RSNet: The Search for Remote Sensing Deep Neural Networks in Recognition Tasks

Junjue Wang, *Student Member, IEEE*, Yanfei Zhong, *Senior Member, IEEE*, Zhuo Zheng, *Graduate Student Member, IEEE*, Ailong Ma, *Senior Member, IEEE*, and Liangpei Zhang, *Fellow, IEEE*

***Abstract*— Deep learning algorithms, especially convolutional neural networks (CNNs), have recently emerged as a dominant paradigm for high spatial resolution remote sensing (HRS) image recognition. A large amount of CNNs have already been successfully applied to various HRS recognition tasks, such as land-cover classification and scene classification. However, they are often modifications of the existing CNNs derived from natural image processing, in which the network architecture is inherited without consideration of the complexity and specificity of HRS images. In this article, the remote sensing deep neural network (RSNet) framework is proposed using an automatically search strategy to find the appropriate network architecture for HRS image recognition tasks. In RSNet, the hierarchical search space is first designed to include module- and transition-level spaces. The module-level space defines the basic structure block, where a series of lightweight operations as candidates, including depthwise separable convolutions, is proposed to ensure the efficiency. The transition-level space controls the spatial resolution transformations of the features. In the hierarchical search space, a gradient-based search strategy is used to find the appropriate architecture. In RSNet, the task-driven architecture training process can acquire the optimal model parameters of the switchable recognition module for HRS image recognition tasks. The experimental results obtained using four benchmark data sets for land-cover classification and scene classification tasks demonstrate that the searched RSNet can achieve a satisfactory accuracy with a high computational efficiency and, hence, provides an effective option for the processing of HRS imagery.**

***Index Terms*— High-resolution remote sensing image, remote sensing recognition, search for convolutional neural networks (CNNs).**

I. Introduction

With the rapid development of remote sensing technology, huge quantities of high-resolution remote sensing images are now available. Compared with low-resolution images, high spatial resolution remote sensing (HRS) images contain more detailed spatial information, which not only brings opportunities but also challenges to the recognition of remote sensing images. Recognition and analysis based on HRS imagery technology have now been applied in various recognition tasks.

In the past decades, extensive efforts have been made to obtain more robust and efficient features in the HRS imagery recognition tasks, such as land-cover classification and scene classification. The land-cover classification task has long been a challenging task in remote sensing. The conventional methods rely purely upon low-level spectral and spatial features, such as the histogram of oriented gradients (HOG) [1], the object spectral index [2], scale-invariant feature transform (SIFT) [3], and the gray-level co-occurrence matrix (GLCM) [4]. The conditional random field (CRF) model, as a contextual classification model, has a natural advantage in pixel-level classification tasks. Wang *et al.* [5] used CRF and Gabor texture features to realize urban forest cover mapping. Zhao *et al.* [6] integrated spectral, spatial-contextual, and location cues within a CRF framework to provide complementary information from varying perspectives and addressed the common problem of spectral variability in land-cover mapping. Differing from land-cover classification, scene classification requires more abstract information. These tasks only focus on the entire image label, which usually contains social semantic information (commercial, residential, industrial, and so on). Based on the low-level features, semantic information can be further abstracted through the bag-of-visual words (BoVW) model [7], feature coding (FC) [8], or the probabilistic topic model (PTM) [9]. When using these traditional methods, feature designing is the key factor that affects the final recognition accuracy. However, handcrafted features may overlook subjective prior information, in which it is difficult to extract the essential features of specific HRS image data sets.

Recently, deep learning algorithms have become the dominant paradigm in machine learning and pattern recognition. From the milestone work of convolutional neural networks (CNNs) for image classification in the ImageNet large-scale visual recognition challenge [10], a variety of CNN-based methods [11]–[14] are now dominating the field of HRS imagery recognition tasks [15]–[18]. Compared with traditional handcrafted features, CNNs are data-driven methods, in which the more representative and essential features are learned end-to-end hierarchically. Due to their

Manuscript received November 11, 2019; revised January 31, 2020 and May 5, 2020; accepted June 4, 2020. Date of publication June 23, 2020; date of current version February 25, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0504202, in part by the National Natural Science Foundation of China under Grant 41771385, in part by the National Natural Science Foundation of China under Grant 41801267, and in part by the China Postdoctoral Science Foundation under Grant 2017M622522. *(Corresponding authors: Yanfei Zhong; Ailong Ma.)*

The authors are with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China, and also with the Hubei Provincial Engineering Research Center of Natural Resources Remote Sensing Monitoring, Wuhan University, Wuhan 430079, China (e-mail: kingdrone@whu.edu.cn; zhongyanfei@whu.edu.cn; zhengzhuo@whu.edu.cn; maailong007@whu.edu.cn; zlp62@whu.edu.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2020.3001401

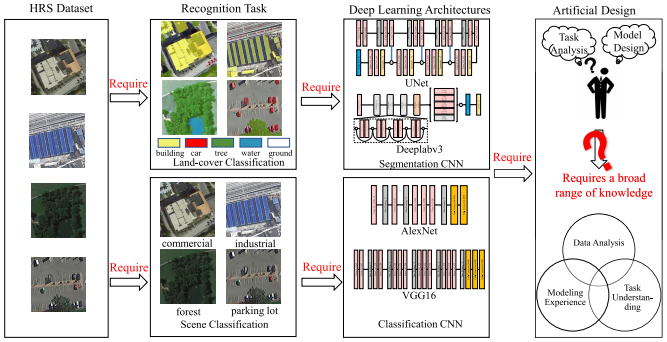


Fig. 1. Overview of a handcrafted CNN for the HRS imagery recognition task.

powerful feature extraction capabilities, more and more advanced CNN architectures have been applied in HRS imagery recognition. Castelluccio *et al.* [19] successfully transferred GoogLeNet and CaffeNet into scene classification tasks, as well as proving the importance of the pretraining strategy. With the appearance of the fully CNN [12], deep learning methods, i.e., segmentation CNNs, have gradually been applied in land-cover classification tasks. The DeepGlobe Workshop at CVPR2018 [20] launched three HRS land-cover classification challenges: road extraction, building extraction, and land-cover mapping, where fully CNNs [21]–[23] produced better performances than the other baseline networks.

Although the performance of HRS imagery recognition has been significantly improved with the help of deep learning methods, designing a good CNN architecture for a specific recognition task requires a broad mix of expertise. As shown in Fig. 1, to artificially design a deep learning architecture, the tasks include different requirements, and different types of CNNs need to be carefully constructed. When designing a CNN, the designer needs to be equipped with the following qualities: 1) excellent data analysis skills; 2) a good understanding of the different HRS imagery recognition tasks; and 3) broad experience in the design of various recognition models. This means that the designer requires a wide range of expertise, in both remote sensing and computer vision. Furthermore, handcrafted CNNs may not fit the task well due to the inadequate experiments or the lack of experience.

To tackle these issues and reduce the difficulty and complexity of HRS imagery recognition, a remote sensing deep neural network (RSNet) search framework based on neural architecture search (NAS) is proposed to automatically find the most suitable RSNet for HRS image recognition tasks. The field of NAS is a branch of automatic machine learning (AutoML), which aims to learn the model architecture directly on the data set of interest in an automatic manner [24]. In terms of the NAS methods, the current techniques usually fall into one of three categories: reinforcement learning (RL) [24]–[26], evolutionary algorithms (EAs) [27], [28], and gradient-based (GB) methods [29], [30]. To frame NAS as an RL problem, the generation of a neural architecture can be considered to be the agent's action, with the action space identical to the search space [31]. The different RL approaches differ in how they design the agent's policy and optimization strategy. Zoph and Le [25] used a reinforcement algorithm to estimate the parameters of a recurrent neural network (RNN), which samples a string specifying the structure of the CNN. They initially chose a reinforcement policy gradient algorithm for the optimization strategy, which was replaced with proximal policy optimization (PPO) in their follow-up work [24].

An alternative to RL is to use EAs. These methods encode the neural network architecture as a sequence of numbers. According to the performance estimates on the validation set, crossover and mutation operations are performed, generating new high-performance architectures. This approach is initially used to search for both the structures and parameters of the network [32]. As the number of neural network parameters continues to grow, more recent works [28], [33] have used an EA to search for the structure while using stochastic gradient descent (SGD) methods to optimize the network parameters. However, RL and EA search algorithms are computationally demanding, despite their remarkable performance. For example, it takes 1800 days of RL to search a state-of-the-art architecture based on the CIFAR-10 data set [25] or 2000 days of EA [34]. Differing from the RL and EA methods using a discrete search space, the GB methods [29], [30] introduce a simple, continuous relaxation scheme for the search space. All the parameters are trained based on gradient descent, which reduces the time required for the neural network search to less than a week. However, these NAS methods are sophisticated and inflexible and cannot be applied in HRS imagery recognition tasks (scene classification and land-cover classification). To the best of our knowledge, this is the first time that the idea of searching for a suitable deep neural network automatically has been proposed in the field of HRS imagery recognition.

Based on the natural advantages of NAS technology, the proposed RSNet search framework not only eliminates the need for manual modeling but also achieves well-performing structures that are suitable for certain data distributions. In summary, the key contributions are threefold as follows.

1) *Remote Sensing Deep Neural Network Search Framework:* The proposed RSNet search framework employs a two-stage cascade optimization strategy. In the GB architecture search (search stage), we set up a hierarchical basic search space to increase the search efficiency, inspired by hierarchical architecture search [30]. The architecture and model parameters are optimized alternately based on the gradient-descent method. After the search finishes, the RSNet can be decoded with the architecture parameters. In the task-driven architecture training (training stage), the RSNet is then retrained again to optimize the model parameters. To enhance the generalization capability of the framework, a switchable recognition module is designed for different HRS imagery recognition tasks.
2) *Gradient-Based Architecture Search:* According to the large-scale HRS imagery data sets, the efficient GB architecture search is proposed. First, the hierarchical search space is carefully designed, i.e., a module-level

space and a transition-level space. Under the restriction of the search space, the GB method introduces a simple, continuous relaxation scheme for the search space, which leads to a differentiable learning objective for the joint optimization of the structure, as well as its parameters. As opposed to an inefficient asynchronous optimization [24], [34], all the parameters can be trained end-to-end, which makes it efficient and quick to search for high-performance networks. Moreover, to ensure the efficiency of the searched network, a series of lightweight operations, including depthwise separable convolutions, is proposed for the module-level search.

