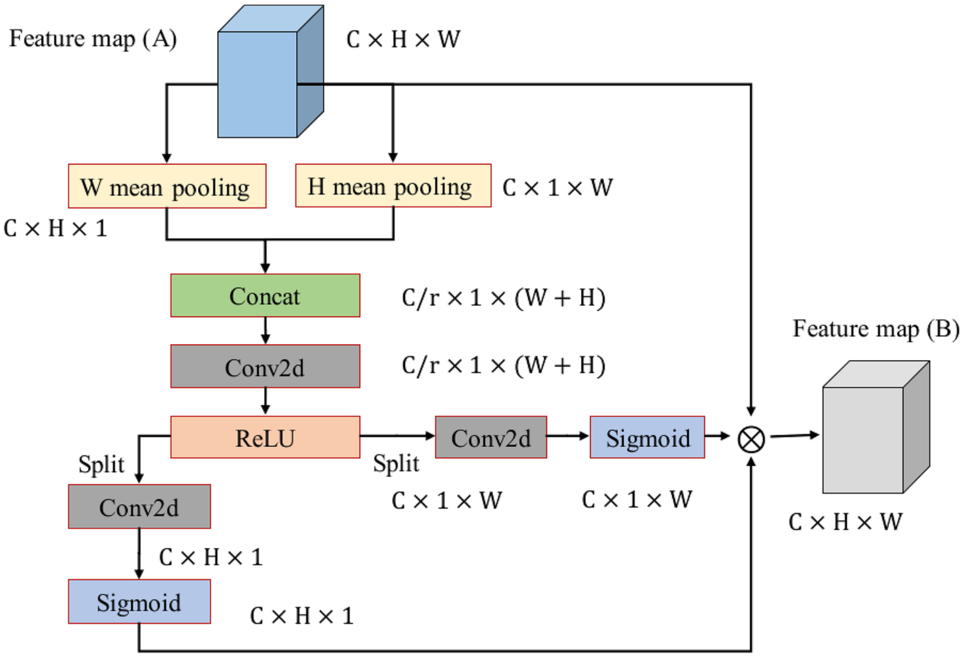


Figure 2. Coordinate attention module. The coordinate attention module can accurately capture the coordinate information of hyperspectral images.

$$y_c^i(i) = \frac{1}{W}\sum\nolimits_{0\leq j\leq W} x_c(i,j). \tag{1}$$

The output of H mean pooling is defined as follows:

$$y_c^j(j) = \frac{1}{H}\sum\nolimits_{0\leq j\leq H} x_c(i,j). \tag{2}$$

In Fig. 2, the coordinate attention module then completes the concatenate, convolution, and activation function operations. The relevant definitions are as follows:

$$y = \delta\left(F\left(\left[y_c^i, y_c^j\right]\right)\right), \tag{3}$$

where [,] is concatenate, F is 1 × 1 convolution operation, and δ is ReLU activation function. y is the output feature map of the ReLU layer.

After the split operation, y can be decomposed into y^i and y^j. Next, y^i completes the weighting of W through convolution and activation function. y^j completes the weighting of H through convolution and activation function. The relevant definitions are as follows:

$$w^i = \sigma\left(F_i\left(y^i\right)\right), \tag{4}$$

$$w^j = \sigma\left(F_j\left(y^j\right)\right), \tag{5}$$

where F_i is the convolution operation on H, and its input is y^i.

w^i is the adaptive weighting of the H direction of the hyperspectral data. F_j is the convolution operation on W, and its input is y^j. w^j is the adaptive weighting of the W direction of the hyperspectral data. σ is the sigmoid activation function.

The output feature map ($\boldsymbol{B}$) of the coordinate attention module is defined as follows:

$$f_c(i,j) = x_c(i,j) \times w_c^i(i) \times w_c^j(j), \tag{6}$$

where c is the c-th channel. $w_c^i(i)$ is the weight of the i-th position in the H direction. $w_c^j(j)$ is the weight of the j-th position in the W direction. Given position (i,j), $x_c(i,j)$ is the value of the input feature map ($\boldsymbol{A}$). $f_c(i,j)$ is the value of the output feature map ($\boldsymbol{B}$).

Long-term strip pooling module. Different from the traditional kernel function, the strip pooling module uses a narrow-band kernel function to enhance the network receptive field. In addition, the strip pooling module can obtain long-term hyperspectral information and avoid the influence of negative information in hyperspectral

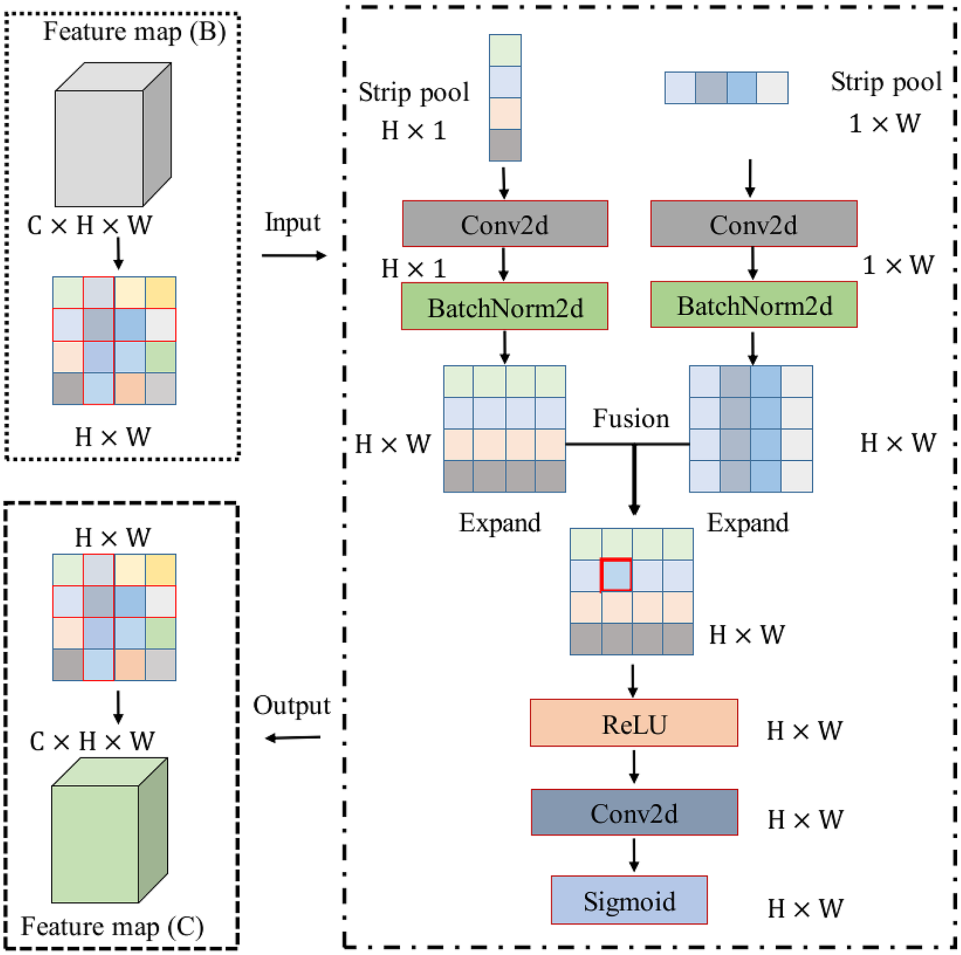


Figure 3. Long-term strip pooling module. The narrowband kernel function is used to expand the network receptive field. Compared with the traditional kernel function, the strip pooling module can better capture the region information of the hyperspectral image.

images. In Fig. 3, strip pooling uses a narrow-band kernel function to focus on the regional information of the hyperspectral image.

The input of the long-term strip pooling module is the feature map (***B***), namely $f_c(i,j)$. The definition of strip pooling is as follows:

The output of W strip pooling is defined as follows:

$$g_c^i(i) = \frac{1}{\mathrm{W}}\sum\nolimits_{0\le j\le \mathrm{W}} f_c(i,j). \tag{7}$$

The output of H strip pooling is defined as follows:

$$g_c^j(j) = \frac{1}{\mathrm{H}}\sum\nolimits_{0\le j\le \mathrm{H}} f_c(i,j). \tag{8}$$

Next, the strip pooling module executes Conv2d and Batchnorm2d. The relevant definitions are as follows:

$$h^i = BN\left(M_i\left(g_c^i(i)\right)\right), \tag{9}$$

$$h^j = BN\left(M_j\left(g_c^j(j)\right)\right), \tag{10}$$

where M_i is the conv2d operation, and its kernel function size is $1 \times W$. M_j is the Conv2d operation, and its kernel function size is $H \times 1$. BN is a Batchnorm2d operation. h^i and h^j are the output of the Batchnorm2d layer, respectively. c is the c-th channel.

