Article

Spectral-Swin Transformer with Spatial Feature Extraction Enhancement for Hyperspectral Image Classification

Yinbin Peng 1, Jiansi Ren 1,2,*, Jiamei Wang 1 and Meilin Shi 1

1 School of Computer Science, China University of Geosciences, Wuhan 430078, China
2 Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430078, China
* Correspondence: renjsv@cug.edu.cn

Abstract: Hyperspectral image classification (HSI) has rich applications in several fields. In the past few years, convolutional neural network (CNN)-based models have demonstrated great performance in HSI classification. However, CNNs are inadequate in capturing long-range dependencies, while it is possible to think of the spectral dimension of HSI as long sequence information. More and more researchers are focusing their attention on transformer which is good at processing sequential data. In this paper, a spectral shifted window self-attention based transformer (SSWT) backbone network is proposed. It is able to improve the extraction of local features compared to the classical transformer. In addition, spatial feature extraction module (SFE) and spatial position encoding (SPE) are designed to enhance the spatial feature extraction of the transformer. The spatial feature extraction module is proposed to address the deficiency of transformer in the capture of spatial features. The loss of spatial structure of HSI data after inputting transformer is supplemented by proposed spatial position encoding. On three public datasets, we ran extensive experiments and contrasted the proposed model with a number of powerful deep learning models. The outcomes demonstrate that our suggested approach is efficient and that the proposed model performs better than other advanced models.

Keywords: transformer; shifted window; spatial feature extraction (SFE); spatial position encoding (SPE); hyperspectral image (HSI) classification

1. Introduction

Because of the rapid advancement of hyperspectral sensors, the resolution and accuracy of hyperspectral images (HSI) have also increased greatly. HSI contains a wealth of spectral information, collecting hundreds of bands of electron spectrum at each pixel. Its rich information allows for excellent performance in classifying HSI, and thus its application has great potential in several fields such as precision agriculture [1] and Jabir et al. [2] used machine learning algorithm for weed detection, medical imaging [3], object detection [4], urban planning [5], environment monitoring [6], mineral exploration [7], dimensionality reduction [8] and military detection [9].

Numerous conventional machine learning methods have been used to the classification of HSI in the past decade or so, such as K-nearest neighbors (KNN) [10], support vector machines (SVM) [11–14], random forests [15,16]. Navarro et al. [17] used neural machine learning algorithm for weed detection, medical imaging [3], object detection [4], urban planning [5], environment monitoring [6], mineral exploration [7], dimensionality reduction [8] and military detection [9].

Deep learning have developed rapidly in recent years, and their powerful fitting ability can extract features from multivariate data. Inspired by
this, the designed deep learning models have proposed in HSI classification tasks, such as recurrent neural network (RNN) [20–22], convolutional neural network (CNN) [23–28], graph convolutional network (GCN) [29,30], capsule network (CapsNet) [31,32], long short term memory (LSTM) networks [33–35]. Although these deep learning models show good performance in several different domains, they have certain shortcomings in HSI classification tasks.

For CNNs, which are good at natural image tasks, Its benefit is that the image’s spatial information can be extracted during the convolution operation. HSI-CNN [36] stacks multi-dimensional data from HSI into two-dimensional data and then extracts features efficiently. 2D-CNN [37] can capture spatial features in HSI data to improve classification accuracy. However, HSI has rich information in the spectral dimension, and if it is not exploited, the performance of the model is bound to be difficult to break through. Although the advent of 3D-CNN [38–41] enables the extraction of both spatial and spectral features, the convolution operation is localized, so the extracted features lack the mining and representation of the global Information.

Recently, transformer has evolved rapidly and shown good performance when performing tasks like natural language processing. Based on its self-attention mechanism, it is very good at processing long sequential information and extracting global relations. Vision transformer (ViT) [42] makes it perform well in several vision domains by dividing images into patches and then inputting them into the model. Swin-transformer [43] enhances the capability of local feature extraction by dividing the image into windows and performing multi-head self-attention (MSA) separately within the windows, and then enabling the exchange of information between the windows by shifting the windows. It improves the accuracy in natural image processing tasks and effectively reduces the computational effort in the processing of high-resolution images. Due to transformer’s outstanding capabilities for natural image processing, more and more studies are applying it to the classification of HSI [44–50]. However, if ViT is applied directly to the HSI classification, there will be some problems that will limit the performance improvement, specifically as follows.

(1) The transformer performs well at handling sequence data (spectral dimension information), but lacks the use of spatial dimension information.

(2) The multi-head self-attention (MSA) of transformer is adept at resolving the global dependencies of spectral information, but it is usually difficult to capture the relationships for local information.

(3) Existing transformer models usually map the image to linear data to be able to input into the transformer model. Such an operation would destroy the spatial structure of HSI.

