



OPEN ACCESS

EDITED BY

Zhenxu Bai,
Hebei University of Technology, China

REVIEWED BY

Liguo Wang,
Dalian Nationalities University, China
Xiaobin Hong,
South China Normal University, China

*CORRESPONDENCE

Haibin Wu,
✉ woo@hrbust.edu.cn

SPECIALTY SECTION

This article was submitted to
Optics and Photonics,
a section of the journal
Frontiers in Physics

RECEIVED 05 February 2023

ACCEPTED 16 February 2023

PUBLISHED 08 March 2023

CITATION

Wang A, Song Y, Wu H, Liu C and
Iwahori Y (2023), A hybrid neural
architecture search for hyperspectral
image classification.
Front. Phys. 11:1159266.
doi: 10.3389/fphy.2023.1159266

COPYRIGHT

© 2023 Wang, Song, Wu, Liu and Iwahori.
This is an open-access article distributed
under the terms of the [Creative
Commons Attribution License \(CC BY\)](#).
The use, distribution or reproduction in
other forums is permitted, provided the
original author(s) and the copyright
owner(s) are credited and that the original
publication in this journal is cited, in
accordance with accepted academic
practice. No use, distribution or
reproduction is permitted which does not
comply with these terms.

A hybrid neural architecture search for hyperspectral image classification

Aili Wang¹, Yingluo Song¹, Haibin Wu^{1*}, Chengyang Liu¹ and Yuji Iwahori²

¹Heilongjiang Province Key Laboratory of Laser Spectroscopy Technology and Application, Harbin University of Science and Technology, Harbin, China, ²Department of Computer Science, Chubu University, Kasugai, Aichi, Japan

Convolution neural network (CNN) is widely used in hyperspectral image (HSI) classification. However, the network architecture of CNNs is usually designed manually, which requires careful fine-tuning. Recently, many technologies for neural architecture search (NAS) have been proposed to automatically design networks, further improving the accuracy of HSI classification to a new level. This paper proposes a circular kernel convolution- β -decay regulation NAS-confident learning rate (CK- β NAS-CLR) framework to automatically design the neural network structure for HSI classification. First, this paper constructs a hybrid search space with 12 kinds of operation, which considers the difference between enhanced circular kernel convolution and square kernel convolution in feature acquisition, so as to improve the sensitivity of the network to hyperspectral information features. Then, the β -decay regulation scheme is introduced to enhance the robustness of differential architecture search (DARTS) and reduce the discretization differences in architecture search. Finally, we combined the confidence learning rate strategy to alleviate the problem of performance collapse. The experimental results on public HSI datasets (Indian Pines, Pavia University) show that the proposed NAS method achieves impressive classification performance and effectively improves classification accuracy.

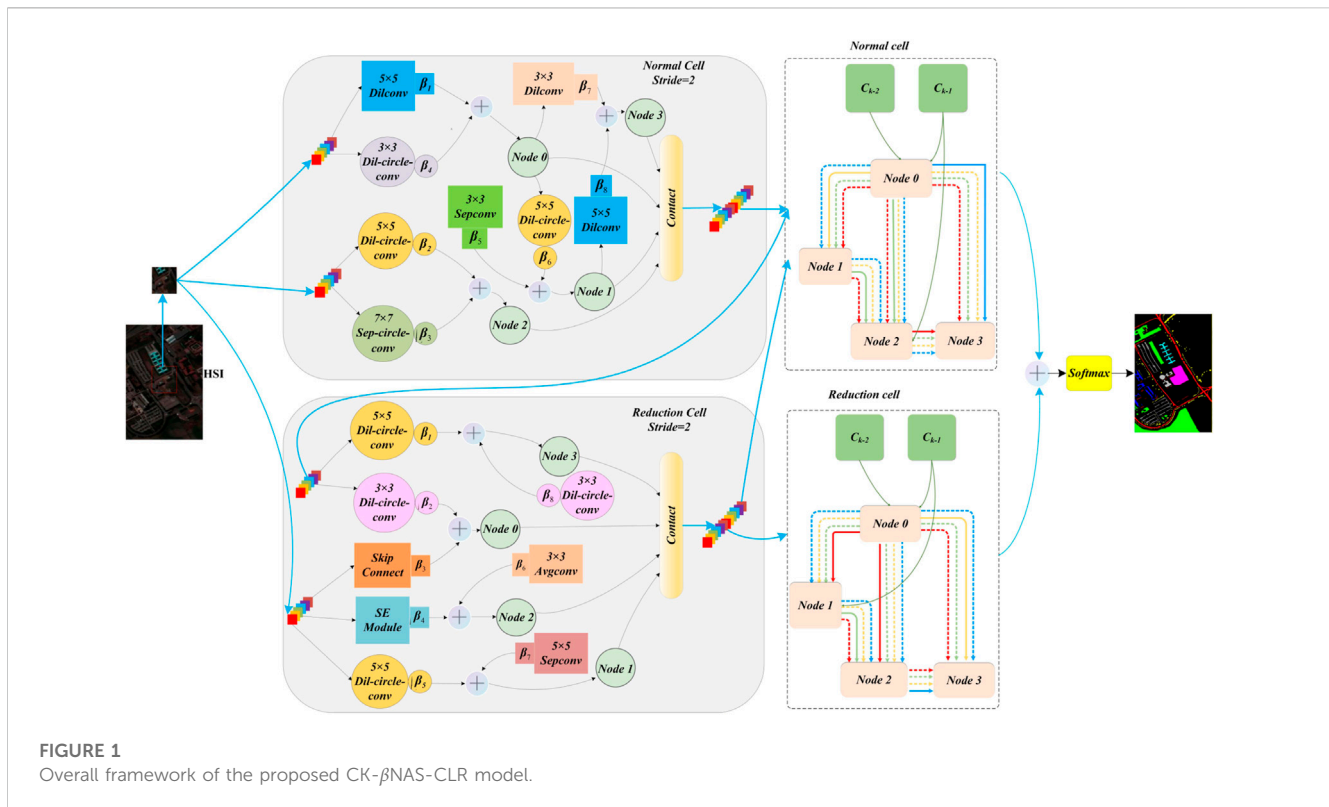
KEYWORDS

hyperspectral image classification, neural architecture search, differentiable architecture search (DARTS), circular kernel convolution, convolution neural network