The related definitions of feature fusion, ReLU, Conv2d, and activation function are as follows:

$$N = \sigma\left(f_2\left(f_1\left(h^i + h^j\right)\right)\right), \tag{11}$$

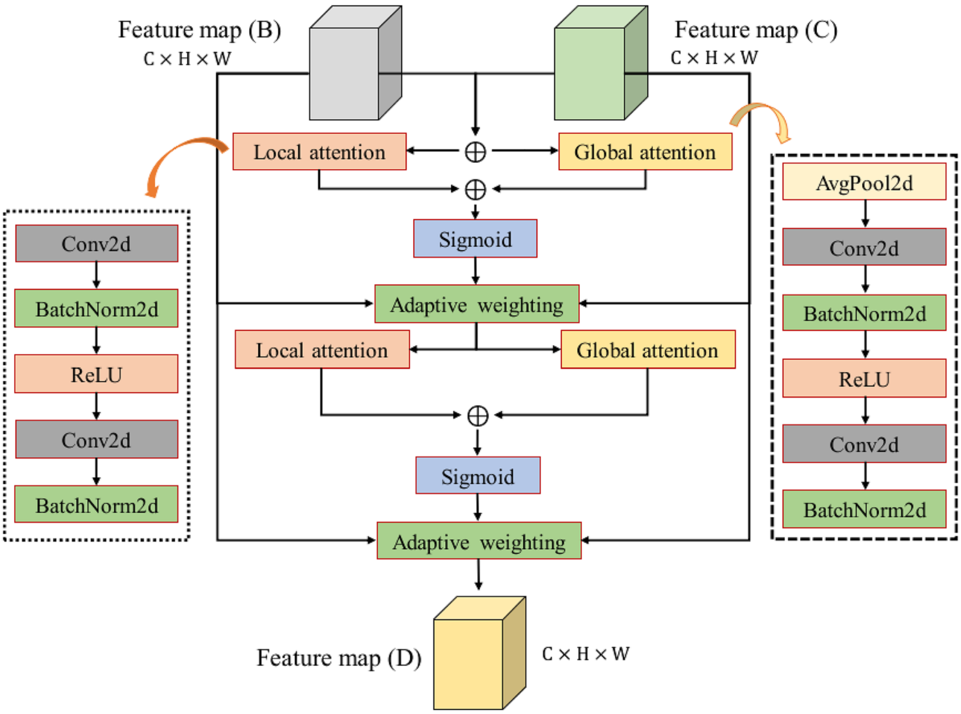


Figure 4. Asymmetric adaptive iterative attention feature fusion module. The input feature maps are B and C, and the output feature map is D.

where f_1 is the ReLU operation, f_2 is the Conv2d operation, and σ is the sigmoid function. N is the feature map ($\boldsymbol{C}$), and N is also the output of the long-term strip pooling module.

Asymmetric adaptive iterative attention feature fusion module (A2IAFFM). In Fig. 4, The goal of A2IAFFM is to complete the fusion of feature map ($\boldsymbol{B}$) and feature map ($\boldsymbol{C}$). A2IAFFM can acquire hyperspectral features with discriminative ability. The output of A2IAFFM is the feature map ($\boldsymbol{D}$). A2IAFFM includes local attention, global attention and adaptive weighting.

To simplify the description, L represents the local attention function, and G represents the global attention function. σ is the sigmoid function. The adaptive weighting is defined as follows:

$$Z = W \otimes X + (1 - W) \otimes Y, \tag{12}$$

where W is the output weight of sigmoid function, and X and Y are the two input feature maps. $\otimes$ is element-wise product.

The first weight calculation process of A2IAFFM is defined as follows:

$$W_1 = \sigma(L(\boldsymbol{B} + \boldsymbol{C}) + G(\boldsymbol{B} + \boldsymbol{C})), \tag{13}$$

where the input feature maps are B and C, and the output is W_1.

The A2IAFFM first adaptive weighting calculation process is defined as follows:

$$Z_1 = W_1 \otimes \boldsymbol{B} + (1 - W_1) \otimes \boldsymbol{C}. \tag{14}$$

The second weight calculation process of A2IAFFM is defined as follows:

$$W_2 = \sigma(L(Z_1) + G(Z_1)). \tag{15}$$

The A2IAFFM second adaptive weighting calculation process is defined as follows:

$$Z_2 = W_2 \otimes \boldsymbol{B} + (1 - W_2) \otimes \boldsymbol{C}. \tag{16}$$

In formula (16), the output feature map (D) of A2IAFFM can be obtained, denoted by Z_2.

Other related technical modules. As shown in Fig. 1, other related technology modules mainly explain the details of the mean pooling layer and the fully connected layer. Among them, the average pooling layer is based on the AdaptiveAvgPool2d operation. There are two fully connected layers. The number of input nodes is the number of frequency bands of the feature map, and the number of output nodes is the number of categories of hyperspectral classification. In addition, this paper uses cross-entropy loss to optimize the network parameters of proposed ACAS2F2N.

Network module	Network layer	Parameter configuration
Coordinate attention module	*W* mean pooling	Output_size = (1, 9)
	H mean pooling	Output_size = (9, 1)
	Conv2d (3 layers)	Kernel_size = (1, 1), stride = (1, 1)
Strip pooling module	ConvH (Conv2d)	Kernel_size = (3, 1), stride = (1, 1), padding = (1, 0)
	ConvW (Conv2d)	Kernel_size = (1, 3), stride = (1, 1), padding = (0, 1)
	Conv2d (1 layers)	Kernel_size = (1, 1), stride = (1, 1)
A2IAFFM	Conv2d (8 layers)	Kernel_size = (1, 1), stride = (1, 1)
Fully connected layer	FC	Input size = 3 × C, output size = N

Table 1. Parameter configuration of the proposed network.

Parameter configuration. Figure 1 shows the overall architecture of the proposed ACAS2F2N, while Figs. 2, 3 and 4 show the network structures of the coordinate attention module, strip pooling module, and A2IAFFM, respectively. Next, the parameter configuration of the proposed algorithm is shown in Table 1. In the coordinate attention module, its core contains 2 mean pooling layers (*W* mean pooling and *H* mean pooling) and 3 convolutional layers (Conv2d). In the strip pooling module, the module contains a total of 3 convolutional layers, namely ConvH, ConvW and Conv2d. In A2IAFFM, the module contains 8 convolutional layers (Conv2d). The three modules coordinate attention module, strip pooling module and A2IAFFM obtain the hyperspectral feature map (***D***) through concat operation. At this time, the number of channels of ***D*** is 3*C*. Therefore, the input of the fully connected layer (FC) is *3C*, and the output of the fully connected layer (FC) is the number of categories.

Results

In order to evaluate the effectiveness of the proposed ACAS2F2N in hyperspectral classification tasks, this paper shows experimental comparison and analysis. Specifically, this paper mainly focuses on dataset and experimental parameter settings, baseline comparison algorithm, algorithm performance comparison, ablation analysis, algorithm performance and complexity analysis.

Dataset and experimental parameter settings. Indian Pines (IP) was collected by the Airborne Visual Infrared Imaging Spectrometer (AVIRIS) in 1992. IP contains a total of 220 continuous bands with a wavelength range of 0.4–2.5 µm. Among them, 20 invalid bands are not considered as research objects, so the experiment contains a total of 200 valid bands. The size of the data is 145 × 145.

The Botswana data set contains a total of 14 categories with a resolution of 1476 × 256 and a total of 145 effective frequency bands.

Kennedy Space Center (KSC) acquisition is based on AVIRIS. A total of 176 available bands are included, and the wavelength range is 0.4 to 2.5 µm. The size of the data is 512 × 614. Spatial resolution is 18 m.

The experiment was performed on tesla V100, and the capacity of the graphics card was 16G. The evaluation indicators are overall accuracy (OA), average accuracy (AA), and kappa coefficient (Kappa). During the experiment, on the three datasets, the ratio of training samples, verification samples and test samples is 3%:3%:94%.

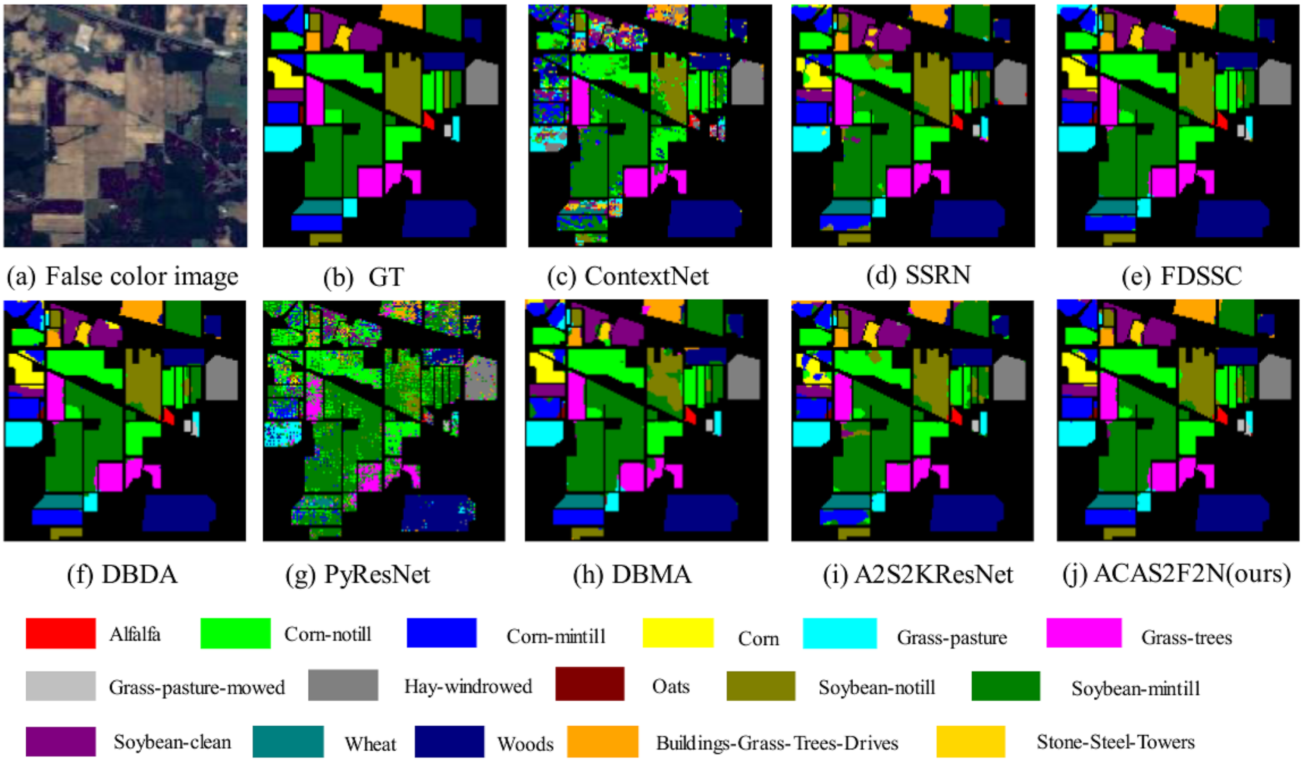
Baseline selection. In order to verify the classification accuracy and real-time performance of the proposed ACAS2F2N, this paper compares the performance of 7 baselines. The specific description is as follows: A2S2KResNet[4] (TGRS, 2020), DBDA[13] (RS, 2020), PyResNet[18] (TGRS2019), DBMA[76] (RS, 2019), SSRN[14] (TGRS, 2018), FDSSC[15] (RS, 2018), ContextNet[17] (TIP, 2017).