3) *Task-Driven Architecture Training:* Because HRS recognition tasks include scene classification and land-cover classification tasks, which have different output requirements, task-driven architecture training is proposed utilizing a switchable recognition module. For the scene classification task, the output of the architecture is processed with global average pooling to obtain the classification probability vector. For land-cover classification task, we use atrous spatial pyramid pooling to process the output for the segmentation probability map, where multiscale contextual information is utilized. This switchable module makes the proposed framework more generalizable for different HRS imagery recognition tasks.

We implemented the proposed method on scene classification and land-cover classification benchmark data sets. All the networks obtained by the framework and the popular CNNs were tested from two aspects: accuracy and efficiency. The experimental results suggest that the searched networks can achieve a good tradeoff between accuracy and efficiency. Through our framework, the structures searched for specific high-resolution remote sensing image data sets can guide the manual design of efficient CNN networks.

The rest of this article is organized as follows. Section II introduces the classic CNN architectures for classification and segmentation tasks. Section III describes the general workflow and the key components of the proposed search framework. The experimental results and a comparative analysis are provided in Section IV, followed by a discussion in Section V. Finally, Section VI discusses our conclusions and future research directions.

II. Background to CNN Architectures

Many classic CNN architectures have been widely applied in recognition tasks. These CNNs can be classified into two categories: classification architectures and segmentation architectures, according to the different recognition requirements.

As for classification CNNs, they aim to capture the global information of the image. AlexNet, as proposed by Krizhevsky *et al.* [10], was the winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 [35]. This ConvNet, which has 60 million parameters and 650 000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully

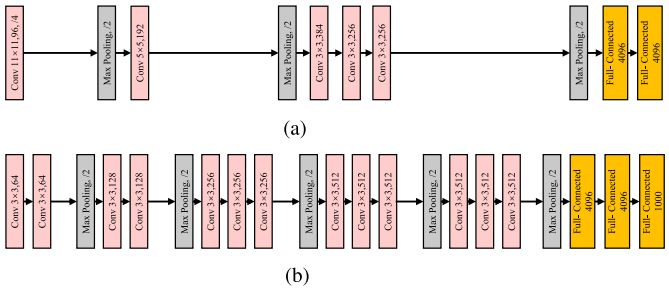


Fig. 2. Classic classification CNN architectures. (a) AlexNet. (b) VGG16.

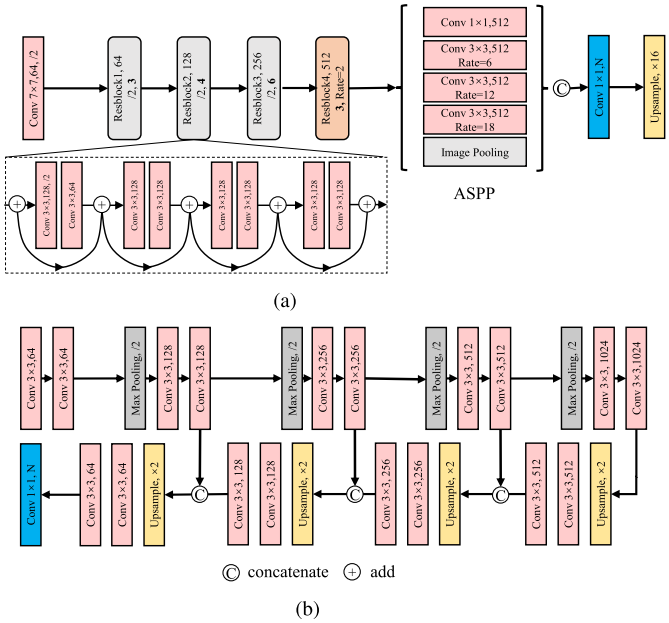


Fig. 3. Classic segmentation CNN architectures. (a) Deeplabv3. (b) UNet.

connected layers. As the earliest successful deep learning model, it has been applied to many HRS imagery classification tasks [36], [37]. The whole architecture is shown in Fig. 2(a). Although AlexNet has surpassed the traditional methods, the shallow layers limit its ability to extract complex features. To fit more complex data distributions, networks are also becoming more complex. One of the solutions is to increase the depth of the architecture. VGG16 is a typical network [11]. This network, for which the architecture is shown in Fig. 2(b), has 13 convolutional layers, five pooling layers, and three fully connected layers. Compared with AlexNet, VGG16 has shown a significant improvement in accuracy, but the efficiency is decreased.

Differing from classification tasks, segmentation tasks not only focus on contextual semantic information but also on local details. As the fully CNNs have emerged, Google has proposed a series of segmentation networks, among which DeepLabv3 [38] is the representative one. The segmentation architectures include encoder and decoder structures. The function of the encoder is to reduce the resolution of the feature maps and extract high-level semantic information. The decoder increases the resolution by upsampling to restore the local details. As shown in Fig. 3(a), DeepLabv3 reuses

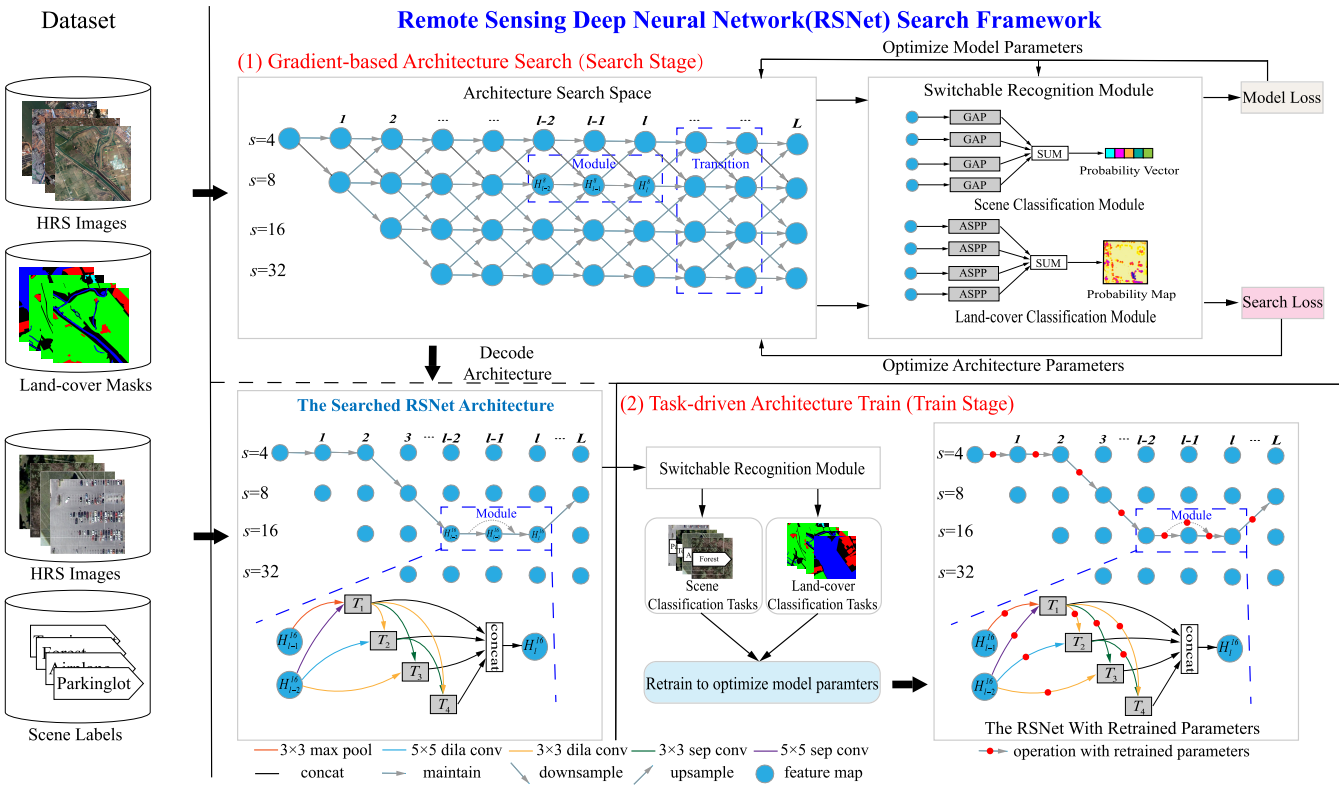


Fig. 4. Overview of the proposed deep neural network search framework for HRS imagery recognition. s: output stride, the reduction factor of the spatial resolution relative to the original image. L: Number of the CNN's architecture layers. GAP: global average pooling. ASPP: atrous spatial pyramid pooling.

ResNet [39] as an encoder and bilinear upsampling as a decoder. In addition, dilated convolutions are used in ResBlock4 to keep the resolution, and atrous spatial pyramid pooling is used to acquire multiscale features. However, in the process of restoring the resolution with 16× upsampling, some details will be lost.

Another classic segmentation network is UNet [40], which was first proposed in medical image analysis. UNet has an elegant symmetric encoding and decoding structure, as shown in Fig. 3(b). To keep the details, high-resolution features from the encoder are combined with the upsampled outputs in the decoder. The two following convolutional layers can then learn to assemble a more precise output based on this information. Although this contracting and expansive path has greatly improved the performance, the high-resolution features processed in the decoder take up a lot of memory and reduce the speed. This architecture has already been successfully applied in HRS imagery segmentation tasks [17], [41].

All of the abovementioned reasons are typical handcrafted deep learning architectures. However, the design of these excellent CNN structures requires rich experience and substantial effort. Furthermore, these architectures, if directly applied in HRS imagery recognition tasks, may result in a bottleneck in efficiency.

