Multibranch Selective Kernel Networks for Hyperspectral Image Classification

T. Alipour-Fard, M. E. Paoletti, Student Member, IEEE, Juan M. Haut, Senior Member, IEEE, H. Arefi, J. Plaza, Senior Member, IEEE, and A. Plaza, Fellow, IEEE

Abstract—Convolutional neural networks (CNNs) have demonstrated excellent performance in hyperspectral image (HSI) classification. However, tuning some critical hyperparameters of a CNN—such as the receptive field (RF) size—presents a major challenge due to the presence of features with different scales in HSIs. Contrary to the conventional design of CNNs, which fixes the RF size, it has been proven that the RF size is modulated by the stimulus and hence, depends on the scene being considered. Such a dilemma has been rarely considered in CNN design. In this letter, a new multibranch selective kernel network (MSKNet) is introduced, in which the input image is convolved using different RF sizes to create multiple branches so that the effect of each branch is adjusted by an attention mechanism according to the input contrast. As a result, our newly developed MSKNet is capable of modeling different scales. Our experimental results, conducted on three widely used HSIs, reveal that the MSKNet can outperform state-of-the-art CNNs in the context of HSI classification problems. The source code of our newly developed MSKNet is available from: https://github.com/mhaut/MSKNet-fully

Index Terms—Convolutional neural networks (CNNs), deep learning, hyperspectral images (HSIs), receptive field (RF), selective kernel networks (SKNets).

I. INTRODUCTION

HYPERSPECTRAL images (HSIs) comprise hundreds of images (at different wavelength channels) for the same area on the surface of the earth. The resulting datacubes provide an opportunity to detect and recognize a wide variety of objects. Classification is one of the most important techniques to extract information from HSIs. However, there are some general challenges for the successful classification of HSIs, including the high dimensionality of the data, the limited availability of training samples (which hampers supervised classification techniques), or the correlation between spectral signatures belonging to different classes. To address these problems, spatial information has been used as a complement to spectral information in HSI classification [1]. In a recent study [2], by considering features in spatial and frequency domains, invariant attribute profiles were adopted to address the different semantic characteristics of the input patches with the same centered pixel label. Convolutional neural networks (CNNs) have become the state-of-the-art of supervised techniques, due to their ability to perform automatic feature generation and also to their generalization power [3]. Despite the advantages of CNNs, the lack of training data and the large number of hyperparameters involved in the training of CNNs (with the subsequent overfitting problem) have become important obstacles to CNN-based HSI classification.

Among CNN’s hyperparameters, the receptive field (RF) plays a very important role [4], [5]. The RF is the region of the input space that affects a particular unit of the CNN. Covering different RFs is important to recognize features with different scales and sizes at a specific layer of the CNN [4]. Fixing the RF size on CNNs is an inefficient assumption. This is because the visual cortex can collect information with different scales at the same processing level [6], [7]. If the RF size is selected to be too large, it can eliminate fine-grained structures. If the RF size is selected to be too small, it can remove coarse-grained structures. In both cases, the HSI classification accuracy can be significantly reduced.

To overcome the disadvantage of using single-branch CNNs, solutions for achieving an optimal architecture have been developed in the computer vision literature [8]–[10]. Specifically, multibranch approaches were introduced to create branches with different RF sizes and combine them to obtain highly informative feature maps. The GoogleNet [8] incorporated an inception module, in which different branches were generated by different RF sizes to aggregate/concatenate information from different scales. The main weakness of GoogleNet lies in the fact that its linear aggregation approach may be insufficient to provide a powerful combination strategy [5]. In addition, various methods such as grouped/depthwise/dilated convolutions have been introduced in order to reduce the number of hyperparameters while incorporating parallel processing strategies [11]. Moreover, Hang et al. [12] designed a two-layer cascade recurrent neural network (where the first layer removes redundant information and the second layer learns in complementary fashion). Afterward, the optimal...
weights for the fusion of the features from the two layers are calculated through a gated recurrent unit.