HSI can be regarded as a sequence in the spectral dimension, and the transform is effective at handling sequence information, so the transformer model is suitable for HSI classification. The research in this paper is based on transformer and considers the above mentioned shortcomings to design a new model, called spectral-swin transformer (SSWT) with spatial feature extraction enhancement, and apply it in HSI classification. Inspired by swin-transformer and the characteristics of HSI data which contain a great deal of information in the spectral dimension, we design a method of dividing and shifting windows in the spectral dimension. MSA is performed within each window separately, aiming to improve the disadvantage of transformer to extract local features. We also design two modules to enhance model’s spatial feature extraction. In summary, the following are the contributions of this paper.

(1) Based on the characteristics of HSI data, a spectral dimensional shifted window multi-head self-attention is designed. It enhances the model’s capacity to capture local information and can achieve multi-scale effect by changing the size of the window.

(2) A spatial feature extraction module based on spatial attention mechanism is designed to improve the model’s ability to characterize spatial features.

(3) A spatial position encoding is designed before each transformer encoder to deal with the lack of spatial structure of the data after mapping to linear.
Three publicly accessible HSI datasets are used to test the proposed model, which is compared with advanced deep learning models. The proposed model is extremely competitive.

The rest of this paper is organized as follows: Section 2 discusses the related work on HSI classification using deep learning, which includes transformer. Section 3 describes the proposed model and the design method for each component. Section 4 presents the three HSI datasets, as well as the experimental setup, results, corresponding analysis. Section 5 concludes with a summary and outlook of the full paper.

2. Related Work

2.1. Deep-Learning-Based Methods for HSI Classification

Deep learning has developed quickly, more and more researchers are using deep learning methods (e.g., RNNs, CNNs, GCNs, CapsNet, LSTM) to the classification tasks of HSI [20,22,23,29–31,33,34]. Mei et al. [51] constructed a network based on bidirectional long short-term memory (Bi-LSTM) for HSI classification. Zhu et al. [52] proposed an end-to-end residual spectral–spatial attention network (RSSAN) for HSI classification, which consists of spectral and spatial attention modules for spectral band and spatial information adaptive selection. Song et al. [53] created a deep feature fusion network (DFFN) to solve the negative effects of excessively increasing network depth.

Due to CNN’s excellent capability of taking the local spatial context information and it’s outstanding capabilities in natural picture processing, many CNN-based HSI classification models have emerged. For example, Hang et al. [54] proposed two CNN sub-networks based on the attention mechanism for extracting the spectral and spatial features of HSI, respectively. Chakraborty et al. [55] designed a wavelet CNN that uses layers of wavelet transforms to display spectral features. Gong et al. [56] proposed a hybrid model that combines 2D-CNN and 3D-CNN in order to include more in-depth spatial and spectral features while using fewer learning samples. Hamida et al. [57] introduced a new 3-D DL method that permits the processing of both spectral and spatial information simultaneously.

However, each of these deep learning approaches has some respective drawbacks that can limit the model performance when processing HSI classification tasks. For CNN, it is good at handling two-dimensional spatial features, but since the data of HSI is stereoscopic and contains a large amount of information in the spectral dimension. It’s possible that CNN will have trouble extracting the spectral features. Moreover, although CNNs have achieved good results by relying on their local feature focus, the inability to deal with global dependencies limits their performance when processing spectral information in the form of long sequences. These shortcomings will be addressed in the transformer.

2.2. Vision Transformers for Image Classification

With the increasing use of transformers in computer vision, researchers have begun to consider images in terms of sequential data, such as ViT [42] and Swin-transformer [43] etc. Fang et al. [58] proposed MSG-Transformer, which presents a specialized token in each region as a messenger (MSG). Information can be transmitted flexibly among areas and computational cost is decreased by manipulating these MSG tokens. Guo et al. [59] proposed CMT, which combines the advantages of CNN and ViT, a new hybrid transformer-based network that captures long-range dependencies using transformers and extracts local information using CNN. Chen et al. [60] designed MobileNet and transformer in parallel, connected in the middle by a two-way bridge. This structure benefits from MobileNet for local processing and Transformer for global communication.

An increasing number of researchers are applying transformer to HSI classification tasks. Hong et al. [44] proposed a model called SpectralFormer (SF) for HSI classification, which divides neighboring bands into the same token for learning features and connects encoder blocks across layers, but the spatial information in HSI was not considered. Sun et al. [45] proposed the Spectral-Spatial Feature Tokenization Transformer (SSFTT) to capture high-level semantic information and spectral-spatial features, resulting in a large
performance improvement. Ayas et al. [61] designs a spectral-swin module in front of the swin transformer, which extracts spatial and spectral features and fuses them with Conv 2-D operation and Conv 3-D operation, respectively. Mei et al. [47] proposed the Group-Aware Hierarchical Transformer (GAHT) to restrict the MSA to a local spatial-spectral range by using a new group pixel embedding module, which enables the model to have improved capability of local feature extraction. Yang et al. [46] proposed a hyperspectral image transformer (HiT) classification network that captures subtle spectral differences and conveys local spatial context information by embedding convolutional operations in the transformer structure, however it is not effective in capturing local spectral features. Transformer is increasingly used in the field of HSI classification and we believe it has great potential for the future.