III. Proposed Framework

The overview of our search framework is shown in Fig. 4. The proposed framework is made up of the following.

A. Architecture Search Space

Classic CNNs [11], [39], [40] are composed of repeated modules and well-designed spatial resolution structures. Referring to the previous work [24], [29], [30], we set up a hierarchical basic search space, i.e., a module-level space and a transition-level space. The architecture search space consists of various filtering and sampling operations, where the input images are transformed into different feature maps, which are represented as the blue nodes in Fig. 4. To relax the search space to be continuous for optimization, each operation is given a weight indicating how much it contributes to the feature map.

The module-level space defines the basic structure block, which is composed of two feature maps as inputs [29], one feature map as output, and a series of basic filters (see Fig. 4). The feature maps maintain the spatial resolution in the module-level space and the basic operations are applied to the inputs according to the normalized module's weight α. The transition-level space controls the spatial resolution [30], which is called the output stride in Fig. 4. The output stride is the reduction factor of the spatial resolution relative to the original image. In the transition-level space, each feature map in the previous layer can be transformed (upsampled/downsampled by a ratio of 2 or maintained) into the next layer. At the same time, the feature map in the next layer gathers the outputs from the different output stride according to the transition's weights. As shown in Fig. 4, the module architectures and transition architectures are inseparable, and they make up the whole search space.

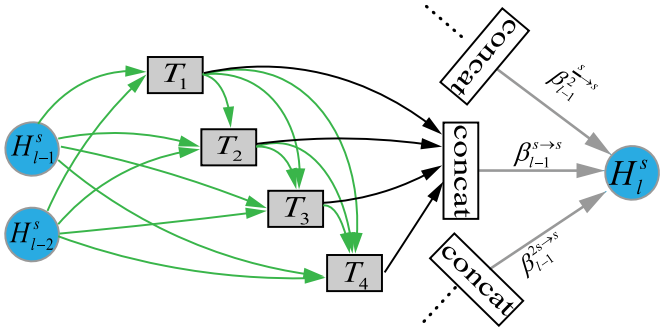


Fig. 5. Module-level search space. The blue nodes represent the feature maps in the network. Every green arrow is associated with module parameters α. The three arrows after concatenation are associated with transition parameters β.

The module's weights and transition's weights are defined as the architecture parameters, which can be easily trained end-to-end by standard backpropagation. Our goal is to find the combination of operations, as well as the resolution transition path, that contributes the most, referring to these architecture parameters.

1) Module-Level Search Space: We define a module to be a combination of B blocks, typically repeated multiple times to form the entire architecture. As shown in Fig. 5, the module is a directed acyclic graph, which has two previous feature maps, H^s_{l-1}, H^s_{l-2} as inputs, where s and l define the output stride and the layer index of the feature map, respectively. These two inputs produce B (B equals 4 in Fig. 5) intermediate results $\{T_1, \ldots, T_b, \ldots, T_B\}$ in order. For T_b, the specific formula is as follows:

$$T_b = \bar{O}^{1\to b}\left(H^s_{l-1}\right) + \bar{O}^{2\to b}\left(H^s_{l-2}\right) + \sum_{i=1}^{b-1} \bar{O}^{i+2\to b}(T_i) \quad (1)$$

where $\bar{O}$ indicates a series of weighted operations, represented by the green arrow in Fig. 5. They are applied on the feature maps and obtain the weighted results according to the module's parameter α, which is defined as

$$\bar{O}(x) = \sum_{O_k \in \mathcal{O}} \alpha_k O_k(x) \quad (2)$$

where

$$\sum_{k=0}^{|\mathcal{O}|} \alpha_k = 1, \quad \alpha_k \geq 0 \quad (3)$$

where α_k are normalized scalars associated with each operator $O_k \in \mathcal{O}$, which is easily implemented as softmax.

We let $\mathcal{O}$ be a set of candidate operations consisting of the following set of eight functions, which are all prevalent in modern CNNs [26]. To ensure that the obtained network has high efficiency, all the convolutions are replaced by lightweight depthwise separable convolutions [42]. The candidate operations are listed in Table I.

In this article, these operations are shown as abbreviations: max pooling (max pool), average pooling (avg pool), depthwise separable convolution (sep conv), and depthwise separable dilated convolution (dila conv).

TABLE I
CANDIDATE OPERATIONS FOR THE MODULE CONSTRUCTION

Type	Kernel Size	Operation
Pooling	3×3	max pooling
	3×3	average pooling
Convolution	3×3	depthwise separable convolution
	5×5	depthwise separable convolution
	3×3	depthwise separable dilated convolution
	5×5	depthwise separable dilated convolution
Other	-	identity
	-	no connection

As shown in Fig. 5, the output of this module $T^{s\to s}_l$ is the concatenation of the intermediate results $\{T_1, \ldots, T_b, \ldots, T_B\}$. Together with (1) and (2), the module-level update can be summarized as

$$T^{s\to s}_l = \text{Module}\left(H^s_{l-1}, H^s_{l-2}, \alpha\right). \quad (4)$$

The next node H^s_l is the sum of $\{T^{s\to s}_l, T^{\frac{s}{2}\to s}_l, T^{2s\to s}_l\}$ according to the normalized β in the transition level.

2) Transition-Level Search Space: Within the module, all the feature maps maintain the same spatial size [29], which enables the (weighted) sum in (1) and (2). However, the feature maps in transition space may take different spatial sizes, as is clearly shown in Fig. 4. To cover the prevalent CNNs, we have designed it such that each layer will have at most four hidden states $\{H^4_l, H^8_l, H^{16}_l, H^{32}_l\}$, with the superscript indicating the spatial resolution [30].

The feature map H^s_{l-1} can be transformed into adjacent resolutions $(s/2)$ and $2s$ or remains unchanged via the following three operations.

1) *Downsample:* Convolution operation with stride 2, both to reduce the spatial size and double the number of filters.
2) *Upsample:* Bilinear interpolation, both to double the spatial size and reduce the number of filters.
3) *Maintain:* No operation.

We limit the feature map to a minimum spatial resolution $s = 32$ and a maximum $s = 4$. β indicates the transition probability between different spatial sizes. After normalization by softmax, β meets the conditions as follows:

$$\beta^{s\to\frac{s}{2}}_l + \beta^{s\to s}_l + \beta^{s\to 2s}_l = 1 \quad \forall l, s \quad (5)$$

$$\beta^{s\to\frac{s}{2}}_l \geq 0 \quad \beta^{s\to s}_l \geq 0 \quad \beta^{s\to 2s}_l \geq 0 \quad \forall l, s. \quad (6)$$

Thus, in the transition space, we combine the previous two feature maps and obtain three combinations: $\{H^{\frac{s}{2}}_{l-1}, H^s_{l-2}\}$, $\{H^s_{l-1}, H^s_{l-2}\}$, and $\{H^{2s}_{l-1}, H^s_{l-2}\}$. Fig. 5 shows the $\{H^s_{l-1}, H^s_{l-2}\}$ as inputs processed by the module. As we associate a scalar with each gray arrow in Figs. 4 and 5, the transition level update is

$$\begin{aligned} H^s_l = {} & \beta^{\frac{s}{2}\to s}_l \text{Module}\left(H^{\frac{s}{2}}_{l-1}, H^s_{l-2}; \alpha\right) \\ & + \beta^{s\to s}_l \text{Module}\left(H^s_{l-1}, H^s_{l-2}; \alpha\right) \\ & + \beta^{2s\to s}_l \text{Module}\left(H^{2s}_{l-1}, H^s_{l-2}; \alpha\right). \end{aligned} \quad (7)$$

In conclusion, α and β are architecture parameters, where α indicates the weights of the mixed operations in the

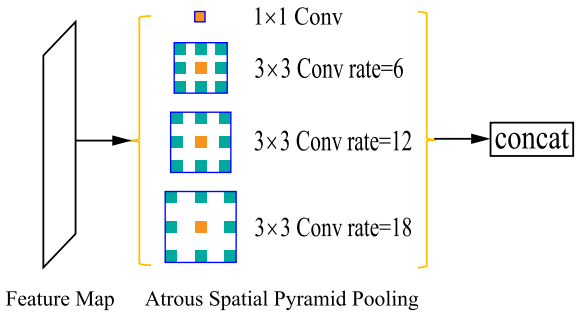


Fig. 6. ASPP attached after each spatial resolution output. The features extracted for each sampling rate are further processed in separate branches and fused to generate the final result.

modules and β represents the state transition probability of the spatial resolution between layers. They are core parameters for parsing the searched neural network.

B. Switchable Recognition Module

The switchable recognition module is designed to gather the different resolution feature maps and includes the scene classification module and land-cover classification module. Based on the scene classification module, the network we obtained is suitable for the scene classification task. Through the land-cover classification module, our obtained network is designed for the land-cover classification task.

1) Scene Classification Module: After four different spatial size feature maps are acquired in the last layer, we use global average pooling layers to generate probability vectors corresponding to the categories in the classification task. Compared with traditional fully connected layers, there are no parameters to optimize in global average pooling layers, thus avoiding overfitting [43]. Finally, the four vectors are summed to obtain the output of the classification network.

2) Land-Cover Classification Module: For the land-cover classification tasks, we have designed an atrous spatial pyramid pooling (ASPP) module [38] for each spatial resolution in the recognition head, as shown in Fig. 4. This strategy exploits multiscale features by employing multiple parallel convolutional filters with different dilation rates. Moreover, the multiscale features are fused by concatenation, which enhances the recognition ability for complex ground objects. The implementation of ASPP in the land-cover classification module is shown in Fig. 6. As we obtain four different multiscale features at the Lth layer, they are bilinearly upsampled to the original resolution before being summed to produce the probability map.

C. Two-Stage Cascade Optimization Strategy

The proposed deep neural network search framework has two types of parameters: the traditional CNN model parameters w and the architecture parameters α, β. Hence, we propose a two-stage cascade optimization strategy. In the search stage, these parameters are optimized alternately. In the training stage, only model parameters in the decoded architectures are optimized. The overall cascade optimization algorithm for HRS imagery recognition tasks is shown in Algorithm 1.