Algorithm performance comparison. In order to evaluate algorithm performance fairly, all algorithms maintain the same parameter settings. On the IP dataset, 3% of the data is used for training. The epoch is 200, the code is run 3 times, and the length of the patch is 4. The experimental results on IP are shown in Table 2 and Fig. 5. It can be seen from Table 2 and Fig. 5 that the performance of the proposed ACAS2F2N algorithm is better than other baselines, and the time complexity of the proposed ACAS2F2N is lower. Therefore, it can show high classification accuracy and low time complexity.

Specifically, in Table 2, A2S2KResNet is proposed in TGRS 2020. The OA, AA, Kappa, training time and test time of A2S2KResNet are 92.34, 90.23, 0.9125, 233.65 and 24.25, respectively. The OA, AA, Kappa, training time, and test time of the proposed ACAS2F2N are 96.02, 92.28, 0.9546, 99.40, and 20.45, respectively. Therefore, compared with A2S2KResNet, the proposed ACAS2F2N OA increased by 3.99%, AA increased by 2.27%, and Kappa increased by 4.61%. In terms of time complexity, the training time was reduced from the original 233.65 s to 99.40 s, and the test time was reduced from the original 24.25 s to 20.45 s. Therefore, the proposed ACAS2F2N has faster convergence. Compared with other baselines, proposed ACAS2F2N can accurately capture position information and enhance the receptive field. In addition, the proposed algorithm uses strip pooling to mine regional information. Proposed ACAS2F2N achieves better classification performance than all the baselines in Table 2 when using 3% data to train the network. In Fig. 5, classification map of the proposed ACAS2F2N is closer to Ground truth (GT), so the experiment verifies the effectiveness of ACAS2F2N on IP.

Class	Training	Test	ContextNet	SSRN	FDSSC	DBDA	PyResNet	DBMA	A2S2KResNet	Ours
1	3	43	48.57 ± 0.107	94.20 ± 0.08	95.49 ± 0.03	94.26 ± 0.04	37.61 ± 0.15	97.83 ± 0.03	86.35 ± 0.07	94.04 ± 0.03
2	42	1337	76.51 ± 0.04	91.17 ± 0.02	97.78 ± 0.01	94.91 ± 0.04	40.70 ± 0.10	95.12 ± 0.02	92.94 ± 0.03	95.61 ± 0.02
3	24	777	53.93 ± 0.11	90.05 ± 0.09	95.65 ± 0.02	92.00 ± 0.04	38.94 ± 0.03	97.31 ± 0.02	90.78 ± 0.03	95.65 ± 0.02
4	7	224	44.46 ± 0.30	91.81 ± 0.08	93.67 ± 0.04	89.22 ± 0.10	28.95 ± 0.18	90.74 ± 0.05	90.52 ± 0.09	94.83 ± 0.03
5	14	455	70.28 ± 0.12	96.96 ± 0.01	94.03 ± 0.08	96.93 ± 0.01	67.95 ± 0.10	96.50 ± 0.04	99.08 ± 0.01	99.15 ± 0.08
6	21	689	91.90 ± 0.03	98.13 ± 0.00	96.48 ± 0.01	97.49 ± 0.01	63.44 ± 0.11	97.26 ± 0.02	95.73 ± 0.03	97.06 ± 0.02
7	3	25	26.51 ± 0.05	66.67 ± 0.47	66.35 ± 0.09	70.90 ± 0.01	52.56 ± 0.41	86.48 ± 0.05	86.04 ± 0.20	65.64 ± 0.11
8	14	447	85.27 ± 0.13	96.95 ± 0.01	99.35 ± 0.01	100.0 ± 0.21	61.26 ± 0.43	100.0 ± 0.00	97.81 ± 0.01	100.0 ± 0.00
9	3	16	13.70 ± 0.08	23.81 ± 0.34	54.89 ± 0.15	69.21 ± 0.17	14.44 ± 0.14	62.89 ± 0.23	58.06 ± 0.13	66.18 ± 0.01
10	29	918	81.39 ± 0.02	73.56 ± 0.06	88.82 ± 0.08	93.13 ± 0.03	48.30 ± 0.18	92.37 ± 0.05	86.54 ± 0.03	92.09 ± 0.03
11	73	2313	79.20 ± 0.03	95.35 ± 0.03	97.69 ± 0.01	96.03 ± 0.00	53.19 ± 0.12	92.37 ± 0.06	91.19 ± 0.02	96.75 ± 0.01
12	17	563	51.12 ± 0.07	89.64 ± 0.03	98.12 ± 0.01	96.63 ± 0.01	52.86 ± 0.05	94.43 ± 0.03	94.30 ± 0.04	95.57 ± 0.02
13	6	193	68.89 ± 0.06	94.43 ± 0.04	92.94 ± 0.02	98.90 ± 0.01	44.91 ± 0.32	99.66 ± 0.00	97.01 ± 0.01	97.77 ± 0.01
14	37	1184	90.52 ± 0.02	93.78 ± 0.02	96.94 ± 0.03	96.24 ± 0.02	78.18 ± 0.09	96.81 ± 0.02	96.47 ± 0.02	97.72 ± 0.02
15	11	367	53.88 ± 0.13	87.03 ± 0.13	96.94 ± 0.01	96.01 ± 0.02	45.10 ± 0.23	94.88 ± 0.01	86.18 ± 0.09	95.21 ± 0.01
16	3	84	38.73 ± 0.20	66.67 ± 0.47	81.45 ± 0.20	97.86 ± 0.01	45.45 ± 0.41	94.21 ± 0.07	94.69 ± 0.02	93.13 ± 0.06
OA (%)	307	9635	76.08 ± 0.03	90.70 ± 0.01	95.43 ± 0.01	95.28 ± 0.01	52.80 ± 0.11	94.48 ± 0.02	92.34 ± 0.00	96.02 ± 0.00
AA (%)			60.93 ± 0.06	84.38 ± 0.08	90.41 ± 0.02	92.48 ± 0.02	48.37 ± 0.17	93.05 ± 0.01	90.23 ± 0.03	92.28 ± 0.01
Kappa			0.7241 ± 0.04	0.8941 ± 0.02	0.9479 ± 0.01	0.9461 ± 0.01	0.4427 ± 0.14	0.9368 ± 0.02	0.9125 ± 0.00	0.9546 ± 0.01
Training time			166.24	236.73	578.67	338.06	507.81	252.64	233.65	99.40
Test time			49.29	37.21	23.93	26.77	103.94	22.34	24.25	20.45

Table 2. The objective indicators on IP dataset (using 3% training, 3% verification, 94% testing).


(a) False color image (b) GT (c) ContextNet (d) SSRN (e) FDSSC

(f) DBDA (g) PyResNet (h) DBMA (i) A2S2KResNet (j) ACAS2F2N(ours)

Figure 5. Hyperspectral classification maps on IP. (**a**) False color image. (**b**) Ground truth. (**c**–**i**) Performance of all baselines. (**j**) Performance of the proposed ACAS2F2N. The other content is color and label information.