Another relevant development along the aforementioned lines is the highway network architecture presented in [13], which uses a gating mechanism to modulate the flow of information from different branches and create a deep network. The training of the highway network is difficult, mainly because of the gradient vanishing problem that resulted in the idea of ResNet (by utilizing skip connections) [9], [14]. FractalNet [15] and multilevel ResNet [16] methods were also designed to aggregate different branches recursively. Other than multibranch methods, pyramidal structures (i.e., multiscale systems) have also been considered. Specifically, Zhao and Du [17] proposed a multiscale CNN (MCNN) to extract high-level spatial features for satellite image classification. In the MCNN, an image pyramid was constructed to capture spatial features across scales, and then high-level spatial features were combined with spectral features to train the CNN. Multiscale covariance maps have also been proposed to extract hand-crafted features able to fully exploit the spectral–spatial information present on HSIs [18]. This was done by extracting patches with various sizes around the labeled pixels and then calculating the covariance matrix between the spectral bands. Gao et al. [19] presented a deformation-based convolution strategy for finding the optimal RF size for targets of interest in the image. A remaining challenge with multibranch approaches is how to incorporate a mechanism to aggregate information from different branches in a nonlinear manner. Recently, attention mechanisms have been developed to focus on key parts of the image, discarding irrelevant information [20] (see Fig. 1). Attention mechanisms can be used to recalibrate the feature response and to model adaptive, nonlinear dependencies between feature maps with the gating mechanism.

In this letter, a new multibranch selective kernel network (MSKNet) for HSI classification was developed. MSKNet incorporates an attention mechanism that conducts nonlinear aggregation from different branches, addressing the inefficiency of the traditional (linear) aggregation approach by proposing an end-to-end framework that aggregates branch information by computing the contrast and determining an effective weight by means of an attention mechanism. Our approach has been compared with a traditional CNN, using several HSI benchmark data sets. The experimental results demonstrate the superiority of the proposed approach, which outperforms CNNs with linear branch aggregation.

II. METHODOLOGY

Fig. 2 shows a general overview of the proposed MSKNet model for HSI classification, which is composed of three groups of selective kernel units (SKunits) followed by normalization–activation functions and a fully connected (FC) classifier at the end. The proposed workflow involves three main steps: 1) preprocessing of the HSI data cube; 2) multikernel feature extraction, conducted by 2-D convolutions; and 3) selection of the most descriptive ones through an attention mechanism based on selective kernel layers.

A. Data Preprocessing

Let the HSI data cube be denoted by $X \in \mathbb{R}^{H \times W \times B}$, where $H$ is the height, $W$ is the width, and $B$ is the number of spectral bands. $X$ is split into training and testing sets, considering for each spectral pixel $x_i$ a neighborhood window of size $S \times S \times B$, to provide spectral–spatial information to enhance the feature learning of the proposed model. These HSI patches are sent as input data to the proposed model, which is understood as a mapping function $M_{\theta}: X \rightarrow Y$ that assigns for each $x_i$ its corresponding land cover label $y_i$, obtaining the final classification map $Y \in \mathbb{R}^{H \times W}$ by adjusting its learnable parameters $\theta$.

B. Multikernel Feature Extraction

The architectural body of the proposed model is composed by blocks of SKunits followed by normalization and nonlinear activation functions. In this sense, SKunits implement a multibranch architecture composed by depth-wise separable convolutional layers that exhibit different kernels size $C \times k \times k \times C_m$ (depending on the branch in which they are located) that are intended to extract multikernel spectral-spatial features. Regarding to this, the convolution-based transformations $F^{(l)}: X^{(l-1)} \rightarrow \hat{U}^{(l)}$ and $\hat{F}^{(l)}: X^{(l-1)} \rightarrow \hat{U}^{(l)}$ (indicated as “split” step in Fig. 2) are applied to the original lth SKunit’s input (denoted as $X^{(l-1)} \in \mathbb{R}^{S \times S \times C_m}$). These transformations apply 2-D-grouped convolutions with kernels $3 \times 3$ and $5 \times 5$, respectively, adapting the zero-padding to maintain the spatial dimensions, and being followed by batch normalization (BN) and rectified linear unit (ReLU) as nonlinear activation function, resulting into the feature volumes $\hat{U}^{(l)} \in \mathbb{R}^{S \times S \times C}$ and $\hat{U}^{(l)} \in \mathbb{R}^{S \times S \times C}$.