1 Introduction

Hyperspectral sensing images (HSIs) collect rich spatial-spectral information in hundreds of spectral bands, which can be used to effectively distinguish ground cover. HSI classification is based on pixel level, and many traditional methods based on machine learning have been used, such as the K-nearest neighbor (KNN) [1] and support vector machine (SVM) [2]. The HSI classification method based on deep learning can effectively extract the robust features to obtain better classification performance [3–5].

Limited by the cost of computing resources and the workload of parameter adjustment, it is inevitable to promote the development of automatic design efficient neural network technology [6]. The goal of NAS (neural architecture search) is to select and combine different neural operations from predefined search spaces and to automate the construction of high-performance neural network structures. Traditional NAS work uses the



reinforcement learning algorithm (RL) [7], evolutionary algorithm (EA) [8], and the gradient-based method to conduct architecture search.

In order to reduce resource consumption, one-shot NAS methods based on supernet are developed [9]. DARTS is a one-shot NAS method with a distinguishable search strategy [10]. By introducing Softmax function, it expands the discrete space into a continuous search optimization process. Specifically, it can reduce the workload of network architecture design and reduce the process of a large number of verification experiments [9].

The method based on the automatic design of convolutional neural network for hyperspectral image classification (CNAS) introduces DARTS into the HSI classification task for the first time. The method uses point-by-point convolution to compress the spectral dimensions of HSI into dozens of dimensions and then uses DARTS to search for neural network architecture suitable for the HSI dataset [11]. Subsequently, based on the method of 3D asymmetric neural architecture search (3D-ANAS), a classification framework from pixel to pixel was designed, and the redundant operation problem was solved by using the 3D asymmetric CNN, which significantly improved the calculation speed of the model [12].

Traditional CNN design uses square kernel to extract image features, which brings significant challenges to the computing system because the number of arithmetic operations increases exponentially with the increase of network size. The features acquired by the square kernel are usually unevenly distributed [13] because the weights at the central intersection are usually large. Inspired by circular kernel (CK) convolution, this paper studies a new NAS paradigm to classify HSI data by

automatically designing hybrid search space. The main contributions of this paper are as follows:

- 1) An effective framework is proposed to design the NAS, called CK- β NAS-CLR, which is composed of a hybrid search space of 12 operations of circular convolution with different convolution methods, different scales, and attention mechanism to effectively improve the feature acquisition ability.
- 2) β -decay regularization is introduced effectively to stabilize the search process and make the searched network architecture transferable among multiple HSI datasets.
- 3) We introduced the confident learning rate strategy to focus on the confidence level when updating the structure weight gradient and to prevent over-parameterization.

2 Materials and methods

As shown in Figure 1, the NAS framework for HSI proposed is described, called as CK- β NAS-CLR. Compared with other HSI classification methods, this method aims to alleviate the shortcomings of traditional microNAS methods from three aspects of search space, search strategy, and architecture resource optimization and effectively improve the classification accuracy.

DARTS is a basic framework which adopts weight sharing and combines hypernetwork training with the search of the best candidate architecture to effectively reduce the waste of computing resources. First, the hyperspectral image is clipped into patch by sliding window as input. Then, the hybrid search space of CK convolution and attention mechanism is constructed,

and the operation search between nodes is carried out in the hybrid search space to effectively improve the feature acquisition ability of the receptive field. At the same time, the architecture parameter set β , which represents the importance of the operator, is attenuated and regularized, effectively strengthening the robustness of DARTS and reducing the discretization differences in the architecture search process. After the search is completed, the algorithm stacks multiple normal cells and reduction cells to form the optimal neural structure, and then the classification results are obtained through Softmax operation. In addition, CLR is proposed to stack decay regularization to alleviate the performance crash of DARTS, improve memory efficiency, and reduce architecture search time.

2.1 The proposed NAS framework for HSI classification

2.1.1 Integrating circular kernel to convolution

The circular kernel is isotropic and can be realized from all directions. In addition, symmetric circular nuclei can ensure rotation invariance, which uses bilinear interpolation to approximate the traditional square convolution kernel to a circular convolution kernel, and uses matrix transformation to reparametrize the weight matrix, replacing the original matrix with the changed matrix to realize the offset of receptive field reception. Without considering the loss, the expression of receptive field H of standard 3×3 square convolution kernel with a dilation of 1 is written as follows:

$$H = \{(-1, 1), (0, 1), (1, 1), (-1, 0), (0, 0), (1, 0), (-1, -1), (0, -1), (1, -1)\}, \tag{1}$$

where H represents the offset set of the neighborhood convolved on the center pixel. By convolution, the feature map is $R \in H^{S \times S}$ and kernel is $J \in H^{M \times N}$. The output feature map $U \in H^{M \times N}$ can be obtained, and the coordinates of each position are shown in formula (2).

$$U_1 \in \sum_{k \in H} R_k J_{1+k}. \tag{2}$$

So, we get $U = R \otimes J$, where \otimes represents the classical convolution operation used by the CNN. Therefore, the change of receptive field of nucleus circularis 3×3 is shown in formula(3).

$$B = \left\{ \left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right), (0, 1), \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right), (-1, 0), (0, 0), (1, 0), \left(-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2} \right), (0, -1), \left(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2} \right) \right\}. \tag{3}$$