Class	Training	Test	ContextNet	SSRN	FDSSC	DBDA	PyResNet	DBMA	A2S2KResNet	Ours
1	22	709	98.25 ± 0.00	99.58 ± 0.01	99.86 ± 0.00	100.0 ± 0.00	87.82 ± 0.10	99.95 ± 0.00	99.63 ± 0.00	99.86 ± 0.00
2	7	232	79.70 ± 0.12	83.42 ± 0.12	96.40 ± 0.05	97.71 ± 0.03	69.57 ± 0.21	96.74 ± 0.05	98.22 ± 0.02	97.10 ± 0.04
3	7	242	63.18 ± 0.05	85.60 ± 0.11	93.23 ± 0.05	99.58 ± 0.01	76.57 ± 0.18	90.89 ± 0.03	87.61 ± 0.08	92.41 ± 0.07
4	7	237	64.27 ± 0.00	71.03 ± 0.13	95.10 ± 0.06	94.52 ± 0.07	74.70 ± 0.19	85.43 ± 0.09	97.24 ± 0.03	98.99 ± 0.01
5	4	156	60.91 ± 0.07	78.38 ± 0.21	88.38 ± 0.09	84.38 ± 0.11	45.84 ± 0.32	95.58 ± 0.04	83.79 ± 0.12	94.76 ± 0.04
6	6	212	73.22 ± 0.19	88.09 ± 0.06	94.82 ± 0.07	96.47 ± 0.03	64.04 ± 0.17	98.94 ± 0.01	92.12 ± 0.06	99.69 ± 0.00
7	3	99	60.11 ± 0.10	50.57 ± 0.41	88.96 ± 0.16	97.46 ± 0.04	57.17 ± 0.33	93.72 ± 0.09	91.93 ± 0.06	100.0 ± 0.00
8	12	410	95.68 ± 0.05	99.83 ± 0.00	100.0 ± 0.00	98.34 ± 0.02	95.14 ± 0.02	99.59 ± 0.01	99.52 ± 0.00	99.67 ± 0.00
9	15	491	93.77 ± 0.07	97.19 ± 0.02	100.0 ± 0.00	100.0 ± 0.00	85.87 ± 0.06	99.93 ± 0.00	99.46 ± 0.01	99.86 ± 0.00
10	12	385	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	99.91 ± 0.00	99.74 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
11	12	399	94.07 ± 0.08	97.98 ± 0.03	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
12	15	470	98.57 ± 0.02	95.31 ± 0.01	98.28 ± 0.01	95.42 ± 0.01	96.67 ± 0.01	94.65 ± 0.02	96.29 ± 0.03	97.34 ± 0.01
13	27	871	99.96 ± 0.00	99.96 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	94.03 ± 0.07	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
OA (%)	149	4913	90.08 ± 0.02	93.91 ± 0.01	97.97 ± 0.00	98.13 ± 0.00	85.83 ± 0.03	97.60 ± 0.00	97.30 ± 0.00	98.86 ± 0.00
AA (%)			83.21 ± 0.01	88.22 ± 0.04	96.54 ± 0.01	97.21 ± 0.00	80.55 ± 0.04	96.57 ± 0.01	95.83 ± 0.00	98.44 ± 0.00
Kappa			0.8896 ± 0.02	0.9322 ± 0.01	0.9774 ± 0.00	0.9792 ± 0.00	0.8419 ± 0.03	0.9733 ± 0.00	0.9699 ± 0.00	0.9873 ± 0.00
Training time			129.90	282.96	774.31	451.19	251.09	433.50	305.02	117.55
Test time			34.03	24.18	28.87	40.90	101.62	49.63	18.50	15.80

Table 3. The objective indicators on KSC dataset (using 3% training, 3% verification, 94% testing).

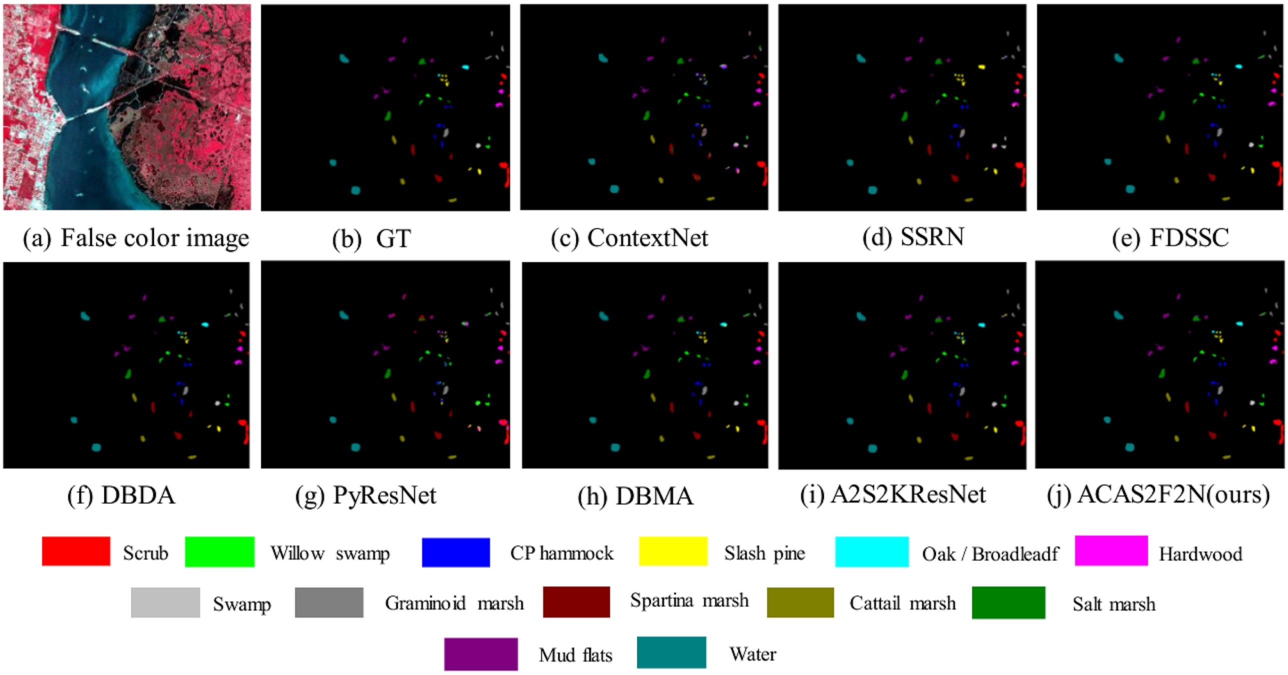

(a) False color image (b) GT (c) ContextNet (d) SSRN (e) FDSSC
(f) DBDA (g) PyResNet (h) DBMA (i) A2S2KResNet (j) ACAS2F2N(ours)

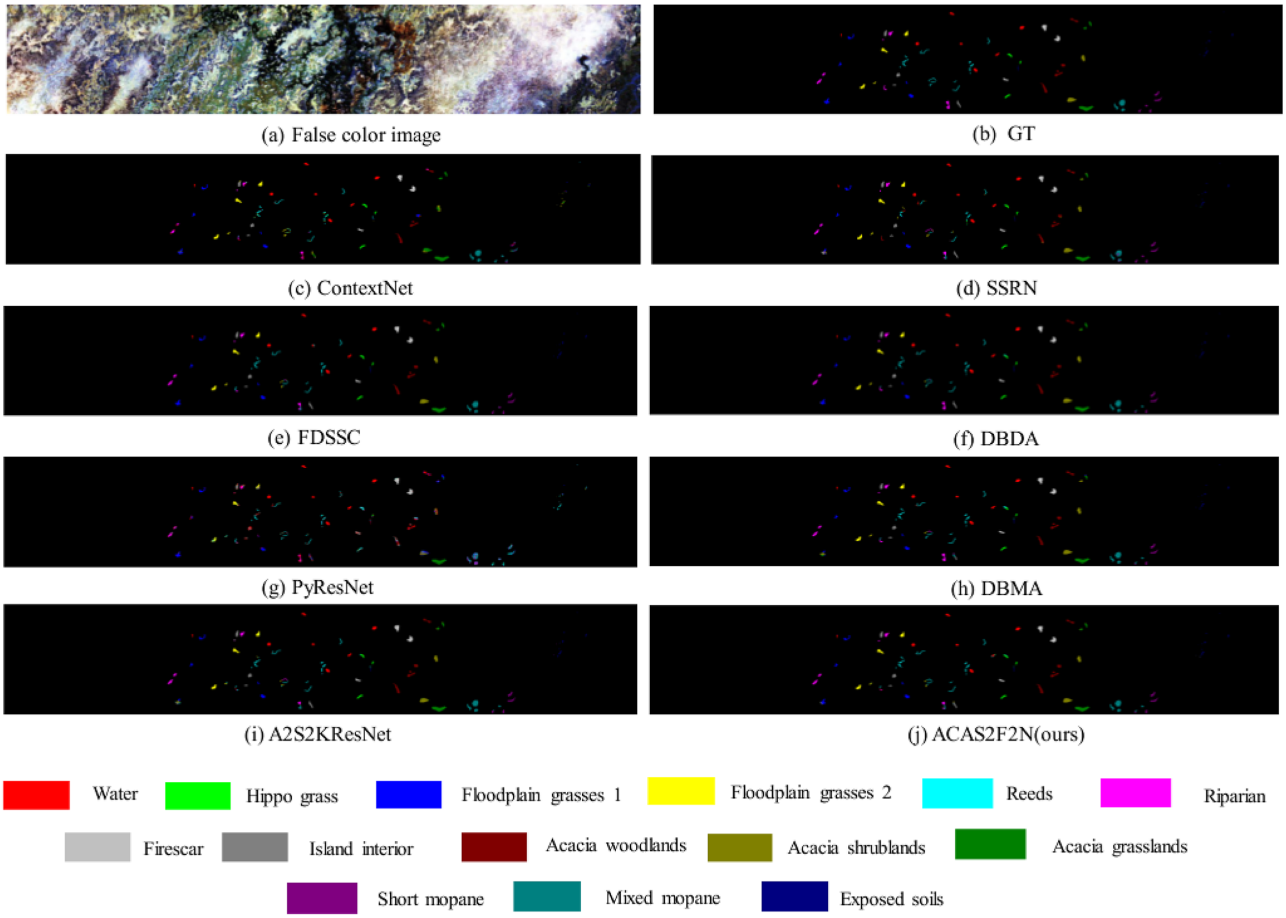
Figure 6. Hyperspectral classification maps on KSC. (**a**) False color image. (**b**) Ground truth. (**c**–**i**) Performance of all baselines. (**j**) Performance of the proposed ACAS2F2N. The other content is color and label information.