$$\hat{U}^{(l)} = \hat{F}^{(l)}(X^{(l-1)}) = \text{ReLU}(\beta(W^{(l)} \ast_{C \times 3 \times 3} X^{(l-1)} + \text{bias}^{(l)}))$$

$$\hat{U}^{(l)} = \hat{F}^{(l)}(X^{(l-1)}) = \text{ReLU}(\beta(W^{(l)} \ast_{C \times 5 \times 5} X^{(l-1)} + \text{bias}^{(l)}))$$

where $\ast_{C \times k \times k}$ denotes the convolutional operation composed by $C$ filters with RF $k \times k$, $W^{(l)}$ and $\text{bias}^{(l)}$ are the weights and biases of the each convolutional layer that belongs to the lth SKunit, and $\beta$ is the BN. Each element of $\ast_{C \times k \times k}$ operation is obtained as

$$\hat{u}^{(l)}_{i,j,c} = \sum_{l=0}^{C_m} \sum_{j=0}^{k-1} \sum_{i=0}^{k-1} (w^{(l)}_{i,j,c} \ast x^{(l-1)}_{i+l,j+i,c}) + \text{bias}^{(l)}$$

Convolutional layers are defined by an n-dimensional kernel $C_{out} \times k \times k \times C_m$, where $C_{in}$ is the number of feature maps from the input volume and $C_{out}$ is the number of filters with RF $k \times k$. 

Authorized licensed use limited to: Antonio Plaza. Downloaded on May 14,2020 at 07:17:38 UTC from IEEE Xplore. Restrictions apply.
where \(u_{ij}^{(l)}\) is the \((i, j)\)th element of the \(c\)th feature map of volumes \(\hat{U}^{(l)}\) or \(\hat{U}^{(l)}\), obtained at the \(l\)th SKunit.

### C. Selective Kernel Attention Mechanism

Once the multikernel feature extraction is performed, control gates are required to regulate the information flow between the different branches, to combine and enhance the most descriptive features in the subsequent layers. The final goal is to allow our model to adaptively adjust the RF \(k \times k\) to handle different scales of information. The selective kernel strategy (based on attention mechanism) has been implemented in two main steps: multikernel data fusion and attention-based selection (see Fig. 2).

At first, each SKunit fuses volumes \(\hat{U}^{(l)}\) and \(\hat{U}^{(l)}\) via element-wise summation, obtaining as a result \(\hat{U}^{(l)} \in \mathbb{R}^{S \times S \times C}\) as \(U^{(l)} = \hat{U}^{(l)} + \hat{U}^{(l)}\). Then, a global average pooling (GAP) is performed to generate the feature response vector (FRV) with the channel-wise statistics of the data, reducing the spatial dimension of \(U^{(l)}\) to \(s^{(l)} \in \mathbb{R}^{C}\) by taking the average of \(S \times S\) spatial elements at each channel \(c\)

\[
s_c^{(l)} = \frac{1}{S^2} \sum_{i=1}^{S} \sum_{j=1}^{S} u_{ij}^{(l)c}.
\]

The obtained FRV vector \(s\) is compacted by an FC layer defined by weights \(W_{fc}^{(l)} \in \mathbb{R}^{d \times C}\) followed by BN and ReLU, in order to obtain the neural activations of the different channel-features, enabling their guidance for adaptive kernel selections. In this sense, the feature weights vector (FWV) \(z^{(l)} \in \mathbb{R}^{d}\) can be defined as \(z^{(l)} = \text{ReLU}(b(W_{fc}^{(l)} \cdot s))\). Parameter \(d\) plays an important role in the performance of the SKunit, as its underestimation significantly reduces the efficiency of the MSKNet. For this reason, \(r\) is considered to control the compression rate of \(z^{(l)}\), being determined by \(d = \max((c/r), L)\), where \(L = 32\) is the minimum value of \(d\). Finally, to achieve an adaptive adjustment, a control gate is designed by means of an attention mechanism to select the most important regions of the FWV \(z^{(l)}\). This is done by applying one FC layer per SKunit’s branch and computing the softmax function to obtain the effective FWV (EFWV) as