For the sampling problem of circular convolution kernels, we selected the offset $(\{\Delta b\})$ of k for different discrete kernel positions and resampled the offset to input J to obtain circular receptive fields. Because the sampling receptive field of circular nucleus has a fraction, we use bilinear interpolation to approximate the sampling value of the receptive field.

$$U_1 \in \sum_{k \in H} R_k J_{1+k+\Delta b}, \tag{4}$$

$$J_b = \sum_{k \in H} \mathcal{V}(k, b) J_k, \tag{5}$$

where b represents the grid position in the circular receptive field and k represents all grid positions in the square receptive field, which is the kernel of two-dimensional bilinear interpolation. According to the bilinear interpolation, \mathcal{V} can be divided into two one-dimensional cores.

$$\mathcal{V}(k, b) = g(k_x, b_x) \cdot g(k_y, b_y), \tag{6}$$

$$g(q, e) = \max(0, 1 - |q - e|). \tag{7}$$

Therefore, $\mathcal{V}(k, b) \neq 0$ and $\mathcal{V}(k, b) = 1$ only correspond to the corresponding grid k of receptive field B with grid location b . Then, we let $\hat{J}_{RF(l)} \in B^{S^2 \times 1}$ and $\hat{R} \in B^{S^2 \times S^2}$ represent the adjusted receptive field centered on position i and nucleus, respectively. Generally, the standard convolution can be defined as shown in formula (8), so after replacing the circular convolution kernel, the circular convolution can be located as shown in formula (9).

$$U_l \in \hat{R}^T \hat{J}_{RF(l)}, \tag{8}$$

$$U_l \in \hat{R}^T (C \hat{J}_{RF(l)}) = \left(\hat{R}^T C \right) \hat{J}_{RF(l)}, \tag{9}$$

where $C \in B^{S^2 \times S^2}$ is a fixed sparse coefficient matrix, so let $J \in B^{S^2 \times S^2}$, $U \in B^{S^2 \times S^2}$, and $R \in B^{S^2 \times S^2}$ be the input feature map, output feature map, and kernel, respectively, so the corresponding definition of formula (9) can be written as formula (10).

$$U_l \in R \otimes (C^* J) = (R^* C) \otimes J, \tag{10}$$

where $C^* J$ is the convolution process of changing the square receptive field into a circular receptive field. Thus, we can calculate the core weight to achieve operation $C^* J$. This calculation method can effectively avoid calculating the offset of multiple convolutions and reduce the cost of core operation. Next, we summarize the analysis of the actual effect of the transformation matrix. We let $\Delta R = R^{a+1} - R^a$, and the value of a change on the output is shown in formula (11). The squared value of a change on the output is shown in formula (12).

$$\Delta U = U^{a+1} - U^a, \tag{11}$$

$$\|\Delta U\|^2 = (C^* J)^T \otimes \Delta R^T \Delta R \otimes (C^* J) = J^T \otimes (C^{T*} \Delta R^T \Delta R^* C) \otimes J \tag{12}$$

In contrast, ΔU of the traditional convolution layer is defined by $\Delta R^T \Delta R$. Therefore, it can be concluded that the transformation matrix C caused by the circular core can provide a better choice for the gradient descent path of DARTS.

2.1.2 β -decay regularization scheme

In order to alleviate unfair competition in DARTS, we introduced the β -decay regulation scheme [14] so as to improve its robustness and generalization ability and effectively reduce the search memory and the search cost to find the best architecture as shown in Figure 2.

$\alpha_o^{(m,n)}$ converts the discretization operation of the optional operation set \mathcal{O} in the search space into an operable continuous set. After implementing Softmax operation, the architecture parameter set $\beta_l^{(m,n)}$ is obtained, which is attenuated and regularized.

$$\beta_l^{(m,n)} = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(m,n)})}{\sum_{l'=1} \exp(\alpha_{o'}^{(m,n)})}, \tag{13}$$