On the KSC dataset, Table 3 and Fig. 6 respectively show the objective and subjective hyperspectral classification results. The OA, AA, Kappa, training time and test time of A2S2KResNet are 97.30, 95.83, 0.9699, 305.02 and 18.5, respectively. The OA, AA, Kappa, training time, and test time of the proposed ACAS2F2N are 98.86, 98.44, 0.9873, 117.55, and 15.80, respectively. Therefore, compared with A2S2KResNet, the proposed ACAS2F2N OA increased by 1.60%, AA increased by 2.72%, and Kappa increased by 1.79%. In terms of time complexity, the training time was reduced from the original 305.02 s to 117.55 s, and the test time was reduced from the original 18.5 s to 15.8 s. Therefore, the proposed ACAS2F2N has faster convergence on KSC.

On the Botswana dataset, Table 4 and Fig. 7 respectively show the objective and subjective hyperspectral classification results. A2S2KResNet was proposed and published in TGRS2020. The OA, AA, Kappa, training time and test time of A2S2KResNet are 91.62, 93.37, 0.9091, 47.53, and 8.92, respectively. The OA, AA, Kappa, training time, and test time of the proposed ACAS2F2N are 94.63, 95.72, 0.9418, 50.29, and 7.09, respectively.

Class	Training	Test	ContextNet	SSRN	FDSSC	DBDA	PyResNet	DBMA	A2S2KResNet	Ours
1	8	254	98.97 ± 0.01	88.88 ± 0.08	98.01 ± 0.02	98.45 ± 0.01	63.78 ± 0.22	95.56 ± 0.03	97.21 ± 0.01	94.36 ± 0.07
2	3	97	76.57 ± 0.08	93.47 ± 0.05	100.0 ± 0.00	100.0 ± 0.00	91.68 ± 0.07	98.99 ± 0.01	98.98 ± 0.01	97.89 ± 0.01
3	7	243	92.27 ± 0.05	99.18 ± 0.01	100.0 ± 0.00	100.0 ± 0.00	60.56 ± 0.24	98.67 ± 0.01	97.70 ± 0.03	99.56 ± 0.01
4	6	203	89.22 ± 0.09	83.11 ± 0.04	92.16 ± 0.02	95.42 ± 0.04	94.79 ± 0.06	93.76 ± 0.03	94.14 ± 0.03	89.91 ± 0.06
5	8	254	73.97 ± 0.16	95.77 ± 0.04	83.24 ± 0.07	89.49 ± 0.02	41.43 ± 0.13	89.09 ± 0.06	85.11 ± 0.05	87.79 ± 0.06
6	8	253	87.75 ± 0.05	87.39 ± 0.04	90.05 ± 0.09	96.64 ± 0.03	48.55 ± 0.12	92.64 ± 0.04	76.53 ± 0.09	98.59 ± 0.01
7	7	248	96.63 ± 0.04	98.93 ± 0.01	100.0 ± 0.00	100.0 ± 0.00	94.09 ± 0.01	99.86 ± 0.00	97.98 ± 0.01	99.87 ± 0.00
8	6	193	95.07 ± 0.01	93.53 ± 0.09	99.82 ± 0.00	98.63 ± 0.01	86.25 ± 0.14	94.48 ± 0.05	96.14 ± 0.03	98.26 ± 0.02
9	9	297	84.62 ± 0.12	81.94 ± 0.09	95.96 ± 0.03	99.56 ± 0.01	78.90 ± 0.10	97.86 ± 0.02	86.06 ± 0.06	89.49 ± 0.08
10	7	229	78.91 ± 0.22	84.19 ± 0.11	99.31 ± 0.01	98.91 ± 0.02	85.99 ± 0.03	91.42 ± 0.10	93.73 ± 0.08	97.62 ± 0.03
11	9	283	78.68 ± 0.14	96.16 ± 0.04	100.0 ± 0.00	100.0 ± 0.00	89.11 ± 0.06	96.74 ± 0.03	97.18 ± 0.02	100.0 ± 0.00
12	5	166	81.34 ± 0.26	90.40 ± 0.07	98.60 ± 0.01	99.46 ± 0.01	88.23 ± 0.06	98.10 ± 0.03	98.90 ± 0.01	100.0 ± 0.00
13	8	252	68.08 ± 0.06	84.86 ± 0.08	90.83 ± 0.07	90.77 ± 0.07	77.68 ± 0.09	88.53 ± 0.08	87.47 ± 0.10	86.70 ± 0.09
14	3	90	88.52 ± 0.16	98.43 ± 0.02	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
OA (%)	94	3060	82.57 ± 0.03	89.19 ± 0.01	95.47 ± 0.01	97.25 ± 0.01	63.05 ± 0.06	94.55 ± 0.02	91.62 ± 0.01	94.63 ± 0.00
AA (%)			85.04 ± 0.01	91.16 ± 0.01	96.28 ± 0.01	97.67 ± 0.01	78.65 ± 0.03	95.41 ± 0.01	93.37 ± 0.01	95.72 ± 0.00
Kappa			0.8107 ± 0.04	0.8829 ± 0.01	0.9509 ± 0.01	0.9702 ± 0.01	0.5981 ± 0.07	0.9409 ± 0.02	0.9091 ± 0.01	0.9418 ± 0.00
Training time			40.53	90.40	179.32	146.67	120.34	72.09	47.53	50.29
Test time			10.74	8.88	8.82	10.85	22.96	6.75	8.92	7.09

Table 4. The objective indicators on Botswana (using 3% training, 3% verification, 94% testing).


(a) False color image

(b) GT

(c) ContextNet

(d) SSRN

(e) FDSSC

(f) DBDA

(g) PyResNet

(h) DBMA

(i) A2S2KResNet

(j) ACAS2F2N(ours)

Figure 7. Hyperspectral classification maps on Botswana. (**a**) False color image. (**b**) Ground truth. (**c–i**) Performance of all baselines. (**j**) Performance of the proposed ACAS2F2N. The other content is color and label information.

Network module			Feature fusion method		Algorithm performance		
CAM	SPM	A2IAFFM	Concat	Add	OA (%)	AA (%)	Kappa
√	√	√	√		96.02	92.28	0.9546
√		√	√		93.23	89.05	0.9229
	√	√	√		93.39	89.43	0.9246
√	√	√		√	93.26	89.63	0.9232
√					92.80	88.37	0.9180
	√				93.31	89.91	0.9238

Table 5. Ablation analysis on IP dataset.

Network module			Feature fusion method		Algorithm performance		
CAM	SPM	A2IAFFM	Concat	Add	OA (%)	AA (%)	Kappa
√	√	√	√		98.86	98.44	0.9873
√		√	√		97.99	98.28	0.9777
	√	√	√		97.63	96.78	0.9736
√	√	√		√	97.52	97.05	0.9724
√					97.60	96.87	0.9332
	√				95.60	95.57	0.9509

Table 6. Ablation analysis on KSC dataset.

Therefore, compared with A2S2KResNet, the proposed ACAS2F2N OA increased by 3.28%, AA increased by 2.52%, and Kappa increased by 3.60%. In terms of time complexity, the test time was reduced from the original 8.92 s to 7.09 s. Therefore, the proposed ACAS2F2N has faster convergence on the Botswana dataset.

In Table 4, among the 7 baseline algorithms, the performance of DBDA and FDSSC algorithms (OA, AA and Kappa) is better than the performance of the proposed ACAS2F2N algorithm. However, the model complexity of DBDA and FDSSC algorithm is higher, and the model time consumption is much greater than the proposed ACAS2F2N algorithm. Comprehensively comparing Tables 2 and 3, the performance of the proposed algorithm is better than that of DBDA and FDSSC.

Algorithm performance comparison summary. Tables 2, 3 and 4 are comparisons of objective indicators of algorithms (OA, AA, Kappa, training time, and test time) on the three datasets (IP, KSC and Botswana). Figures 5, 6 and 7 are the results of algorithm classification maps, which respectively illustrate the results of hyperspectral classification and classification color labels. The experiment selects 7 baseline algorithms for performance comparison, of which 7 baselines are A2S2KResNet[4] (TGRS, 2020), DBDA[13] (RS, 2020), PyResNet[18] (TGRS2019), DBMA[76] (RS, 2019), SSRN[14] (TGRS, 2018), FDSSC[15] (RS, 2018), and ContextNet[17] (TIP, 2017). Experimental results verify the effectiveness and real-time performance of the proposed ACAS2F2N. The proposed ACAS2F2N has lower model complexity while improving the performance of hyperspectral classification.

In terms of specific indicators (OA, AA, Kappa), compared to A2S2KResNet (TGRS 2020), the proposed ACAS2F2N OA increased by 3.99%, AA increased by 2.27%, and Kappa increased by 4.61% on IP dataset; the proposed ACAS2F2N OA increased by 1.60%, AA increased by 2.72%, and Kappa increased by 1.79% on KSC dataset; the proposed ACAS2F2N OA increased by 3.28%, AA increased by 2.52%, and Kappa increased by 3.60% on Botswana dataset.

In terms of model complexity, compared to A2S2KResNet (TGRS 2020), the training time was reduced from the original 233.65 s to 99.40 s, and the test time was reduced from the original 24.25 s to 20.45 s on IP dataset; the training time was reduced from the original 305.02 s to 117.55 s, and the test time was reduced from the original 18.5 s to 15.8 s on KSC dataset; the training time was increased from the original 47.53 s to 50.29 s, and the test time was reduced from the original 8.92 s to 7.09 s on Botswana dataset.

Ablation analysis.