Algorithm 1 Cascade Optimization

Data: Training dataset
Result: RSNet
Search stage:
Create a search framework for the specific task.
Divide the training dataset into $TrainA$ and $TrainB$.
while *not converged* **do**
  $w \leftarrow w - \nabla_w L_{TrainA}(w, \alpha, \beta)$;
  $\alpha, \beta \leftarrow \alpha, \beta - \nabla_{\alpha,\beta} L_{TrainB}(w, \alpha, \beta)$
end
Decode the RSNet based on α, β.
Training stage:
Merge the training datasets to $Train$ and retrain RSNet.
while *not converged* **do**
  $w \leftarrow w - \nabla_w L_{Train}(w)$
end
Get RSNet with trained parameters.

In the search stage, the training set needs to be divided into two disjoint parts: $TrainA$ for network parameter optimization and $TrainB$ for architecture parameter optimization. Both are trained end-to-end by standard backpropagation on $TrainA$ and $TrainB$ alternately. Following [29], we treat this bilevel optimization problem using gradient descent. On the one hand, based on $TrainA$, we obtain train loss L_{TrainA} and optimize the model parameters via backpropagation. On the other hand, the architecture parameters are optimized on search loss L_{TrainB} based on $TrainB$. Both losses L_{TrainA} and L_{TrainB} are determined not only by the architecture parameters but also by the model parameters in the network. As the search finishes, the searched RSNet architecture can be decoded according to the architecture parameters α and β.

In the training stage, the RSNet with the corresponding recognition module is supposed to be retrained on the whole training data set. The following training process is the same as the traditional deep learning method. After the training stage, RSNet with trained parameters is obtained and the high-performance and efficient RSNet can be applied to the associated HRS imagery recognition task.

D. Decoding Architecture

After the search finishes, the best module architecture is chosen by the ranking module's weights, and the transition architecture is decoded using the Viterbi algorithm [44]. Finally, the trimmed deep neural network is the output of our search framework, which we call RSNet (see Fig. 4). The details of the decoding architecture are as follows.

1) Decoding Module Architecture: We decode the module by first retaining the two strongest predecessors for each block [29], where the strength of an edge is denoted as $max\{\alpha_k | 0 \leq k < 7\}$, noting that no connection operation is excluded here. In other words, the maximum normalized α_k for each edge indicates the strength, and we retain the two

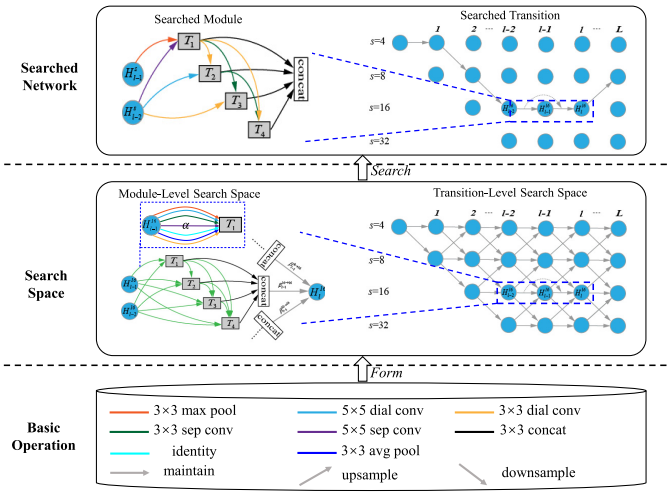


Fig. 7. Relationship from basic operation and search space to the searched network.

strongest edges by ranking. Each edge only keeps the operation O_k with the largest weight.

2) Decoding Transition Architecture: As shown in Fig. 4, each gray arrow indicates the transition probability between two spatial sizes across the adjacent layers. Intuitively, our goal is to find the path with the "maximum probability" from start to finish [30]. We then treat it as a dynamic planning problem, and this path can be decoded using the classic Viterbi algorithm [44]. The details of the algorithm are given in Algorithm 2.

Algorithm 2 Decoding Transition Architecture

Data: transition weights β
Result: The maximum probability transition path
$I_{max} = (i_1, \ldots, i_l, \ldots, i_L)$
Initialize the start node transition probability $p_0^4 \leftarrow 1$.
Initialize the four transition paths $\{I^4, I^8, I^{16}, I^{32}\}$.
while $l < L$ **do**
  for *s in {4, 8, 16, 32}* **do**
    $p_l^s \leftarrow \max\{p_{l-1}^{\frac{s}{2}} \beta_l^{\frac{s}{2} \to s}, p_{l-1}^s \beta_l^{s \to s}, p_{l-1}^{2s} \beta_l^{2s \to s}\}$;
    update the current path I^s.
  end
end
Get the maximum probability transition path $I_{max} \leftarrow I^s$.

In conclusion, a series of basic operations forms the architecture search space, where the best architecture can be searched. The relationship from basic operation and search space to the searched network is shown in Fig. 7.

IV. Experiments

A. Data Set Description

We carried out two groups of experiments to assess the performance of the proposed approach in HRS imagery recognition tasks, compared with the state-of-the-art deep learning methods. For the HRS scene classification task, we applied the proposed framework to the well-known

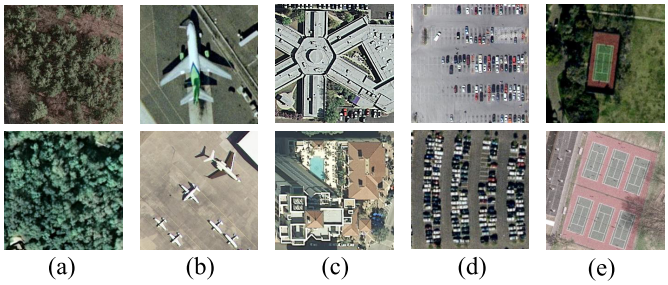


Fig. 8. Examples from the UC Merced data set. (a) Forest. (b) Airplane. (c) Buildings. (d) Parking lot. (e) Tennis court.

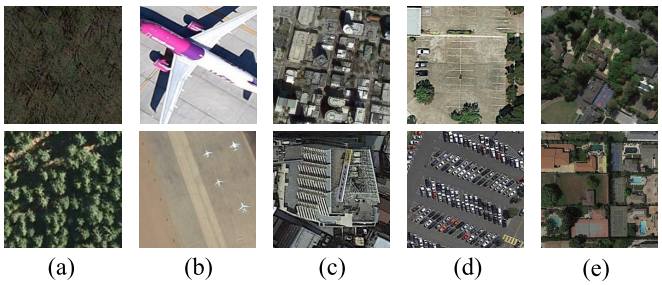


Fig. 9. Examples from the NWPU45 data set. (a) Forest. (b) Airplane. (c) Commercial. (d) Parking lot. (e) Tennis court.

UC Merced data set [45] and the NWPU RESISC45 [46] (NWPU45) data set. For the HRS land-cover classification task, two land-cover data sets were adopted: the Gaofen Image data set (GID) [47] and the 2019 IEEE GRSS Data Fusion Contest Track1 data set (DFCTrack1) [48], [49].

The UC Merced data set was released in 2010 [45] with 21 classes, taken over various regions of the United States. Each class in the UC Merced data set consists of 100 optical aerial images with 256 × 256 pixels and a 0.3-m resolution. This data set has been widely used for the task of remote sensing image scene classification since its release. Some examples from this data set are shown in Fig. 8.

The NWPU45 data set is a large-scale data set for aerial scene classification, which contains 31 500 images divided into 45 scene classes [46]. Each class consists of 700 optical images with a size of 256 × 256 pixels. The spatial resolution varies from about 30 to 0.2 m per pixel for most of the scene classes. Due to its rich image changes and huge data volume, many classic CNNs have been applied to this benchmark data set. Some examples from this data set are shown in Fig. 9.

The GID data set is a pixel-wise land-cover classification data set [47]. It contains 150 high-resolution Gaofen-2 (GF-2) images acquired from more than 60 different cities in China. These images cover a geographic area that exceeds 50 000 km^2. As shown in Fig. 10, five representative land-cover categories are annotated: built-up, farmland, forest, meadow, and water. Each image contains 6800 × 7200 pixels with the pan-sharpened spatial resolution of 1 m. Areas that do not belong to the abovementioned five categories or that cannot be artificially recognized are labeled as unknown. Due to the large within-class diversity and high between-class similarity, this data set is more challenging.

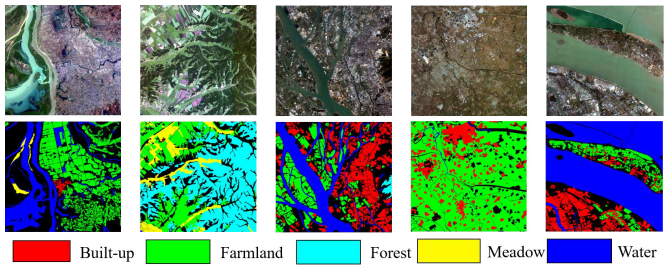


Fig. 10. Visualization of the GID data set.

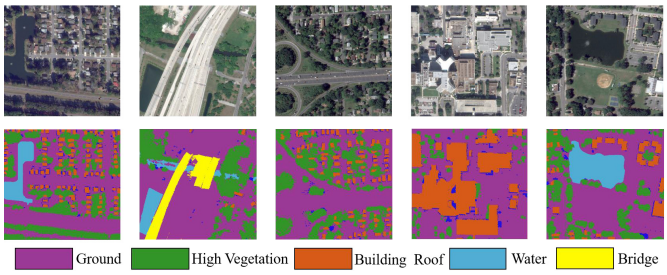


Fig. 11. Visualization of the DFCTrack1 data set.