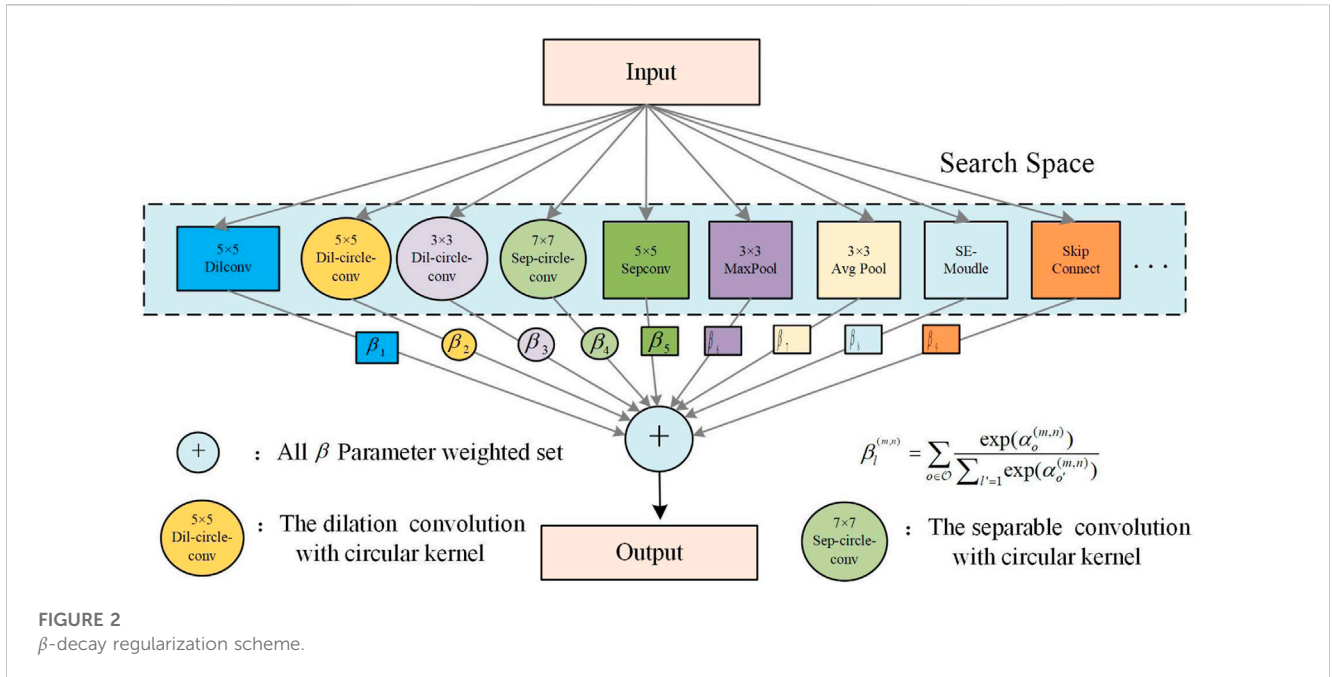


FIGURE 2 β -decay regularization scheme.

where $\beta_l^{(m,n)}$ is the combination of architecture parameters between node m and node n and l is the number of optional operations. Each cell can have up to N nodes, and $\alpha_o^{(m,n)}$ represents the architecture parameters. A special coefficient for each candidate operation $\beta_o^{(m,n)}$ is defined.

$$\beta_o^{(m,n)} = \frac{\exp(\alpha_o^{(m,n)})}{\sum_{o' \in O} \exp(\alpha_{o'}^{(m,n)})} \quad (14)$$

Starting from the default setting of regularization, consider the one-step update of architecture parameter α , where ς_α represents the learning rate a of architecture parameters.

$$\alpha_l^{t+1} \leftarrow \alpha_l^t - \eta_\alpha \cdot \nabla_\alpha \mathcal{L}_{valid} \quad (15)$$

For the special gradient descent algorithm of DARTS, these regularized gradients need to be normalized ($\mathcal{N}\mathcal{L}$) through the sum size and to realize the average distribution of the total gradient without normalization.

$$\alpha_l^{t+1} \leftarrow \alpha_l^t - \eta_\alpha \cdot \nabla_\alpha \mathcal{L}_{valid} - \varsigma_\alpha \delta \mathcal{N}\mathcal{L}(\alpha_l^t), \quad (16)$$

$$\alpha_k^{t+1} \leftarrow \alpha_k^t - \eta_\alpha \cdot \nabla_\alpha \mathcal{L}_{valid} - \varsigma_\alpha \delta(\alpha_k^y). \quad (17)$$

In the DARTS search process, the architecture parameter set, β , is used to express the importance of all operators. The research on explicit regularization of β can more clearly standardize the optimization of architecture parameters so as to improve the robustness and architecture universality of DARTS. We use the χ function with α as the independent variable to express the total impact of attenuation regularization.

$$\bar{\beta}_l^{t+1} = \chi_l^{t+1}(\alpha_l^t) \beta_l^{t+1}, \quad (18)$$

$$\alpha_l^{t+1} \leftarrow \alpha_l^t - \eta_\alpha \cdot \nabla_\alpha \mathcal{L}_{valid} - \varsigma_\alpha \delta R(\alpha_l^t), \quad (19)$$

where the χ function (R is the independent variable) represents the overall influence of β attenuation regularization and R is the

mapping function. We can iterate for dividing the single-step update parameter value β_l^{t+1} and parameter value weighted sum $\bar{\beta}_l^{t+1}$ of β .

$$\frac{\bar{\beta}_l^{t+1}}{\beta_l^{t+1}} = \frac{\sum_{l'=1}^{|\mathcal{O}|} \exp(\alpha_{l'}^{t+1})}{\sum_{l'=1}^{|\mathcal{O}|} [\exp(R(\alpha_l^t) - R(\alpha_{l'}^t))]^{\delta_{\varsigma_\alpha}} \exp(\alpha_{l'}^{t+1})} \quad (20)$$

It can be found that mapping function R determines the impact of α on β . To avoid excessive regularization and optimization difficulties, Softmax is used to normalize α .

$$R(\alpha_l) = \frac{\exp(\alpha_l)}{\sum_{l'=1}^{|\mathcal{O}|} \exp(\alpha_{l'})} \quad (21)$$

We can obtain the impact and effect of our method.

$$\chi_l^{t+1}(\alpha_l^t) = \frac{\sum_{l'=1}^{|\mathcal{O}|} \exp(\alpha_{l'}^{t+1})}{\sum_{l'=1}^{|\mathcal{O}|} \left[\exp\left(\frac{\exp(\alpha_l^t) - \exp(\alpha_{l'}^t)}{\sum_{r=1}^{|\mathcal{O}|} \exp(\alpha_r^t)}\right) \right]^{\delta_{\varsigma_\alpha}}} \exp(\alpha_{l'}^{t+1}) \chi_l^{t+1}(\alpha_l^t). \quad (22)$$

2.1.3 Confident learning rate strategy

When the NAS method is used to classify hyperspectral datasets, a large number of parameters will be generated. When the training samples are limited, the performance of the network may be reduced due to the over-fitting phenomenon, which will lead to low memory utilization during the training process. CLR is used to alleviate these two problems [15].

After applying the Softmax operation, the structure is relaxed. The gradient descent algorithm is used to optimize the $\alpha = \{\alpha^{(m,n)}\}$ matrix, and the original weight of the network is called w . Then, the cross-entropy formula is used to calculate the loss value in the training stage and the parameters \mathcal{L}_{train} and \mathcal{L}_{valid} are updated.

In order to enable both to achieve the optimization strategy at the same time, it is necessary to fix the value of $\alpha = \{\alpha^{(m,n)}\}$ matrix on the training set, update w using the gradient descent algorithm,

TABLE 1 Performance comparison of different methods of the Indian Pines dataset.