Next, this paper shows the ablation analysis of the algorithm. Among them, Table 5 is the ablation analysis on the IP data set, Table 6 is the ablation analysis on the KSC data set, and Table 7 is the ablation analysis on the Botswana data set. For the convenience of description, coordinate attention module is abbreviated as CAM, strip pooling module is abbreviated as SPM, and Asymmetric adaptive iterative attention feature fusion module is abbreviated as A2IAFFM.

The overall result of the algorithm in this paper is shown in Fig. 1. The network modules of ablation analysis include CAM, SPM and A2IAFFM. Feature fusion methods include concat and add operations. If the proposed algorithm removes the A2IAFFM module, then the algorithm only includes CAM or SPM network blocks, and the algorithm does not include feature fusion methods. The ablation analysis experiment is shown in Tables 5, 6 and 7. The experimental results show that the mentioned modules (CAM, SPM and A2IAFFM) all have a positive

Network module			Feature fusion method		Algorithm performance		
CAM	SPM	A2IAFFM	Concat	Add	OA (%)	AA (%)	Kappa
√	√	√	√		94.63	95.72	0.9418
√		√	√		90.21	92.19	0.8938
	√	√	√		89.92	92.08	0.8907
√	√	√		√	89.51	91.81	0.8863
√					89.53	91.49	0.8865
	√				89.72	91.82	0.8886

Table 7. Ablation analysis on Botswana dataset.

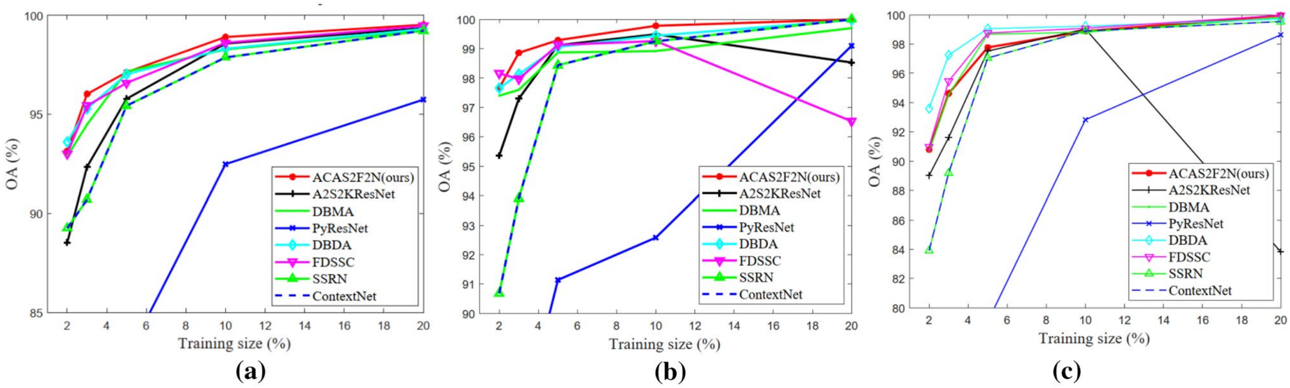


Figure 8. OA performance under different training sizes. (**a**) IP dataset. (**b**) KSC dataset. (**c**) Botswana dataset.

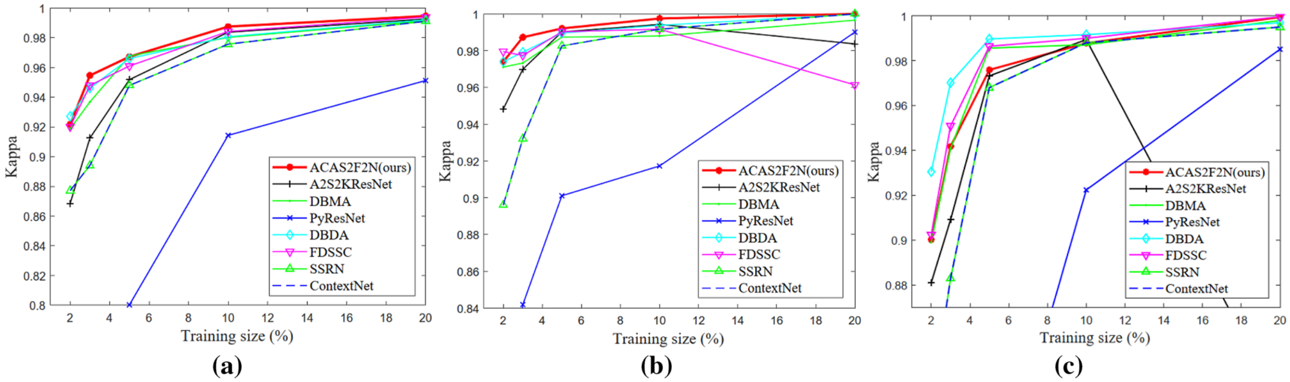


Figure 9. Kappa performance under different training sizes. (**a**) IP dataset. (**b**) KSC dataset. (**c**) Botswana dataset.

effect on the hyperspectral classification. In this paper, Concat is used to fuse the features of the three modules. At this time, the algorithm in this paper can experiment with the best hyperspectral classification performance.

The impact of training size. The number of training samples has a great influence for hyperspectral classification performance. Therefore, we analyze the impact of training size on OA, Kappa, Training time and test time. Figure 8 shows the OA results of different algorithms under different training sizes. As the training size increases, the OA of the proposed ACAS2F2N also increases. On the IP and KSC datasets, as the training size changes, the proposed ACAS2F2N both show the best performance. On the Botswana dataset, as the training size increases, the proposed ACAS2F2N hyperspectral classification accuracy also increases. Although the accuracy of the proposed algorithm is not optimal on the Botswana dataset, the algorithm has the lowest time complexity. In addition, the proposed algorithm is easier to extend. In addition, the proposed algorithm has the best performance on the IP and KSC datasets, and the proposed ACAS2F2N has less time complexity. Figure 9 shows Kappa performance under different training sizes. Figures 9 and 8 have similar conclusions and verify the effectiveness of the proposed ACAS2F2N.

Specifically, On the IP dataset, when the training size is 2%, 3%, 5%, 10%, and 20%, compared to the A2S2KResNet (TGRS 2020) algorithm, the OA performance of the proposed ACAS2F2N is improved by 5.21%,

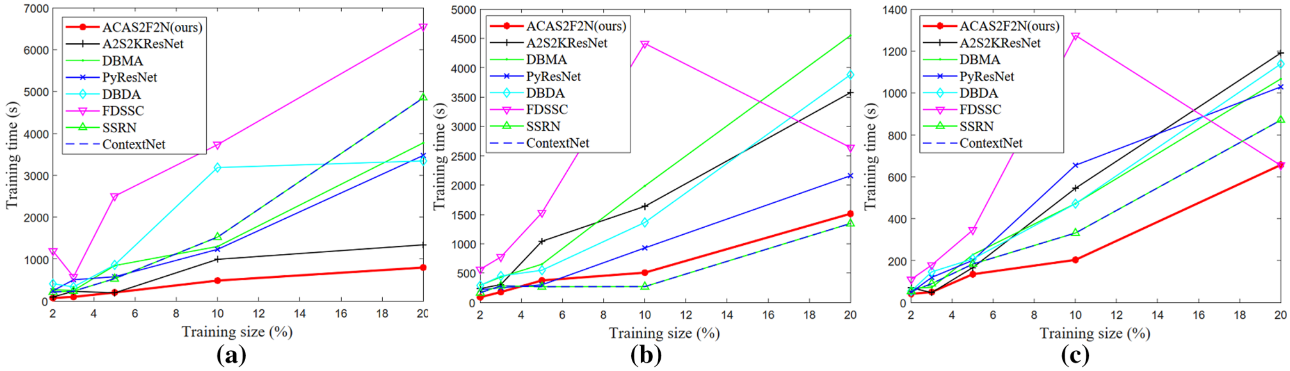


Figure 10. Training time under different training sizes. (**a**) IP dataset. (**b**) KSC dataset. (**c**) Botswana dataset.

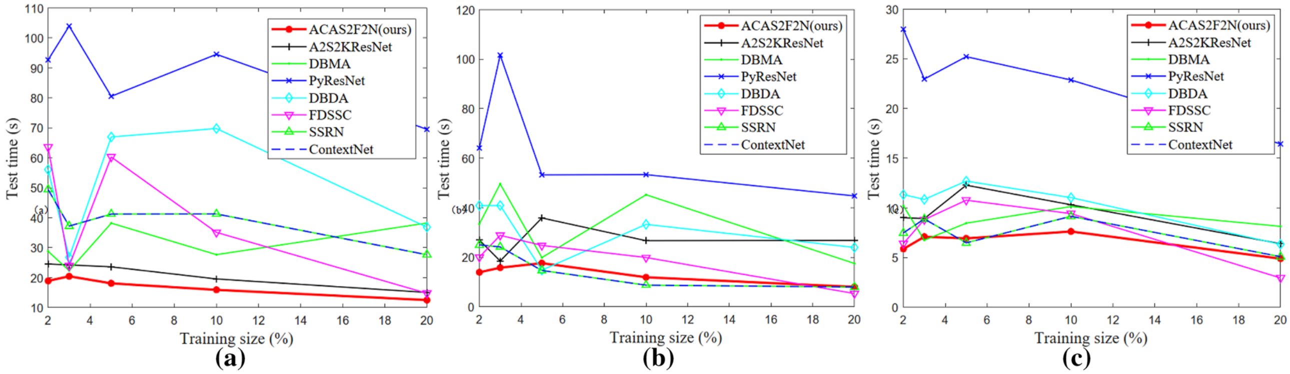


Figure 11. Test time under different training sizes. (**a**) IP dataset. (**b**) KSC dataset. (**c**) Botswana dataset.