The DFCTrack1 training data set contains 2783 WorldView-3 images with a size of 1024 × 1024 pixels, together with the corresponding 2-D labeled land-cover classification images and the nDSM images in meters with the same image size. The resolution of this data set is 0.3 m for the RGB images and 1.2 m for the eight-band multispectral images. The validation data set containing 50 images was released without labels; however, it has a public evaluation system online, where the predictions for the validation images can be uploaded to evaluate the performance. As we focus on the 2-D land-cover classification task, the experiments only considered RGB color images and 2-D land-cover annotations. This data set contains five land-cover classes: ground, high vegetation, building roof, water, and bridge. The unrecognized areas are not classified. A visualization of the DFCTrack1 data set is shown in Fig. 11.

B. Evaluation Metrics

Accuracy and efficiency are important metrics for HRS imagery recognition tasks. We use the overall accuracy (OA) and Kappa index to evaluate the classification accuracy, while the pixel accuracy (PA) and mean intersection over union (mIoU) are reported for the land-cover classification tasks. These indices are calculated as follows:

$$\mathrm{OA} = \frac{\sum_{i=1}^{n} X_{ii}}{M} \tag{8}$$

$$\mathrm{Kappa} = \frac{M \sum_{i=1}^{n} X_{ii} - \sum_{i=1}^{n} \left(\sum_{j=1}^{n} X_{ij} \times \sum_{j=1}^{n} X_{ji} \right)}{M^2 - \sum_{i=1}^{n} \left(\sum_{j=1}^{n} X_{ij} \times \sum_{j=1}^{n} X_{ji} \right)} \tag{9}$$

where X_{ij} denotes the number of image class i predicted as class j. Let n be the number of classes and M be the total number of images

$$\mathrm{PA} = \frac{\sum_{i=1}^{n} x_{ii}}{T} \tag{10}$$

$$\mathrm{mIoU} = \frac{1}{n} \sum_{i=1}^{n} \frac{x_{ii}}{\sum_{j=1}^{n} x_{ij} + \sum_{j=1}^{n} x_{ji} - x_{ii}} \tag{11}$$

where x_{ij} denotes the number of pixel class i predicted as class j and T represents the total number of pixels.

For the model efficiency, the theoretical indices and practical efficiency are both evaluated. The theoretical indices are the floating-point operations (FLOPs) and the model parameter number. The FLOPs [50] indicate the computational power required by the model. For the convolutional kernels, we compute these as follows:

$$\mathrm{FLOPs} = 2\mathrm{HW}(C_{in} K^2 + 1) C_{\mathrm{out}} \tag{12}$$

where H, W, and C_{in} are the height, weight, and number of channels of the input feature map, respectively, C_{out} denotes the number of channels of the output feature map, and K denotes the kernel size of the convolution operation. For the fully connected operations, we have

$$\mathrm{FLOPs} = (2N_{\mathrm{in}} - 1) N_{\mathrm{out}} \tag{13}$$

where N_{in} and N_{out} represent the number of input and output channels, respectively. This index is related to the model prediction speed in a runtime environment. For the practical efficiency experiments, we recorded the GPU memory occupation (which is related to the model parameter number and the intermediate variable number) and the prediction speed.

C. Scene Classification Experiments

We used four promising classification CNNs as baselines. Several architectures have been already proposed and tested in [36]. Due to the limitations of GPU memory, we considered a total of $L = 7$ layers in the network and $B = 4$ blocks in a module. A stem layer contains two stride = 2 convolutional layers to reduce the spatial resolution to $s = 4$ as well as increase the filters to filters = 128.

1) Search on the UC Merced Data Set: Five-fold cross-validation was performed, in which the data set was partitioned into five equal subsets. During the search, we conducted our search on four folds of data, half of which was $TrainA$ and the other half was $TrainB$.

The architecture search optimization was conducted for a total of 4000 steps. To make the convergence faster, we set 1000 warm-up steps, where only model parameter w was optimized. The batch size was 16. We used a moderate data augmentation strategy, i.e., 224 × 224 patches were randomly cropped from the 256 × 256 images with random mirroring and rotation, to increase the effective training set size. When optimizing the model parameters w, we used an SGD optimizer with momentum 0.9, a cosine learning rate that decayed from 0.03 to 0.001, and a weight decay of 0.0003. When learning the architecture parameters α and β, we used the Adam optimizer with a learning rate of 0.003 and a weight decay of 0.001. The entire architecture search optimization

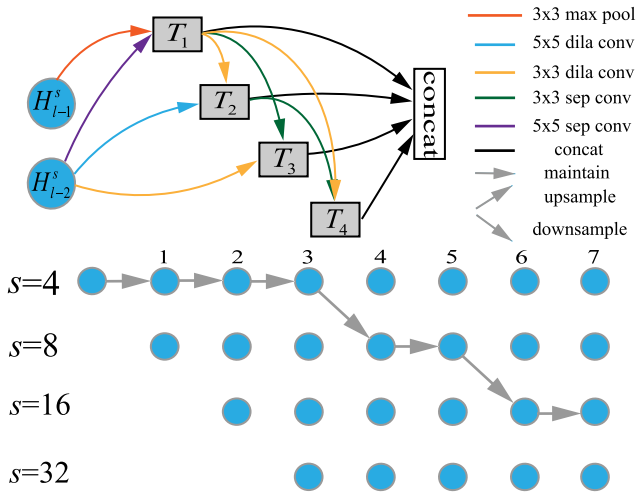


Fig. 12. Decoded architecture searched on the UC Merced data set.

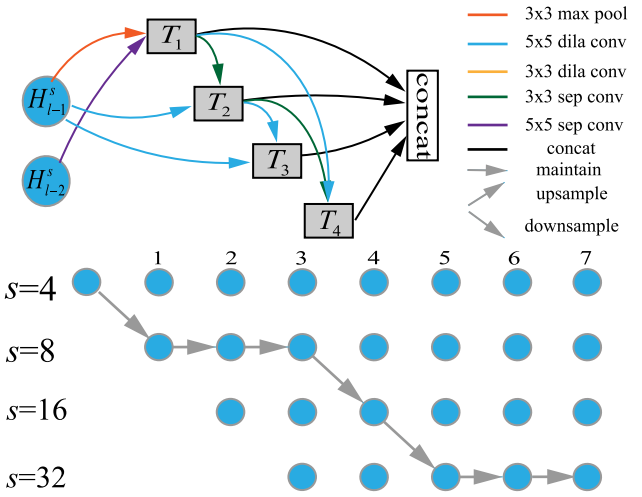


Fig. 13. Decoded architecture searched on the NWPU45 data set.

TABLE II
PERFORMANCE OF THE REFERENCE AND SEARCHED CNNS ON THE UC MERCED DATA SET

Model	Overall Accuracy (%)	Kappa	Design method
AlexNet	90.76 ± 0.72	0.8963 ± 0.0076	manual
VGG16	96.76 ± 1.50	0.9669 ± 0.0177	manual
ResNet34	95.48 ± 0.67	0.9524 ± 0.0071	manual
GoogLeNet	94.19 ± 0.65	0.9419 ± 0.0148	manual
RSNet (UCM)	96.78 ± 0.60	0.9670 ± 0.0045	automatic

TABLE III
PERFORMANCE OF THE REFERENCE AND SEARCHED CNNS ON THE NWPU45 DATA SET

Model	Overall Accuracy (%)	Kappa	Design method
AlexNet	87.911 ± 0.335	0.8140 ± 0.0034	manual
VGG16	93.873 ± 0.439	0.9361 ± 0.0045	manual
ResNet34	92.340 ± 0.627	0.9216 ± 0.0109	manual
GoogLeNet	92.578 ± 0.401	0.9225 ± 0.0041	manual
RSNet (NWPU45)	92.657 ± 0.419	0.9247 ± 0.0041	automatic

took about one day on one P100 GPU accelerator. When the training result converged, the decoded architecture was obtained, as shown in Fig. 12. The searched results show that for the module architecture, three 3×3 separable dilated convolutions were chosen, which indicates the need for a large receptive field. For the transition architecture, the features maintain a high spatial resolution ($s = 4$) at the shallow layers and are successively reduced finally to $s = 16$ in the deep layers. This result matches our perception of the classification network design. With the reduction of the features' spatial correlation, the noise is suppressed, and features of higher abstraction are captured [51].

After we obtained the best architecture, we retrained our network on the UC Merced data set. To ensure the fairness of the experiments, all the compared CNNs were trained from scratch with the same training settings. Through preliminary experiments, we decided on the training strategy and learning rate schedules. We trained the CNNs for 25 200 steps with 16 examples per mini-batch. An SGD optimizer with momentum 0.9 was adopted. We set a base learning rate of 0.03 and gradually reduced this to one-tenth at 16 800 and 22 400 steps.

Table II shows the mean accuracy of the five-fold cross-validation and architecture efficiency compared with the other benchmark classification CNNs [52].

As can be seen in Table II, our searched network achieves the best performance compared with the benchmark CNNs.

2) Search on the NWPU45 Data Set: For the NWPU45 data set, five-fold cross-validation was also applied. The total number of search steps was 12 000, and only the model parameters were optimized during the first 4000 steps. During the search, the training strategy and optimizer settings were the same as for the UC Merced data set. The suitable network architecture was acquired after two days of searching, as shown in Fig. 11. When the search results converged, the decoded architecture was obtained, as shown in Fig. 13.

Compared with the architecture searched on the UC Merced data set (see Fig. 12), this architecture uses three 5×5 dilated convolution operations in each module, which makes the receptive field of the network larger. Moreover, the transition result shows that a lower resolution feature map ($s = 32$) is obtained for the classification. Because the NWPU45 data set has multiple spatial resolutions and more complex scene categories, a wider range of contextual information and deeper semantic information is needed.

To prove the validity of the model we obtained, we also conducted five-fold cross validation on the NWPU45 data set. The data augmentation and optimizer settings were the same as for the UC Merced experiments. We set the number of training iterations as 60 000, and the base learning rate was set to 0.01, which was gradually reduced to one-tenth at 40 000 and 54 000 steps. Table III lists the classification accuracies of the reference and searched CNNs on the NWPU45 data set.