Method	EMP-SVM	SSRN	ResNet	PyResNet	CNAS	MLP-Mixer	ANAS-CPA-LS	CK-βNAS-CLR
Alfalfa	51.16 ± 17.51	100 ± 0.00	95.36 ± 0.28	97.43 ± 0.36	95.14 ± 4.27	100 ± 0.00	99.39 ± 0.32	100.0 ± 0.75
Corn-no till	66.26 ± 3.37	88.95 ± 1.19	93.36 ± 3.76	98.87 ± 1.04	96.19 ± 1.21	96.56 ± 0.09	96.43 ± 0.79	96.93 ± 0.31
Corn-min till	70.40 ± 3.74	92.87 ± 0.54	95.64 ± 0.37	89.44 ± 0.34	92.86 ± 2.64	92.10 ± 0.03	94.25 ± 0.97	96.10 ± 0.34
Corn	56.63 ± 6.68	96.56 ± 0.91	94.99 ± 4.32	97.47 ± 0.57	96.56 ± 3.94	98.15 ± 0.68	99.07 ± 0.74	99.95 ± 0.00
Grass-pasture	87.04 ± 3.52	90.52 ± 7.56	97.69 ± 2.30	97.38 ± 0.95	97.13 ± 0.27	97.87 ± 1.16	99.56 ± 0.55	94.59 ± 1.16
Grass-trees	87.30 ± 1.82	96.91 ± 1.16	97.27 ± 1.80	98.40 ± 0.12	94.02 ± 0.22	96.98 ± 0.86	96.44 ± 0.97	98.39 ± 0.79
Grass-pasture-mowed	82.78 ± 8.11	86.60 ± 0.89	99.01 ± 0.00	100 ± 0.00	92.31 ± 0.88	100 ± 0.00	90.68 ± 2.66	95.21 ± 0.05
Hay-windrowed	89.45 ± 1.75	96.28 ± 3.71	93.64 ± 1.17	95.81 ± 1.68	99.44 ± 4.66	95.80 ± 0.60	97.91 ± 0.67	95.73 ± 1.95
Oats	64.83 ± 16.3	84.37 ± 0.06	96.66 ± 1.66	94.67 ± 1.11	98.96 ± 0.56	100 ± 0.00	99.51 ± 2.02	93.15 ± 2.87
Soybean-not till	71.90 ± 2.12	92.46 ± 0.07	92.88 ± 1.90	88.20 ± 7.13	92.00 ± 0.56	93.38 ± 0.34	95.02 ± 0.43	98.69 ± 1.04
Soybean-mint till	73.01 ± 1.62	95.33 ± 0.06	94.27 ± 0.11	95.26 ± 1.87	93.90 ± 0.92	95.34 ± 0.37	98.36 ± 0.65	99.04 ± 0.79
Soybean-clean	66.49 ± 4.56	94.69 ± 2.77	96.13 ± 1.43	95.42 ± 1.79	94.88 ± 0.19	94.34 ± 1.32	99.07 ± 0.83	98.38 ± 3.19
Wheat	88.62 ± 4.35	97.84 ± 1.14	99.02 ± 0.01	99.55 ± 0.62	94.65 ± 0.45	100 ± 0.00	95.28 ± 0.57	99.41 ± 0.12
Woods	90.44 ± 1.19	94.82 ± 2.31	93.97 ± 1.67	96.84 ± 1.14	98.33 ± 0.29	95.64 ± 0.06	95.57 ± 0.99	98.77 ± 0.62
Buildings-grass-trees-drivers	71.35 ± 7.39	91.29 ± 2.52	93.66 ± 0.11	93.56 ± 3.90	94.99 ± 0.45	95.55 ± 0.15	95.14 ± 1.66	98.65 ± 1.24
Stone-steel-towers	98.10 ± 1.82	86.70 ± 1.22	94.47 ± 1.43	96.34 ± 1.72	89.57 ± 2.66	88.12 ± 2.12	88.28 ± 0.96	85.30 ± 0.85
OA/%	81.64 ± 0.02	93.58 ± 1.12	94.53 ± 0.59	94.95 ± 1.07	95.00 ± 0.56	95.95 ± 0.17	96.57 ± 0.56	97.90 ± 0.31
AA/%	75.98 ± 5.36	92.88 ± 1.64	95.50 ± 1.39	95.91 ± 1.52	95.05 ± 1.51	96.23 ± 0.48	96.24 ± 0.98	96.76 ± 1.00
100 K	71.92 ± 2.82	92.67 ± 1.29	93.74 ± 0.67	94.31 ± 0.19	94.88 ± 0.87	95.38 ± 0.20	95.69 ± 0.39	96.97 ± 0.90

fix the w value on the verification set, update the $\alpha = \{\alpha^{(m,n)}\}$ value using the gradient descent algorithm, and obtain the best parameter value repeatedly. Stop the optimization after finding the best architecture neural architecture α_0^* and minimize the verification loss $\mathcal{L}_{valid}(w^*, \alpha^*)$.

$$\min_{\alpha} \mathcal{L}_{valid}(w^*(\alpha), \alpha), \quad (23)$$

$$\text{s.t. } w^*(\alpha) = \arg \min_w \mathcal{L}_{train}(w, \alpha). \quad (24)$$

NAS architecture parameters will be over-parameterized with the increase of training time. Therefore, the gradient confidence obtained from the parameterized DARTS should increase with the training time of the architecture weight update.

$$LR_{Confident}(a) = \left(\frac{a}{A}\right)^{\tau} \times LR_{\alpha}, \quad (25)$$

where α represents the number of epochs currently trained, A represents the preset total epochs, and τ is the confidence factor of CLR. Through the update of the confidence learning rate, the network obtains \mathcal{L}_{valid} and uses it for gradient update.