3.99%, 1.40%, 0.33% and 0.17%, respectively. On the IP dataset, when the training size is 2%, 3%, 5%, 10%, and 20%, compared to the A2S2KResNet, the Kappa performance of the proposed ACAS2F2N is improved by 6.14%, 4.61%, 1.62%, 0.39% and 0.20%, respectively.

On the KSC dataset, when the training size is 2%, 3%, 5%, 10%, and 20%, compared to the A2S2KResNet (TGRS 2020) algorithm, the OA performance of the proposed ACAS2F2N is improved by 2.41%, 1.60%, 0.17%, 0.29% and 1.49%, respectively. On the KSC dataset, when the training size is 2%, 3%, 5%, 10%, and 20%, compared to the A2S2KResNet, the Kappa performance of the proposed ACAS2F2N is improved by 2.71%, 1.79%, 0.19%, 0.32% and 1.66%, respectively.

On the Botswana dataset, when the training size is 2%, 3%, 5%, 10%, and 20%, compared to the A2S2KResNet (TGRS 2020) algorithm, the OA performance of the proposed ACAS2F2N is improved by 1.99%, 3.29%, 0.26%, – 0.15% and 19.24%, respectively. On the Botswana dataset, when the training size is 2%, 3%, 5%, 10%, and 20%, compared to the A2S2KResNet, the Kappa performance of the proposed ACAS2F2N is improved by 2.18%, 3.60%, 0.28%, -0.16% and 20.93% respectively. The experimental results show the retrieval performance of the algorithm under different training sizes, and verify the effectiveness of the proposed ACAS2F2N.

Time consumption analysis.

Time consumption analysis demonstrates the convergence of the algorithm. Time consumption analysis is also a concrete manifestation of model complexity. This article gives the time consumption of all algorithms under different training sizes, including training time and test time.

Figure 10 shows Training time under different training sizes. Figure 11 shows Test time under different training sizes. Figures 10 and 11 jointly illustrate the time complexity of the proposed algorithm and the baseline. It can be seen from Figs. 10 and 11 that the proposed ACAS2F2N has less time consumption. In Figs. 8 and 9, we illustrate the impact of different training sizes on OA and Kappa. In terms of time complexity and accuracy of hyperspectral classification, the performance of the proposed ACAS2F2N is better than that of the baselines. Experiments verify the effectiveness of the proposed algorithm.

Training and verification accuracy.

Figure 12 shows the training and verification accuracy. Figure 12 shows that the performance of the proposed ACAS2F2N algorithm is better than that of A2S2KResNet. In short, the proposed ACAS2F2N shows better classification performance on three hyperspectral datasets. Achieving high classification accuracy is accompanied by lower time consumption.

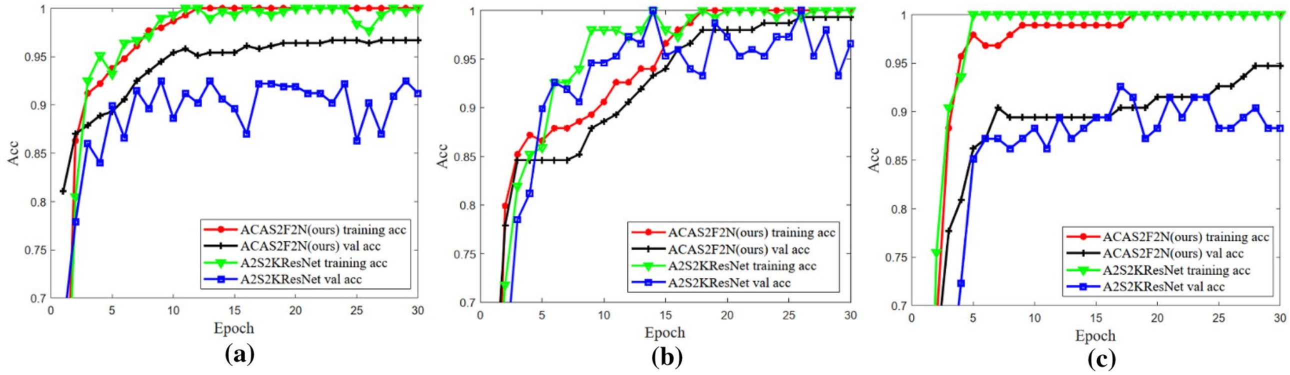


Figure 12. Training and verification accuracy. (**a**) IP dataset. (**b**) KSC dataset. (**c**) Botswana dataset.

Conclusions

In this paper, we propose an asymmetric coordinate attention spectral-spatial feature fusion network (ACAS2F2N) to complete the hyperspectral classification task. Compared with the baselines, the proposed algorithm can improve the performance of hyperspectral image classification and has lower model complexity. Specifically, Coordinate attention is used to obtain accurate coordinate information and channel relationship. The strip pooling module was introduced to increase the network's receptive field and avoid irrelevant information brought by conventional convolution kernels. Asymmetric adaptive iterative attention feature fusion module is adopted to extract the discriminative spectral-spatial features. The experimental was performed on three datasets (IP, KSC and Botswana), and the experimental results show that the performance of the proposed ACAS2F2N is better than the baselines. In addition, the proposed ACAS2F2N has low model complexity and time consumption.

Received: 19 May 2021; Accepted: 19 August 2021
Published online: 31 August 2021

References

1. Yu, C., Han, R., Song, M., Liu, C. & Chang, C.-I. Feedback attention-based dense CNN for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2021.3058549 (2021).
2. Roy, S. K., Chatterjee, S., Bhattacharyya, S., Chaudhuri, B. B. & Platoš, J. Lightweight spectral–spatial squeeze-and- excitation residual bag-of-features learning for hyperspectral classification. *IEEE Trans. Geosci. Remote Sens.* **58**(8), 5277–5290 (2020).
3. Tang, X. *et al.* Hyperspectral image classification based on 3-D octave convolution with spatial–spectral attention network. *IEEE Trans. Geosci. Remote Sens.* **59**(3), 2430–2447 (2021).
4. Roy, S. K., Manna, S., Song, T. & Bruzzone, L. Attention-based adaptive spectral-spatial kernel resnet for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2020.3043267 (2021).
5. Xu, Y., Du, B. & Zhang, L. Beyond the patchwise classification: Spectral-spatial fully convolutional networks for hyperspectral image classification. *IEEE Trans. Big Data* **6**(3), 492–506 (2020).
6. Hao, Q., S. Li, S., Fang, L. & Kang, X. Multiscale feature extraction with gaussian curvature filter for hyperspectral image classification. In *IGARSS 2020—2020 IEEE International Geoscience and Remote Sensing Symposium*, 80–83 (2020).
7. Chang, C.-I. *et al.* Self-mutual information-based band selection for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2020.3024602 (2021).
8. Roy, S. K., Krishna, G., Dubey, S. R. & Chaudhuri, B. B. HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **17**(2), 277–281 (2020).
9. Guo, W., Ye, H. & Cao, F. Feature-grouped network with spectral-spatial connected attention for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2021.3051056 (2021).
10. He, X., Chen, Y. & Ghamisi, P. Heterogeneous transfer learning for hyperspectral image classification based on convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **58**(5), 3246–3263 (2020).
11. Gao, H., Yang, Y., Li, C., Gao, L. & Zhang, B. Multiscale residual network with mixed depthwise convolution for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **59**(4), 3396–3408 (2021).
12. Lu, Z., Xu, B., Sun, L., Zhan, T. & Tang, S. 3-D channel and spatial attention based multiscale spatial–spectral residual network for hyperspectral image classification. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **13**, 4311–4324 (2020).
13. Li, R., Zheng, S., Duan, C., Yang, Y. & Wang, X. Classification of hyperspectral image based on double-branch dual-attention mechanism network. *Remote Sens.* https://doi.org/10.20944/preprints201912.0059.v2 (2020).
14. Zhong, Z., Li, J., Luo, Z. & Chapman, M. Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework. *IEEE Trans. Geosci. Remote Sens.* **56**(2), 847–858 (2018).
15. Wang, W., Dou, S., Jiang, Z. & Sun, L. A fast dense spectral-spatial convolution network framework for hyperspectral images classification. *Remote Sens.* **10**(7), 1068 (2018).
16. Ge, Z., Cao, G., Li, X. & Fu, P. Hyperspectral image classification method based on 2D–3D CNN and Multibranch Feature Fusion. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **13**, 5776–5788 (2020).
17. Lee, H. & Kwon, H. Going deeper with contextual CNN for hyperspectral image classification. *IEEE Trans. Image Process.* **26**(10), 4843–4855 (2017).
18. Paoletti, M. E. *et al.* Deep pyramidal residual networks for spectral–spatial hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **57**(2), 740–754 (2019).
19. Mou, L., Lu, X., Li, X. & Zhu, X. X. Nonlocal graph convolutional networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **58**(12), 8246–8257 (2020).
20. Wan, S. *et al.* Multiscale dynamic graph convolutional network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **58**(5), 3162–3177 (2020).
21. Wan, S. *et al.* Hyperspectral image classification with context-aware dynamic graph convolutional network. *IEEE Trans. Geosci. Remote Sens.* **59**(1), 597–612 (2021).