\[
a_c^{(l)} = \frac{e^{A_c^{(l)}x^{(l)}}}{e^{A_c^{(l)}x^{(l)}} + e^{B_c^{(l)}x^{(l)}}}, \quad b_c^{(l)} = \frac{e^{B_c^{(l)}x^{(l)}}}{e^{A_c^{(l)}x^{(l)}} + e^{B_c^{(l)}x^{(l)}}}.
\]

where \(A_c^{(l)}, B_c^{(l)} \in \mathbb{R}^{C \times d}\) and \(a_c^{(l)}, b_c^{(l)} \in \mathbb{R}^{C}\) are the soft attention vectors of \(\hat{U}^{(l)}\) and \(\hat{U}^{(l)}\). Then, the final recalibrated feature map \(X^{(l)}\) in the \(l\)th SKunit is obtained by applying the attention-based vectors \(a_c^{(l)}\) and \(b_c^{(l)}\) along the channel dimension

\[
X^{(l)c} = a_c^{(l)} \cdot \hat{U}^{(l)} + b_c^{(l)} \cdot \hat{U}^{(l)}, \quad \text{subject to } a_c^{(l)} + b_c^{(l)} = 1.
\]

Resulting \(X^{(l)}\) is fed to the next SKunit until the end of the network is reached, where a FC layer is applied to perform the final classification. Fig. 2 provides a detailed summary of the proposed model in terms of its layers, kernel size, and output map dimensions. All weights are randomly initialized and trained using the back-propagation algorithm with Adam optimizer and cross-entropy loss. We use minibatches of size 100, and train the network for 200 epochs without data augmentation.

### III. Experimental Results

#### A. Hyperspectral Data Sets

Three real HSI data sets have been considered in our experiments. The first one is the Indian Pines (IP) captured by
TABLE II

<table>
<thead>
<tr>
<th></th>
<th>INDIAN PINEIS</th>
<th>UNIVERSITY OF PAVIA</th>
<th>UNIVERSITY OF HOUSTON</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSDNet</td>
<td>CNN</td>
<td>MSKNet</td>
</tr>
<tr>
<td>Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>80.15±0.20</td>
<td>80.00±0.02</td>
<td>81.83±0.23</td>
</tr>
<tr>
<td>2</td>
<td>69.0±6.73</td>
<td>78.37±5.17</td>
<td>85.98±1.17</td>
</tr>
<tr>
<td>3</td>
<td>80.6±2.16</td>
<td>84.42±4.88</td>
<td>88.85±5.39</td>
</tr>
<tr>
<td>4</td>
<td>84.3±5.74</td>
<td>94.1±9.38</td>
<td>97.50±3.48</td>
</tr>
<tr>
<td>5</td>
<td>57.9±6.41</td>
<td>50.45±10.30</td>
<td>67.67±6.33</td>
</tr>
<tr>
<td>6</td>
<td>90.3±3.36</td>
<td>83.49±3.04</td>
<td>91.95±5.63</td>
</tr>
<tr>
<td>7</td>
<td>0±0.00</td>
<td>0±0.00</td>
<td>0±0.00</td>
</tr>
<tr>
<td>8</td>
<td>96.6±4.23</td>
<td>84.07±4.24</td>
<td>95.1±4.06</td>
</tr>
<tr>
<td>9</td>
<td>38.8±4.69</td>
<td>80.6±2.00</td>
<td>65.0±1.70</td>
</tr>
<tr>
<td>10</td>
<td>36.7±7.89</td>
<td>79.72±6.48</td>
<td>85.55±3.82</td>
</tr>
<tr>
<td>11</td>
<td>79.02±4.30</td>
<td>86.99±1.28</td>
<td>84.91±2.33</td>
</tr>
<tr>
<td>12</td>
<td>49.9±4.53</td>
<td>48.9±15.99</td>
<td>54.43±9.29</td>
</tr>
<tr>
<td>13</td>
<td>93.5±4.15</td>
<td>97.92±3.26</td>
<td>95.96±3.57</td>
</tr>
<tr>
<td>14</td>
<td>60.0±4.06</td>
<td>91.01±3.82</td>
<td>95.57±1.34</td>
</tr>
<tr>
<td>15</td>
<td>24.9±4.28</td>
<td>39.02±2.45</td>
<td>65.6±14.07</td>
</tr>
<tr>
<td>16</td>
<td>40.1±0.78</td>
<td>88.77±3.41</td>
<td>64.0±13.85</td>
</tr>
</tbody>
</table>