$$grad_{\alpha} = \nabla_{\alpha} \mathcal{L}_{valid}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha). \quad (26)$$

The confidence learning rate is established in the process of architecture gradient update.

$$grad_{\alpha} = LR_{Confident} * \nabla_{\alpha} \mathcal{L}_{valid}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha). \quad (27)$$

3 Results

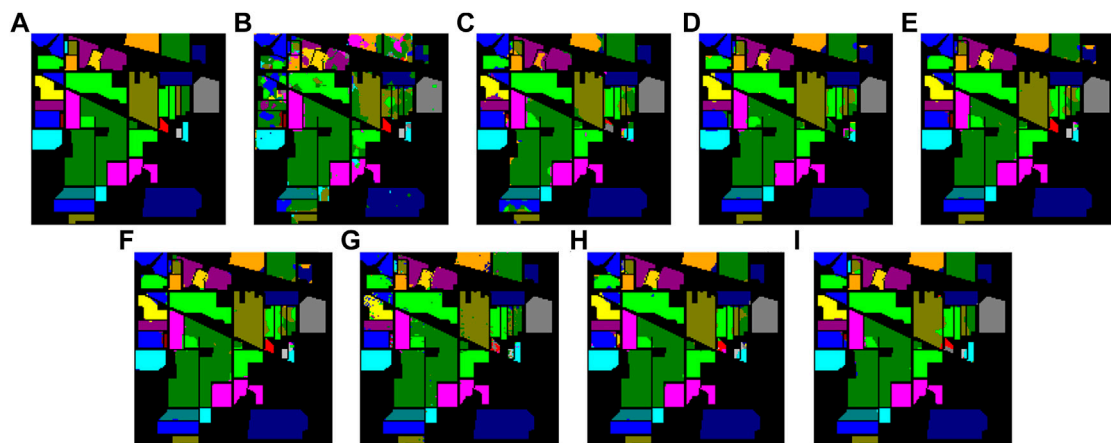
Our experiments are conducted using Intel (R) Xeon (R) 4208 CPU@2.10GHz Processor and Nvidia GeForce RTX 2080Ti graphics card. We selected the average of 10 experiments to compute the overall accuracy (OA), average accuracy (AA), and Kappa coefficient (K).

3.1 Comparison with state-of-the-art methods

In this section, we select some advanced methods to make comparison so as to evaluate the classification performance, which include extended morphological profile combined with support vector machine (EMP-SVM) [16], spectral spatial residual network (SSRN) [17], residual network (ResNet) [18], pyramid residual network (PyResNet) [19], multi-layer perceptron mixer (MLP Mixer) [20], CNAS [11], and efficient convolutional neural architecture search (ANAS-CPA-LS) [21]. All experimental results are shown in Tables 1, 2. The sample is clipped by using the sliding window strategy size of 32×32 , and the overlap rate is set to 50%. We randomly selected 30 samples as the training dataset and 20 samples as the validation dataset. The training time is set to 200, and the learning rate of the three data sets is set to 0.001.

TABLE 2 Performance comparison of different methods of the Pavia University dataset.

Method	EMP-SVM	SSRN	ResNet	PyResNet	CNAS	MLP-Mixer	ANAS-CPA-LS	CK-βNAS-CLR
Asphalt	89.06 ± 1.33	90.59 ± 0.62	95.59 ± 3.43	94.01 ± 0.54	95.55 ± 0.25	95.56 ± 1.26	97.56 ± 1.34	98.75 ± 0.69
Meadows	88.12 ± 0.23	89.26 ± 1.05	97.10 ± 2.06	99.40 ± 0.70	98.91 ± 0.71	99.45 ± 0.15	100 ± 0.00	100 ± 0.00
Gravel	78.65 ± 3.05	78.89 ± 2.65	87.53 ± 5.27	98.26 ± 1.69	93.81 ± 4.60	93.83 ± 1.03	99.78 ± 0.03	98.73 ± 0.28
Trees	88.95 ± 0.53	89.05 ± 1.45	99.03 ± 0.33	98.73 ± 0.61	99.35 ± 0.28	98.78 ± 0.45	93.73 ± 1.64	97.12 ± 0.02
Metal	93.23 ± 1.29	94.55 ± 0.67	98.56 ± 1.36	99.64 ± 0.27	99.67 ± 0.17	99.81 ± 0.81	99.24 ± 0.01	98.93 ± 0.36
Bare soil	90.13 ± 0.54	90.23 ± 1.23	98.35 ± 1.06	99.28 ± 0.04	98.39 ± 0.18	98.97 ± 0.97	100 ± 0.00	100 ± 0.00
Bitumen	81.66 ± 3.31	83.69 ± 2.82	99.29 ± 0.51	96.35 ± 0.19	92.31 ± 0.88	99.22 ± 0.19	94.63 ± 0.24	97.61 ± 0.48
Bricks	83.05 ± 1.61	83.57 ± 2.91	94.61 ± 0.50	84.58 ± 1.34	89.44 ± 4.66	88.88 ± 1.92	92.89 ± 0.42	98.23 ± 0.69
Shadows	95.26 ± 0.56	94.68 ± 0.46	99.39 ± 0.58	99.74 ± 0.05	98.96 ± 0.56	98.92 ± 1.46	96.33 ± 0.32	97.16 ± 0.91
OA/%	91.07 ± 0.85	91.32 ± 1.22	96.49 ± 1.78	96.97 ± 1.32	97.05 ± 0.45	97.55 ± 0.13	97.61 ± 0.87	98.46 ± 0.57
AA/%	87.57 ± 1.38	88.28 ± 1.54	96.61 ± 1.68	96.67 ± 0.13	96.27 ± 1.36	97.05 ± 0.91	97.12 ± 0.44	98.50 ± 0.34
100 K	88.72 ± 1.44	89.43 ± 1.09	95.31 ± 2.41	96.07 ± 0.41	96.22 ± 0.11	96.76 ± 0.18	96.86 ± 0.74	98.06 ± 0.67

**FIGURE 3**

Classification results of the Indian pines dataset. (A) Ground-truth map, (B) EMP-SVM, (C) SSRN, (D) ResNet, (E) PyResNet, (F) CNAS, (G) MLP-Mixer, (H) ANAS-CPA-LS, and (I) CK-βNAS-CLR.