22. Yang, P. *et al.* Hyperspectral image classification with spectral and spatial graph using inductive representation learning network. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **14**, 791–800 (2021).
23. Liu, S. *et al.* DFL-LC: Deep feature learning with label consistencies for hyperspectral image classification. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **14**, 3669–3681 (2021).
24. Ding, Y., Guo, Y., Chong, Y., Pan, S. & Feng, J. Global consistent graph convolutional network for hyperspectral image classification. *IEEE Trans. Instrum. Meas.* **70**, 1–16 (2021).
25. Hong, D. *et al.* Graph convolutional networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2020.3015157 (2021).
26. Sha, A., Wang, B., Wu, X. & Zhang, L. Semisupervised classification for hyperspectral images using graph attention networks. *IEEE Geosci. Remote Sens. Lett.* **18**(1), 157–161 (2021).
27. Zhao, Z., Wang, H. & Yu, X. Spectral-spatial graph attention network for semisupervised hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* https://doi.org/10.1109/LGRS.2021.3059509 (2021).
28. Tao, C., Wang, H., Qi, J. & Li, H. Semisupervised variational generative adversarial networks for hyperspectral image classification. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **13**, 914–927 (2020).
29. Zhao, W., Chen, X., Bo, Y. & Chen, J. Semisupervised hyperspectral image classification with cluster-based conditional generative adversarial net. *IEEE Geosci. Remote Sens. Lett.* **17**(3), 539–543 (2020).
30. Roy, S. K., Haut, J. M., Paoletti, M. E., Dubey, S. R. & Plaza, A. Generative adversarial minority oversampling for spectral-spatial hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2021.3052048 (2021).
31. Hu, W. *et al.* Spatial–spectral feature extraction via deep ConvLSTM neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **58**(6), 4237–4250 (2020).
32. Wang, J. *et al.* NAS-guided lightweight multiscale attention fusion network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2021.3049377 (2021).
33. Li, H.-C. *et al.* Robust capsule network based on maximum correntropy criterion for hyperspectral image classification. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **13**, 738–751 (2020).
34. Wang, J. *et al.* Dual-channel capsule generation adversarial network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2020.3044312 (2021).
35. Xu, Q., Wang, D. & Luo, B. Faster multiscale capsule network with octave convolution for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **18**(2), 361–365 (2021).
36. Hao, S., Wang, W. & Salzmann, M. Geometry-aware deep recurrent neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **59**(3), 2448–2460 (2021).
37. Xue, Z., Zhang, M., Liu, Y. & Du, P. Attention-based second-order pooling network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2020.3048128 (2021).
38. Guo, H., Liu, J., Yang, J., Xiao, Z. & Wu, Z. Deep collaborative attention network for hyperspectral image classification by combining 2-D CNN and 3-D CNN. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **13**, 4789–4802 (2020).
39. Gao, H., Miao, Y., Cao, X. & Li, C. Densely connected multiscale attention network for hyperspectral image classification. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **14**, 2563–2576 (2021).
40. Hang, R., Li, Z., Liu, Q., Ghamisi, P. & Bhattacharyya, S. S. Hyperspectral image classification with attention-aided CNNs. *IEEE Trans. Geosci. Remote Sens.* **59**(3), 2281–2293 (2021).
41. Li, Z. *et al.* Hyperspectral image classification with multiattention fusion network. *IEEE Geosci. Remote Sens. Lett.* https://doi.org/10.1109/LGRS.2021.3052346 (2021).
42. Xi, B. *et al.* Multi-direction networks with attentional spectral prior for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2020.3047682 (2021).
43. Xi, B. *et al.* Deep prototypical networks with hybrid residual attention for hyperspectral image classification. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **13**, 3683–3700 (2020).
44. Sun, H., Zheng, X. & Lu, X. A supervised segmentation network for hyperspectral image classification. *IEEE Trans. Image Process.* **30**, 2810–2825 (2021).
45. Pan, B. *et al.* DSSNet: A simple dilated semantic segmentation network for hyperspectral imagery classification. *IEEE Geosci. Remote Sens. Lett.* **17**(11), 1968–1972 (2020).
46. Liu, X., Hu, Q., Cai, Y. & Cai, Z. Extreme learning machine-based ensemble transfer learning for hyperspectral image classification. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **13**, 3892–3902 (2020).
47. Qu, Y. *et al.* Physically constrained transfer learning through shared abundance space for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2020.3045790 (2021).
48. Paoletti, M. E., Haut, J. M., Pereira, N. S., Plaza, J. & Plaza, A. Ghostnet for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2021.3050257 (2021).
49. Shen, Y. *et al.* Efficient deep learning of nonlocal features for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2020.3014286 (2021).
50. Li, Z. *et al.* Deep multilayer fusion dense network for hyperspectral image classification. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **13**, 1258–1270 (2020).
51. Xie, J., He, N., Fang, L. & Ghamisi, P. Multiscale densely-connected fusion networks for hyperspectral images classification. *IEEE Trans. Circuits Syst. Video Technol.* **31**(1), 246–259 (2021).
52. Meng, Z., Jiao, L., Liang, M. & Zhao, F. Hyperspectral image classification with mixed link networks. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **14**, 2494–2507 (2021).
53. Wei, W., Xu, S., Zhang, L., Zhang, J. & Zhang, Y. Boosting hyperspectral image classification with unsupervised feature learning. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2021.3054037 (2021).
54. Zhang, C. *et al.* Deep Feature aggregation network for hyperspectral remote sensing image classification. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **13**, 5314–5325 (2020).
55. Li, X., Ding, M. & Pižurica, A. Deep feature fusion via two-stream convolutional neural network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **58**(4), 2615–2629 (2020).
56. Yuan, Y., Wang, C. & Jiang, Z. Proxy-based deep learning framework for spectral-spatial hyperspectral image classification: Efficient and robust. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2021.3054008 (2021).
57. Zheng, J., Feng, Y., Bai, C. & Zhang, J. Hyperspectral image classification using mixed convolutions and covariance pooling. *IEEE Trans. Geosci. Remote Sens.* **59**(1), 522–534 (2021).
58. Jiang, Y., Li, Y., Zou, S., Zhang, H. & Bai, Y. Hyperspectral image classification with spatial consistence using fully convolutional spatial propagation network. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2021.3049282 (2021).
59. Mu, C., Zeng, Q., Liu, Y. & Qu, Y. A two-branch network combined with robust principal component analysis for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* https://doi.org/10.1109/LGRS.2020.3013707 (2021).
60. Ahmad, M. *et al.* A fast and compact 3-D CNN for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* https://doi.org/10.1109/LGRS.2020.3043710 (2021).
61. Huang, H., Pu, C., Li, Y. & Duan, Y. Adaptive residual convolutional neural network for hyperspectral image classification. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **13**, 2520–2531 (2020).

62. Paoletti, M. E., Haut, J. M., Plaza, J. & Plaza, A. Neural ordinary differential equations for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **58**(3), 1718–1734 (2020).
63. Li, R., Pan, Z., Wang, Y. & Wang, P. A convolutional neural network with mapping layers for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **58**(5), 3136–3147 (2020).
64. Li, K. *et al.* Depthwise separable ResNet in the MAP framework for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* https://doi.org/10.1109/LGRS.2020.3033149 (2021).
65. Li, Q., Zheng, B., Tu, B., Wang, J. & Zhou, C. Ensemble EMD-based spectral-spatial feature extraction for hyperspectral image classification. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **13**, 5134–5148 (2020).
66. Fang, L., Zhao, W., He, N. & Zhu, J. Multiscale CNNs ensemble based self-learning for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **17**(9), 1593–1597 (2020).
67. Zhao, Y. *et al.* Hyperspectral image classification via spatial window-based multiview intact feature learning. *IEEE Trans. Geosci. Remote Sens.* **59**(3), 2294–2306 (2021).
68. Feng, Z. *et al.* Hyperspectral image classification using spectral-spatial LSTMs. *Neurocomputing* **328**, 39–47 (2017).
69. Liu, Q. *et al.* Bidirectional-convolutional LSTM based spectral-spatial feature learning for hyperspectral image classification. *Remote Sens.* **9**(12), 1330 (2017).
70. Hang, R., Liu, Q., Hong, D. & Ghamisi, P. Cascaded recurrent neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **57**(8), 5384–5394 (2019).
71. Dai, Y., Gieseke, F., Oehmcke, S., Wu, Y. & Barnard, K. Attentional feature fusion. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 3559–3568 (2021).
72. Fu, J. *et al.* Dual attention network for scene segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3141–3149 (2019).
73. Wang, X., Girshick, R., Gupta, A. & He, K. Non-local neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7794–7803 (2018).
74. Hou, Q., Zhang, L., Cheng, M.-M. & Feng, J. Strip pooling: Rethinking spatial pooling for scene parsing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4002–4011 (2020).
75. Hou, Q., Zhou, D. & Feng, J. *Coordinate Attention for Efficient Mobile Network Design*. *CoRR*. Preprint at http://arXiv.org/abs/2103.02907 (2021).
76. Ma, W., Yang, Q., Wu, Y., Zhao, W. & Zhang, X. Double-branch multi-attention mechanism network for hyperspectral image classification. *Remote Sens.* **11**, 1307. https://doi.org/10.3390/rs11111307 (2019).

Author contributions

Conceptualization, A.D.; methodology, A.D.; software, S.C.; validation, A.D.; formal analysis, L.W.; writing—original draft preparation, S.C.; writing—review and editing, L.W.; visualization, A.D. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by the Natural Science Foundation of Xinjiang Uygur Autonomous Region Grant Number 2020D01C034, Tianshan Innovation Team of Xinjiang Uygur Autonomous Region grant number 2020D14044, National Science Foundation of China under Grant U1903213, 61771416 and 62041110, the National Key R&D Program of China under Grant 2018YFB1403202, Creative Research Groups of Higher Education of Xinjiang Uygur Autonomous Region under Grant XJEDU2017T002.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to L.W.