**OA** 66.9±1.38  65.6±1.32  81.79±1.92  85.64±1.36  89.04±1.61  90.66±1.72  71.57±2.25  85.5±1.4  88.28±1.33

**AA** 55.3±1.09  66.13±2.57  71.4±2.09  82.4±1.16  86.93±1.93  88.09±2.33  71.69±2.76  85.9±1.39  88.7±1.41

**Kappa** 61.54±1.89  73.07±2.17  79.2±1.91  80.72±1.59  85.6±2.23  87.36±1.08  62.85±3.17  83.24±2.32  87.28±1.14

- **OA**: Overall Accuracy
- **AA**: Average Accuracy
- **Kappa**: Kappa coefficient

The superiority of the proposed method over the other two methods is clear from Fig. 3, especially when the number of training samples is limited. In the IP data set, the OA obtained for training percentages of 1%, 3%, and 5% is poor due to the overfitting problem. However, when the percentage increases, the proposed method achieves OA values higher than 95% (with smaller variance). Focusing on Fig. 3(b), the MSKNet method is superior to the traditional CNN method when the training data are limited. Specifically, the OA of MSKNet rises above 98% with 3% of training samples. Another interesting remark concerning Fig. 3(b) is that, when the training percentage is just 1%, the variance of the MSKNet is significantly lower than the other two methods. It is apparent that the performance of MSKNet consistently yields higher OAs than those obtained by the MSDNet and CNN.

Table II reports the classification accuracies of proposed method, MSDNet and traditional CNN for the considered data sets. Focusing on the UP data set, the OA, AA, and Kappa of our MSKNet are higher than those reported by MSDNet and CNN, where MSDNet is the worst classifier and CNN the thinnest method in terms of parameters. Remarkably, the accuracies obtained for most classes by the proposed method are also higher than those obtained by the MSDNet and CNN. Regarding the UP data set, the OA of the proposed method is increased to 90.66%, while the number of model parameters and the standard variation of the OA is the lowest. Most of the results obtained from two previous data sets hold for the UH data set; as can be seen, the OA, AA, and Kappa are increased to 88.28%, 88.87%, and 0.8728, respectively.
work that exploits different scales of information present in the quality of the proposed method is increased compared to the traditional CNN. As it can be observed, by incorporating the SKunit, the visual performance of the proposed method and the traditional CNN on the UP image.

Other different attention mechanisms into our model. CNN architectures such as ResNet. We also plan to incorporate quantitatively and also in terms of visual performance. In the three real HSIs (with very limited training samples) outperformed the parking area in the bottom leftmost part of the image (area with trees and asphalt) in Fig. 4(c) and (d), we can observe that the regions obtained by our method are better connected.

**IV. CONCLUSION**

In this letter, we presented a new HSI classification framework that exploits different scales of information present in the input data. Our newly developed method, called MSKNet, creates different branches by convolving the input HSI data cubes with different kernel sizes, and then aggregates the resulting information using a nonlinear attention mechanism. The classification results obtained by the proposed method on three real HSIs (with very limited training samples) outperform those achieved by the traditional CNN and the MSDNet quantitatively and also in terms of visual performance. In the future, we will combine our model with more sophisticated CNN architectures such as ResNet. We also plan to incorporate other different attention mechanisms into our model.

**REFERENCES**


Fig. 4. Classification maps obtained for the UP scene (using 1% of the available labeled samples). The obtained OAs are shown in brackets. (a) Ground-truth. (b) MSDNet (86.44%). (c) CNN (91.33%). (d) MSKNet (95.09%).