In Table 1, compared with EMP-SVM, SSRN, ResNet, PyResNet, CNAS, MLP Mixer, and ANAS-CPA-LS, OA obtained by our proposed method is increased by 16.26%, 4.32%, 3.37%, 2.95%, 2.9%, 1.95%, and 1.33%, respectively, on the Indian Pines dataset. Figures 3, 4 shows the classification diagram obtained from a visual perspective. By comparing the classification diagrams obtained, we can draw a conclusion that our algorithm achieves better performance. Compared with CNAS, our method uses a hybrid search space, which can effectively expand the receptive field acquired by pixels, improve the flexibility of different convolution kernel operations to process spectrum and space, and achieve higher classification accuracy.

4 Discussion

The ablation study results are provided in Table 3. When CNAS is combined with hybrid search space, OA increases by 0.70%, 0.35%, and 0.54%, which proves that the hybrid search space can improve the sensitivity of the network to hyperspectral information features and slightly improve the classification performance of the model. Compared with CNAS, CK-NAS has no time change in the search time on the three datasets but has achieved better classification accuracy. CK-βNAS-CLR search gets better results with fewer parameters and involves less computational complexity.

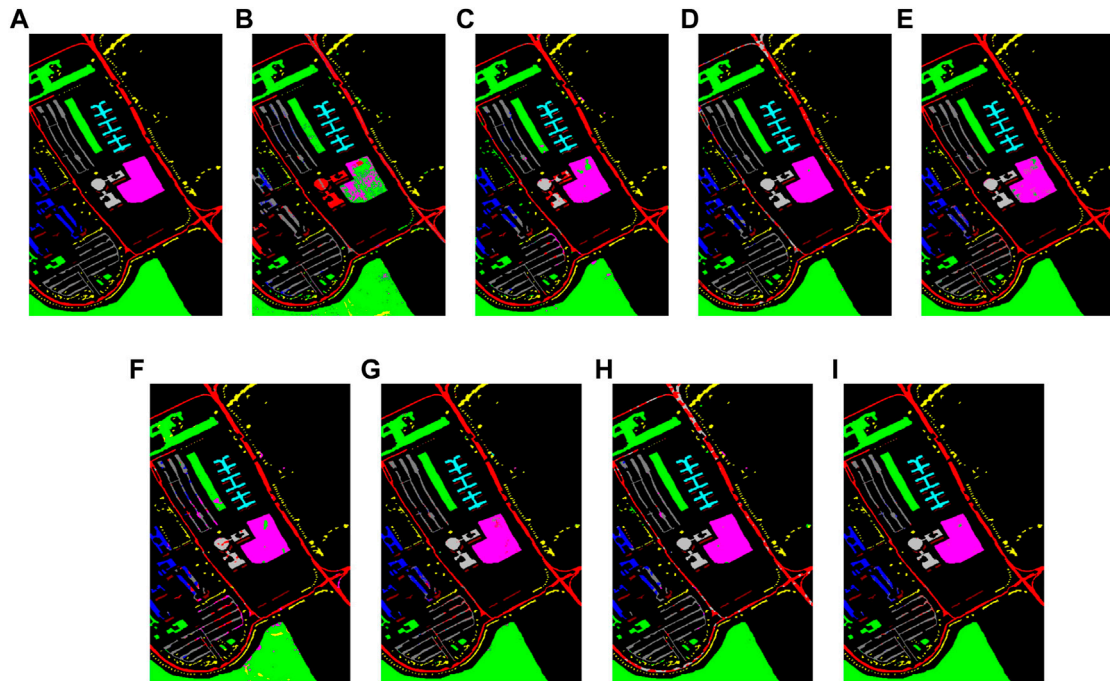


FIGURE 4

Classification results of the Pavia University dataset. (A) Ground-truth map, (B) EMP-SVM, (C) SSRN, (D) ResNet, (E) PyResNet, (F) CNAS, (G) MLP-Mixer, (H) ANAS-CPA-LS, and (I) CK- β NAS-CLR.

TABLE 3 Ablation results on the two datasets.

Dataset	Index	CNAS	CK + NAS	CK+ β +NAS	CK+ β +NAS + CLR
Indian Pines	OA (%)	95.00 \pm 0.56	95.70 \pm 0.87	96.80 \pm 0.31	97.82 \pm 0.31
	AA (%)	95.05 \pm 1.51	93.12 \pm 0.42	94.60 \pm 0.42	96.76 \pm 1.00
	100 K	94.88 \pm 0.87	95.11 \pm 0.17	96.95 \pm 0.17	96.87 \pm 0.90
	Search cost (h)	2.702	2.687	2.562	2.473
	Params (M)	0.082	0.075	0.075	0.070
Pavia University	OA (%)	97.05 \pm 0.45	97.40 \pm 0.29	97.82 \pm 0.20	98.46 \pm 0.57
	AA (%)	96.27 \pm 1.36	97.14 \pm 1.36	97.51 \pm 1.83	98.50 \pm 0.86
	100 K	96.22 \pm 0.11	96.65 \pm 0.79	97.25 \pm 2.51	98.06 \pm 0.67
	Search cost (h)	3.054	3.013	2.867	2.733
	Params (M)	0.176	0.172	0.172	0.168

5 Conclusion

In this paper, the neural network structure CK- β NAS-CLR is proposed. First of all, we introduce a hybrid search space with circular kernel convolution, which can not only enhance the robustness of the model and the ability of receptive field acquisition but also achieve a better role in optimizing the path. Second, we quoted the β -decay regulation scheme, which reduced the discretization difference and the search time. Finally, the

confidence learning rate strategy is introduced to improve the accuracy of model classification and reduce computational complexity. The experiment was conducted on two HSI datasets, and CK- β NAS-CLR is compared with seven methods, and the experimental results show that our method achieves the most advanced performance while using less computing resources. In future, we will use an adaptive subset of the data even when training the final architecture, which may lead to faster runtime and lower regularization term.