As shown in Table III, the VGG16 model slightly exceeds our model accuracy by 1.22%. We conclude that the NWPU45 data set is more complex, and a deeper network with a larger number of parameters has advantages in fitting a complex data distribution. However, our framework has more potential when we deepen the search layers and sufficient computational power.

TABLE IV
EFFICIENCY OF THE REFERENCE AND SEARCHED CNNS FOR THE SCENE CLASSIFICATION TASK

Model	Params (M)	Memory (M)	GFLOPs	Speed (samples/sec)
AlexNet	61.106	1027	0.724	1215
VGG16	138.377	1857	15.45	151
ResNet34	21.259	945	4.15	447
GoogLeNet	6.646	875	1.51	635
RSNet (UCM)	1.222	621	1.19	936
RSNet (NWPU45)	1.395	765	1.28	775

3) Efficiency Analysis: To analyze the efficiency of the various scene classification models, we set up a series of comparative experiments in a real environment. The model theoretical indices and practical efficiency were tested. As we carried out the performance tests on a single P100 GPU accelerator in the case of double floating-point precision operations, the inputs were 224×224 images with a batch size of 16. All the experiments were performed in the native environment of the PyTorch deep learning framework, without any additional optimization or acceleration.

The results are shown in Table IV, where AlexNet has the fastest predictive speed due to its simple network structure, as shown in Fig. 2(a). However, because of the huge amount of parameters, i.e., 138.177 M, VGG16 requires the most GPU memory occupation of 1857 M and takes the slowest speed of 151 samples/s. The searched networks RSNet (UCM) and RSNet (NWPU45) have the least amount of theoretical parameter size, as well as practical memory usage. In addition, the searched networks possess considerable predictive speeds.

We can conclude from the results of the performance and efficiency experiments that on the one hand, compared with a naive structure such as AlexNet, the searched networks can achieve a better accuracy. On the other hand, compared with a cumbersome network such as VGG16, the searched networks possess higher efficiency. Therefore, the search framework can find a more suitable network for the HRS imagery classification task than handcrafted architectures in an affordable timeframe.

D. Land-Cover Classification Experiments

In the land-cover classification experiments, we took five classic segmentation CNNs as a reference. Due to the limitations of the GPU memory, we considered a total of $L = 9$ layers in the network and $B = 4$ blocks in a module.

1) Search on the GID Data Set: Every image was clipped to a 1024×1024 size to generate a large-scale data set of 6300 images. We kept the same five-fold cross-validation settings as before, and we set 15 000 steps for the architecture search optimization and 5000 for the warm-up steps. Due to the limitations of the GPU memory, we set the batch size as 2 and the crop size as 256×256. The data augmentation and learning rate policy were the same as for the UC Merced data set. We set the model optimizer's initial learning rate as 0.01 and the architecture optimizer's initial learning rate as 0.001. The entire architecture search optimization took about a week on one P100 GPU accelerator. The searched architecture is shown in Fig. 14.

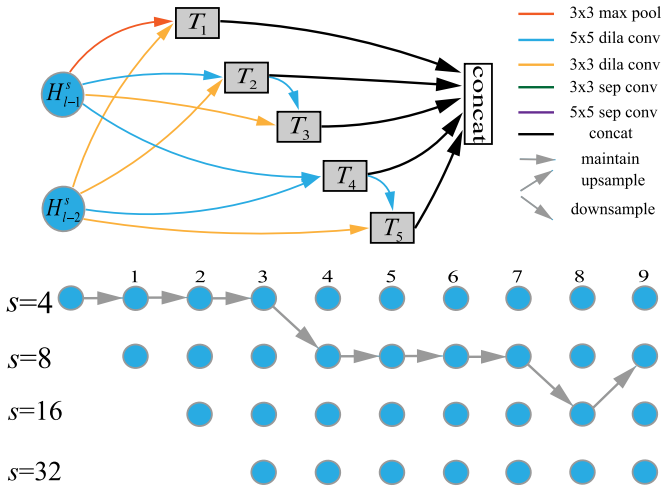


Fig. 14. Decoded architecture searched on the GID data set.

TABLE V
PERFORMANCE OF THE REFERENCE AND SEARCHED CNNS ON THE GID DATA SET

Model	Pixel accuracy (%)	Mean IoU	Design method
FCN8S	93.87 ± 0.593	0.8384 ± 0.0198	manual
PSPNet	96.76 ± 0.144	0.9224 ± 0.0054	manual
Deeplabv3	96.54 ± 0.163	0.9203 ± 0.0068	manual
Deeplabv3+	96.78 ± 0.159	0.9222 ± 0.0071	manual
UNet	97.14 ± 0.136	0.9279± 0.0043	manual
RSNet (GID)	97.19 ± 0.106	0.9354 ± 0.0032	automatic

In terms of the module architecture, most of the operations are dilated convolutions, which once again proves the importance of contextual information for remote sensing image recognition tasks. For the transition architecture, the spatial resolution first reduces twice to $s = 16$ and then increases to $s = 16$. This may be because land-cover classification tasks not only focus on contextual semantic information but also on local details, which helps with semantic segmentation. Compared with the handcrafted architectures shown in Fig. 3, the searched network has a larger receptive field, as well as lighter operations.

In order to verify the performance of the searched network, we retrained the searched network and reference networks under the same conditions. All the networks are trained for 12 000 steps using the SGD optimizer. We used a weight decay of 10^{-4} and a momentum of 0.9. The initial learning rate was set to 0.01, controlled by a "poly" policy with power 0.9. The data augmentation included random flip and rotate. More specifically, we used 512×512 random image crops from the 1024×1024 images in the training data set.

We can conclude from the results listed in Table V that UNet achieves a better performance than the other reference CNNs. This is because UNet has an elegant spatial transition architecture [see Fig. 3(b)]. The contracting path and the expansive path in UNet keep the high-resolution feature map, which learns to assemble more precise localization details [40]. However, keeping the high-resolution feature maps takes up a large amount of memory and also reduces the efficiency

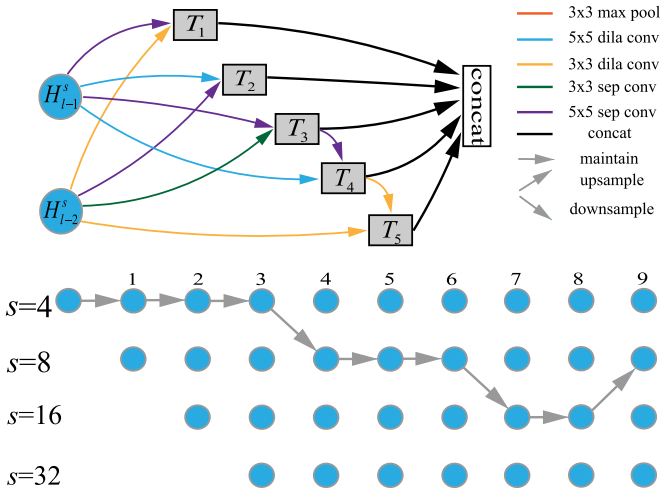


Fig. 15. Decoded architecture searched on the DFCTrack1 data set.

TABLE VI

PERFORMANCE OF THE REFERENCE AND SEARCHED CNNS ON THE DFCTRACK1 DATA SET

Model	Mean IoU	Design method
FCN8S	0.6628	manual
PSPNet	0.6968	manual
DeepLabv3	0.6745	manual
DeepLabv3+	0.6803	manual
UNet	0.7095	manual
RSNet (DFCTrack1)	0.7192	automatic

TABLE VII

EFFICIENCY OF THE REFERENCE AND SEARCHED CNNS FOR THE LAND-COVER CLASSIFICATION TASK

Model	Params (M)	Memory (M)	GFLOPs	Velocity (samples/sec)
FCN8S	134.292	16165	185.59	8
PSPNet	27.501	10639	159.22	5
DeepLabv3	22.257	2849	30.17	21
DeepLabv3+	25.527	3067	34.91	20
UNet	9.852	10547	80.53	10
RSNet (GID)	2.417	2561	10.17	45
RSNet (DFCTrack1)	2.997	2759	9.95	47

(see Table VII). Although the searched network RSNet (GID) does not use contracting and expansive structures, it exceeds the other benchmarks in PA and mean IoU. This proves the effectiveness of our search framework. The framework also has great potential to be improved for mining special structures.

2) Search on the DFCTrack1 Data Set: The DFCTrack1 data set was part of the 2019 Data Fusion Contest provided by the IEEE Geoscience and Remote Sensing Society and is still open for contributions. The images in the test set are publicly available, but its land-cover maps are kept secret. An IoU and elevation error evaluation can be obtained by submitting the predicted results online, and we only focus on mean IoU. Only RGB images and 2-D land-cover annotations were used in the experiments.

For searching, we used similar settings: 12 000 steps for the architecture search optimization and 4000 for the warm-up steps. Images of 256 × 256 were randomly cropped from the 1024 × 1024 images with a batch size of 2. The data augmentation and learning rate policy were the same as for the UC Merced data set. We initially set the model optimizer learning rate as 0.03 and the architecture optimizer learning rate as 0.003. The architecture search took about four days on a P100 GPU accelerator. The searched architecture is shown in Fig. 15.

As can be seen in Fig. 15, the module architecture is composed of 5 × 5 separable convolutions and dilated convolutions, which is similar to RSNet (GID). Moreover, the transition architecture is similar to RSNet (GID) because they are both land-cover classification tasks and there is a tradeoff between localization accuracy and the use of context.

A series of comparative tests was conducted. We retrained our searched network RSNet (DFCTrack1) and the other benchmark CNNs on the DFCTrack1 data set to evaluate their performances. The optimizer settings and data augmentation remained the same as for the GID data set.

We trained all the networks for 24 000 steps. The initial learning rate was set to 0.03 and was divided by 10 at 16 000 and 21 000 steps. The evaluation results are listed in Table VI, where it can be seen that our searched network RSNet (DFCTrack1) achieves the best performance.

3) Efficiency Analysis: We also carried out efficiency analysis experiments with the searched networks and the benchmark segmentation CNNs. All the experiments were performed in the same environment. The inputs were 512×512 images with a batch size of 8.

Table VII lists the efficiency analysis results. The two most notable differences with respect to the classification case (see Table IV) are the high memory occupation and low efficiency. This is mainly because segmentation CNNs focus on semantic segmentation, whereas high-resolution feature maps are required to participate in the calculation. Especially for UNet, although it has a relatively few parameters, i.e., 9.852 M, there are a lot of high-resolution feature map operations in the contracting and expansive path structure, taking up a lot of memory, i.e., 10 547 MB. In addition, UNet also requires the most amount of computation, i.e., 80.53 GFLOPs and has the lowest prediction speed of 10 samples/s. Our searched networks RSNet (GID) and RSNet (DFCTrack1) demonstrate excellent efficiency in the land-cover classification tasks. The amount of parameters is very small, and the memory is less occupied in the runtime environment. With regard to speed, RSNet (GID) and RSNet (DFCTrack1) theoretically require less computation (indicated by GFlops) and, in practice, the prediction speeds are four times faster than UNet and twice as fast as DeepLabv3.

We can conclude from the land-cover classification experiments that the proposed framework can obtain a good tradeoff between accuracy and efficiency. The searched networks RSNet (GID) and RSNet (DFCTrack1) not only achieve the best accuracy but also have a higher efficiency.

V. DISCUSSION

In this article, we have shown that the proposed framework can obtain superior CNN architectures for the HRS scene classification and land-cover classification tasks. The two

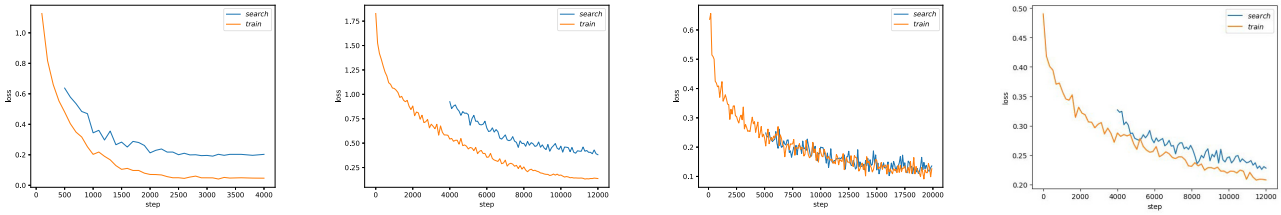


Fig. 16. Losses in the different experiments during the search. (a) Losses in the UC Merced experiment. (b) Losses in the NWPU45 experiment. (c) Losses in the GID experiment. (d) Losses in the DFCTrack1 experiment.

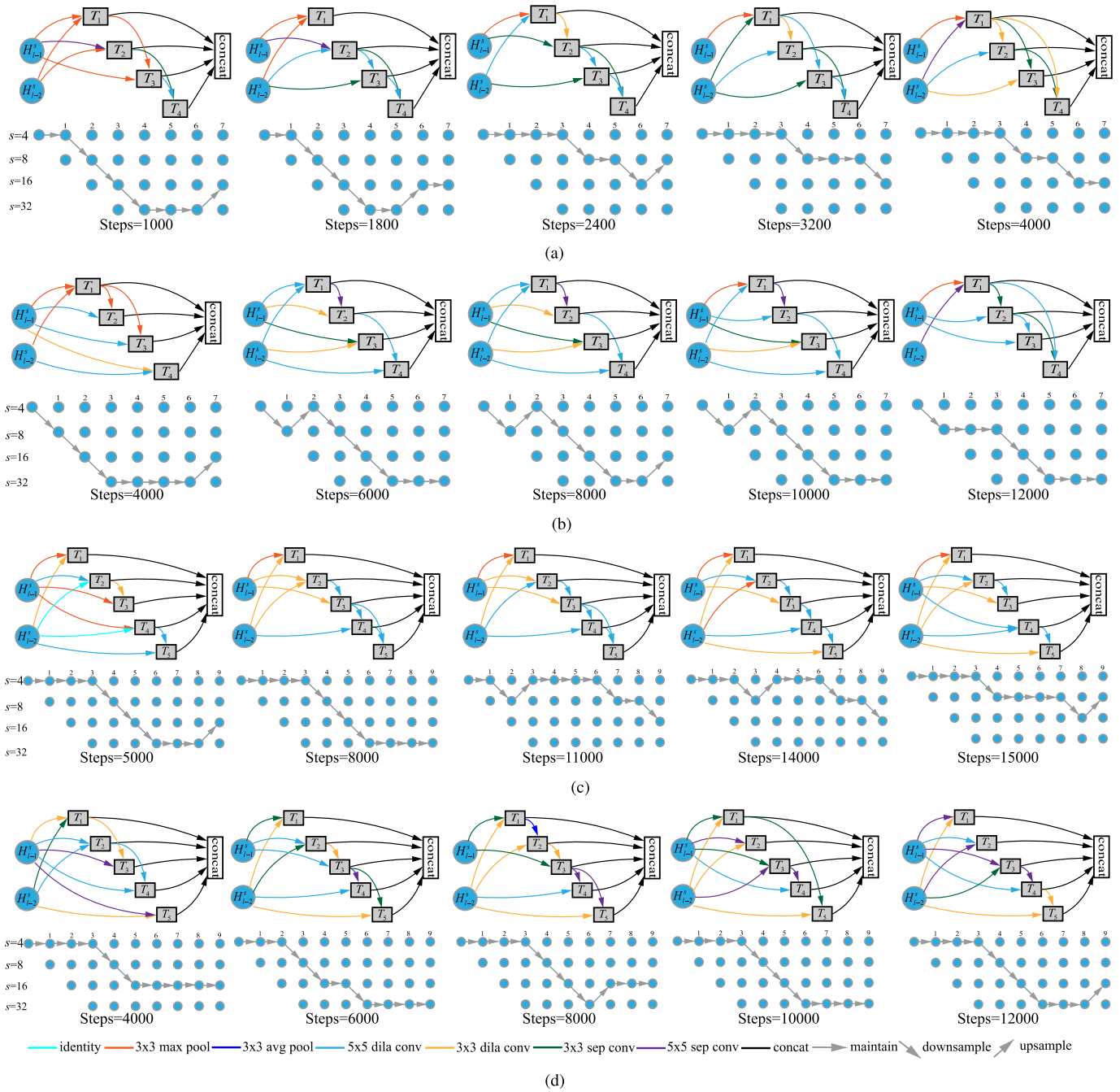


Fig. 17. Search visualization in the different experiments. (a) Search visualization in the UC Merced experiment. (b) Search visualization in the NWPU45 experiment. (c) Search visualization in the GID experiment. (d) Search visualization in the DFCTrack1 experiment.

most notable characteristics of the module structure are the multibranch structure and the dilated convolutions.

1) *Multibranch Structure:* Differing from the VGG module design [see Fig. 2(b)], which stacks convolutions

repeatedly, our searched network tends to select a complex multibranch structure. This may be because HRS images contain multiscale ground objects, such as airplanes and buildings. The current methods for processing multiscale features include multiscale training and designing multibranch structures for a network such as GoogLeNet. Therefore, driven by the UC Merced data set, our search framework chooses multibranch structures to accommodate multiscale features.

2) *Dilated Convolutions:* The classic classification networks, such as AlexNet, VGG, and GoogLeNet, do not consider the use of dilated convolutions. This is mainly due to two reasons. On the one hand, research into classification tasks began early, and these classification networks were put forward before the development of dilated convolutions [38]. On the other hand, these networks were initially designed for natural images obtained by close-range photography. These images have high resolutions but cover a small spatial range, making the included object information simple. Therefore, dense sampling operators, such as normal convolutions, are already enough for natural image classification tasks. However, HRS images are obtained by ultralong distance observation and thus have low resolutions and cover wide ranges. Extracting the relationships between complex ground objects is important for HRS scene classification. Dilated convolutions cover a larger receptive field through sparse sampling. Therefore, driven by the data set, our framework automatically chooses dilated convolutions to acquire long-distance information.

In the search stage, we recorded the training and search losses based on the four recognition experiments. As shown in Fig. 16, we found that the search loss is often higher than the training loss and is more difficult to converge. This may be because the amount of model parameters is much higher than the architecture parameters, which makes the model more inclined to fit the $TrainA$ data set. Compared with the other experiments, the losses in the GID experiment had a smaller difference due to the similar distribution between $TrainA$ and $TrainB$. We list some intermediate results at regular intervals in Fig. 17.

VI. Conclusion

In this article, an RSNet architecture search framework has been proposed to automatically find the most suitable CNNs for HRS image recognition tasks. Specifically, the proposed RSNet search framework employs a two-stage cascade optimization strategy. In the GB architecture search (search stage), a hierarchical basic search space is designed and the architecture as well as model parameters are optimized alternately based on the gradient-descent method. In the task-driven architecture training (training stage), the searched RSNet with the corresponding recognition module is retrained again to optimize the model parameters.

Experiments were carried out on four benchmark data sets, from the two aspects of accuracy and efficiency. The CNNs obtained from our framework achieved a better tradeoff between accuracy and efficiency than the state-of-the-art deep learning architectures of AlexNet, GoogLeNet, VGG16, FCN8S, PSPNet, DeepLabv3, DeepLabv3+, and UNet. The searched RSNets always showed excellent performances. Compared with the advanced handcrafted CNNs, the RSNets showed a higher efficiency and a comparable accuracy. Because our work can provide more lightweight deep neural network architectures that are more suitable for remote sensing imagery recognition, we will extend this to in-orbit satellite data processing [53], [54] and hyperspectral image analysis [55] in further research.

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893.

[2] J. M. Peña-Barragán, M. K. Ngugi, R. E. Plant, and J. Six, "Object-based crop identification using multiple vegetation indices, textural features and crop phenology," *Remote Sens. Environ.*, vol. 115, no. 6, pp. 1301–1316, Jun. 2011.