Data availability statement

Publicly available datasets were analyzed in this study. These data can be found here: <https://www.ehu.es/ccwintco/index.php?%20title=Hyperspectral-Remote-Sensing-Scenes>.

Author contributions

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Funding

This work was funded by the Reserved Leaders of Heilongjiang Provincial Leading Talent Echelon of 2021, high and foreign expert's introduction program (G2022012010L), and

References

- Chandra B, Sharma RK. On improving recurrent neural network for image classification. In: Proceeding of the International Joint Conference on Neural Networks. Alaska, USA: IJCNN (2017). p. 1904–7. doi:10.1109/IJCNN.2017.7966083
- Samadzadegan F, Hasani H, Schenk T. Simultaneous feature selection and SVM parameter determination in classification of hyperspectral imagery using Ant Colony Optimization. *Remote Sens* (2012) 38:139–56. doi:10.5589/m12-022
- Hu W, Huang Y, Wei L, Zhang F, Li H. Deep convolutional neural networks for hyperspectral image classification. *J Sensors* (2015) 2015:1–12. doi:10.1155/2015/258619
- Makantasis K, Karantzalos K, Doulamis A, Doulamis N. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). Milan, Italy: IGARSS (2015). p. 4959–62.
- Li W, Wu G, Zhang F, Du Q. Hyperspectral image classification using deep pixel-pair features. *IEEE Trans Geosci Remote Sensing* (2017) 55(2):844–53. doi:10.1109/tgrs.2016.2616355
- Kaifeng B, Lingxi X, Chen X, Longhui W, Tian Q. GOLD-NAS: Gradual, one-level, differentiable (2020). p. 03331. arXiv:2007.
- Real E, Aggarwal A, Huang Y, Le QV. "Regularized evolution for image classifier architecture search," in Proc. AAAI Conf. Artif. Intell (2019) 33, 4780–9. doi:10.1609/aaai.v33i01.33014780
- Tan M, Chen B, Pang R, Vasudevan V, Sandler M, Howard A, et al. *MnasNet: Platform-aware neural architecture search for mobile*. Long Beach, CA, USA: CVPR (2019). 2820–8.
- Liang H. DARTS+: Improved differentiable architecture search with early stopping (2020). arXiv:1909.06035. [Online]. Available: <https://arxiv.org/abs/1909.06035> (Accessed October 20, 2020).
- Guo Z. "Single path one-shot neural architecture search with uniform sampling," in Proc. IEEE Eur. Conf. Comput. Vis (2020), 544–60.
- Chen Y, Zhu K, Zhu L, He X, Ghamisi P, Benediktsson JA. Automatic design of convolutional neural network for hyperspectral image classification. *IEEE Trans Geosci Remote Sensing* (2019) 57(9):7048–66. doi:10.1109/tgrs.2019.2910603
- Zhang H, Gong C, Bai Y, Bai Z, Li Y. *3d-anas: 3d asymmetric neural architecture search for fast hyperspectral image classification* (2021). arXiv preprint arXiv: 2101.04287, 2021. [Online]. Available: <https://arxiv.org/abs/2101.04287> (Accessed January 12, 2021).
- Li G, Qian G, Delgadillo IC, Muller M, Thabet A, Ghanem B. Sgas: Sequential greedy architecture search. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit (2020). p. 1620–30.
- Ye P, Li B, Li Y, Chen T, Fan J, Ouyang W, et al. beta-DARTS: Beta-Decay regularization for differentiable architecture search, in Proceeding of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA (2022), 10864–73.
- Ding Z, Chen Y, Li N, Zhao D. BNAS-v2: Memory-Efficient and performance-collapse-prevented broad neural architecture search. *IEEE Trans Syst Man, Cybernetics: Syst* (2022) 52(10):6259–72. doi:10.1109/TSMC.2022.3143201
- Melgani F, Bruzzone L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans Geosci Remote Sensing* (2004) 42(8):1778–90. doi:10.1109/tgrs.2004.831865
- 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 Sensing* (2018) 56(2):847–58. doi:10.1109/tgrs.2017.2755542
- Liu X, Meng Y, Fu M. "Classification research based on residual network for hyperspectral image," In, Proceeding of the 2019 IEEE 4th International Conference on Signal and Image Processing (IC) (2019). 911–5.
- Paoletti ME, Haut JM, Fernandez-Beltran R, Plaza J, Plaza AJ, Pla F. Deep pyramidal residual networks for spectral-spatial hyperspectral image classification. *IEEE Trans Geosci Remote Sensing* (2019) 57(2):740–54. doi:10.1109/TGRS.2018.2860125
- He X, Chen Y. Modifications of the multi-layer Perceptron for hyperspectral image classification. *Remote Sensing* (2021) 13(17):3547. doi:10.3390/rs13173547
- Wang A, Xue D, Wu H, Gu Y. Efficient convolutional neural architecture search for LiDAR DSM classification. *IEEE Trans Geosci Remote Sensing* (2022) 60:1–17. Art no. 5703317. doi:10.1109/TGRS.2022.3171520

the Key Research and Development Program Guidance Project (GZ20220123).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.