[3] T. Kobayashi, "Dirichlet-based histogram feature transform for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3278–3285.

[4] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[5] H. Wang, C. Wang, and H. Wu, "Using GF-2 imagery and the conditional random field model for urban forest cover mapping," *Remote Sens. Lett.*, vol. 7, no. 4, pp. 378–387, Apr. 2016.

[6] J. Zhao, Y. Zhong, H. Shu, and L. Zhang, "High-resolution image classification integrating spectral–spatial-location cues by conditional random fields," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4033–4045, Sep. 2016.

[7] Q. Zhu, Y. Zhong, B. Zhao, G.-S. Xia, and L. Zhang, "Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 6, pp. 747–751, Jun. 2016.

[8] B. Zhao, Y. Zhong, L. Zhang, and B. Huang, "The Fisher kernel coding framework for high spatial resolution scene classification," *Remote Sens.*, vol. 8, no. 2, p. 157, Feb. 2016.

[9] Y. Zhong, Q. Zhu, and L. Zhang, "Scene classification based on the multifeature fusion probabilistic topic model for high spatial resolution remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 11, pp. 6207–6222, Nov. 2015.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[12] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[13] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[15] C. Zhang *et al.*, "An object-based convolutional neural network (OCNN) for urban land use classification," *Remote Sens. Environ.*, vol. 216, pp. 57–70, Oct. 2018.

[16] X. Lu, Y. Zhang, Y. Yuan, and Y. Feng, "Gated and axis-concentrated localization network for remote sensing object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 179–192, Jan. 2020.

[17] J. Zhang, P. Zhong, Y. Chen, and S. Li, "$L_{1/2}$-regularized deconvolution network for the representation and restoration of optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2617–2627, May 2014.

[18] X. Lu, X. Zheng, and Y. Yuan, "Remote sensing scene classification by unsupervised representation learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5148–5157, Sep. 2017.

[19] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," 2015, *arXiv:1508.00092*. [Online]. Available: http://arxiv.org/abs/1508.00092

[20] I. Demir *et al.*, "Deepglobe 2018: A challenge to parse the earth through satellite images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2018, pp. 17200–17209.

[21] S. Aich, W. van der Kamp, and I. Stavness, "Semantic binary segmentation using convolutional networks without decoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 182–1824.

[22] M. Dickenson and L. Gueguen, "Rotated rectangles for symbolized building footprint extraction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Salt Lake City, UT, USA, Jun. 2018, pp. 18–22.

[23] T.-S. Kuo, K.-S. Tseng, J.-W. Yan, Y.-C. Liu, and Y.-C.-F. Wang, "Deep aggregation net for land cover classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 252–256.

[24] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.

[25] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," 2016, *arXiv:1611.01578*. [Online]. Available: http://arxiv.org/abs/1611.01578

[26] C. Liu *et al.*, "Progressive neural architecture search," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 19–34.

[27] E. Real *et al.*, "Large-scale evolution of image classifiers," in *Proc. 34th Int. Conf. Mach. Learning-*, vol. 70, 2017, pp. 2902–2911.

[28] R. Miikkulainen *et al.*, "Evolving deep neural networks," in *Artificial Intelligence in the Age of Neural Networks and Brain Computing*. Amsterdam, The Netherlands: Elsevier, 2019, pp. 293–312.

[29] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," 2018, *arXiv:1806.09055*. [Online]. Available: https://arxiv.org/abs/1806.09055

[30] C. Liu *et al.*, "Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 82–92.

[31] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *J. Mach. Learn. Res.*, vol. 20, no. 55, pp. 1–21, 2019.

[32] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evol. Comput.*, vol. 10, no. 2, pp. 99–127, Jun. 2002.

[33] L. Xie and A. Yuille, "Genetic CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1379–1388.

[34] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 4780–4789.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[36] K. Nogueira, O. A. B. Penatti, and J. A. dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognit.*, vol. 61, pp. 539–556, Jan. 2017.

[37] X. Lu, H. Sun, and X. Zheng, "A feature aggregation convolutional neural network for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7894–7906, Oct. 2019.

[38] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[40] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*. Cham, Switzerland: Springer, 2015, pp. 234–241.

[41] A. Bokhovkin and E. Burnaev, "Boundary loss for remote sensing imagery semantic segmentation," in *Advances in Neural Networks—ISNN*, H. Lu, H. Tang, and Z. Wang, Eds. Cham: Springer, 2019, pp. 388–401.

[42] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[43] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*. [Online]. Available: https://arxiv.org/abs/1312.4400

[44] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[45] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2010, pp. 270–279.

[46] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.

[47] X.-Y. Tong *et al.*, "Land-cover classification with high-resolution remote sensing images using transferable deep models," *Remote Sens. Environ.*, vol. 237, Feb. 2020, Art. no. 111322.

[48] M. Bosch, K. Foster, G. Christie, S. Wang, G. D. Hager, and M. Brown, "Semantic stereo for incidental satellite images," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2019, pp. 1524–1532.

[49] B. L. Saux, N. Yokoya, R. Hänsch, M. Brown, and G. Hager, "2019 data fusion contest [technical committees]," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 1, pp. 103–105, Mar. 2019.

[50] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," 2016, *arXiv:1611.06440*. [Online]. Available: http://arxiv.org/abs/1611.06440

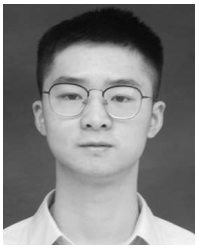
[51] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, Jun. 2015, pp. 1–9.

[52] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," *Acta Ecologica Sinica*, vol. 28, no. 2, pp. 627–635, 2015.

[53] D. Li, M. Wang, Z. Dong, X. Shen, and L. Shi, "Earth observation brain (EOB): An intelligent earth observation system," *Geo-Spatial Inf. Sci.*, vol. 20, no. 2, pp. 134–140, Feb. 2017.

[54] Y. Zhong, W. Li, X. Wang, S. Jin, and L. Zhang, "Satellite-ground integrated destriping network: A new perspective for EO-1 hyperion and Chinese hyperspectral satellite datasets," *Remote Sens. Environ.*, vol. 237, Jun. 2020, Art. no. 111416.

[55] Y. Zhong *et al.*, "Mini-UAV-borne hyperspectral remote sensing: From observation and processing to applications," *IEEE Geosci. Remote Sens. Mag.*, vol. 6, no. 4, pp. 46–62, Dec. 2018.



Junjue Wang (Student Member, IEEE) received the B.S. degree from the School of Geography and Information Engineering, China University of Geosciences, Wuhan, China, in 2019. He is pursuing the master's degree in photogrammetry and remote sensing with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan.

His major research interests are high-resolution remote sensing imagery semantic segmentation and computer vision.

Mr. Wang received the Second Place Prize in the Single-view Semantic 3D Challenge of the 2019 IEEE GRSS Data Fusion Contest.

Yanfei Zhong (Senior Member, IEEE) received the B.S. degree in information engineering and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2002 and 2007, respectively.

Since 2010, he has been a Full Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University. He organized the Intelligent Data Extraction, Analysis and Applications of Remote Sensing (RSIDEA) Research Group. He has published more than 100 research articles in international journals, such as *Remote Sensing of Environment*, the *ISPRS Journal of Photogrammetry and Remote Sensing*, and the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. His research interests include hyperspectral remote sensing information processing, high-resolution remote sensing image understanding, and geoscience interpretation for multisource remote sensing data and applications.

Dr. Zhong is a fellow of the Institution of Engineering and Technology (IET). He was a recipient of the 2016 Best Paper Theoretical Innovation Award from the International Society for Optics and Photonics (SPIE). He received the Second-Place Prize in the 2013 IEEE GRSS Data Fusion Contest and the Single-view Semantic 3-D Challenge of the 2019 IEEE GRSS Data Fusion Contest. He is serving as an Associate Editor for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING and the International Journal of Remote Sensing.

Zhuo Zheng (Graduate Student Member, IEEE) received the B.S. degree from the School of Geography and Information Engineering, China University of Geosciences, Wuhan, China, in 2018. He is pursuing the Ph.D. degree in photogrammetry and remote sensing with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan.

His major research interests are multisource remote sensing imagery panoptic parsing and computer vision.

Mr. Zheng received the Second-Place Prize in the Single-view Semantic 3D Challenge of the 2019 IEEE GRSS Data Fusion Contest.

Ailong Ma (Senior Member, IEEE) received the B.S. degree from the China University of Petroleum, Qingdao, China, in 2010, and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2017.

He is working as a Research Associate with Wuhan University. His major research interests are remote sensing image processing, evolutionary computing, and pattern recognition.

Liangpei Zhang (Fellow, IEEE) received the B.S. degree in physics from Hunan Normal University, Changsha, China, in 1982, the M.S. degree in optics from the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China, in 1988, and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 1998.

He is a Chair Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing (LIESMARS), Wuhan University. He was a Principal Scientist for the China State Key Basic Research Project from 201 to 2016 appointed by the Ministry of National Science and Technology of China to lead the remote sensing program in China. He has published more than 700 research articles and 5 books. He is the Institute for Scientific Information (ISI) highly cited author. He holds 30 patents. His research interests include hyperspectral remote sensing, high-resolution remote sensing, image processing, and artificial intelligence.

Dr. Zhang is a fellow of the Institute of Electrical and Electronics Engineers (IEEE) and the Institution of Engineering and Technology (IET). He was a recipient of the 2010 Best Paper Boeing Award, the 2013 Best Paper ERDAS Award from the American Society of Photogrammetry and Remote Sensing (ASPRS), and the 2016 Best Paper Theoretical Innovation Award from the International Society for Optics and Photonics (SPIE). His research teams received the top three prizes of the IEEE GRSS 2014 Data Fusion Contest, and his students have been selected as the winners or finalists of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS) student paper contest in recent years. He is the Founding Chair of the IEEE Geoscience and Remote Sensing Society (GRSS) Wuhan Chapter. He also serves as an associate editor or an editor of more than ten international journals. He is also serving as an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.