3. Methodology

In this section, we will introduce the proposed spectral-swin transformer (SSWT) with spatial feature extraction enhancement, which will be described in four aspects: the overall architecture, spatial feature extraction module (SFE), spatial position encoding (SPE), and spectral swin-transformer module.

3.1. Overall Architecture

In this paper, we design a new transformer-based method SSWT for the HSI classification. SSWT consists of two major Components for solving the challenges in HSI classification, namely, spatial feature extraction module (SFE) and spectral swin (S-Swin) transformer module. An overview of the proposed SSWT for the HSI classification is shown in Figure 1. The input to the model is a patch of HSI, the data is first input to SFE to perform initial spatial feature extraction, the module consists of convolution layers and spatial attention. In Section 3.2, it is explained in further detail. The data is then flattened and entered into the s-swin transformer module. A spatial position encoding is added in front of each s-swin transformer layer to add spatial structure to the data. This part will be described in Section 3.3. The s-swin transformer module uses the spectral-swin self attention, which will be introduced in Section 3.4. The final classification results are obtained by linear layers.

![Figure 1. Overall structure of the proposed SSWT model for HSI classification.](image)

3.2. Spatial Feature Extraction Module

Due to transformer’s lack of ability in handling spatial information and local features, we designed a spatial feature extraction (SFE) module to compensate. It consists of two parts, the first one consists of convolutional layers to preliminary extraction of spatial features and batch normalization to prevent overfitting. The second part is a spatial...
attention mechanism, which aims to enable the model to learn the important spatial locations in the data. The structure of SFE is shown in Figure 1.

For the input HSI patch cube $I \in \mathbb{R}^{H \times W \times C}$, where $H \times W$ is the spatial size and $C$ is the number of spectral bands. Each pixel space in $I$ consists of $C$ spectral dimensions and forms a one-hot category vector $S = [s_1, s_2, s_3, \ldots, s_n] \in \mathbb{R}^{1 \times 1 \times n}$, where $n$ is the number of ground object classes.

Firstly, the spatial features of HSI are initially extracted by CNN layers, and the formula is shown as follows:

$$X = GELU(BN(Conv(I)))$$

where $Conv(\cdot)$ represents the convolution layer. $BN(\cdot)$ represents batch normalization. $GELU(\cdot)$ denotes the activation function. The formula for the convolution layer is shown below:

$$Conv(I) = \bigoplus_{j=0}^{J} (I * W_j^{r_1 \times r_2} + b_j)$$

where I is the input, $J$ is the number of convolution kernels, $W_j^{r_1 \times r_2}$ is the $j$th convolution kernel with the size of $r_1 \times r_2$, and $b_j$ is the $j$th bias. $\bigoplus$ denotes concatenation, and $*$ is convolution operation.

Then, the model may learn important places in the data thanks to a spatial attention mechanism (SA). The structure of SA is shown in Figure 2. For an intermediate feature map $X \in \mathbb{R}^{H' \times W' \times C}$ ($H' \times W'$ is the spatial size of $X$), the process of SA is shown in the following formula:

$$S_M = MaxPooling(X)$$

$$S_A = AvgPooling(X)$$

$$X_{SA} = \sigma(Conv(Conv(S_M, S_A)) \odot X)$$

MaxPooling and AvgPooling are global maximum pooling and global average pooling along the channel direction. Concat denotes concatenation in the channel direction. $\sigma$ is activation function. $\odot$ denotes the elementwise multiplication.

![Figure 2. The structure of the spatial attention in SFE.](image)

3.3. Spatial Position Encoding

The HSI of the input transformer is mapped to linear data, which can damage the spatial structure of HSI. To describe the relative spatial positions between pixels and to maintain the rotational invariance of samples, a spatial position encoding (SPE) is added before each transformer module.

The input to HSI classification is a patch of a region, but only the label of the center pixel is the target of classification. The surrounding pixels can provide spatial information for the classification of center pixel, and their importance tends to decrease with the distance
to the center. SPE is to learn such a center-important position encoding. The pixel positions of a patch is defined as follows.

$$\text{pos}(x_i, y_i) = |x_i - x_c| + |y_i - y_c| + 1$$  (6)

where $(x_c, y_c)$ denotes the coordinate of central position of the sample, that is the pixel to be classified. $(x_i, y_i)$ denotes the coordinates of other pixels in the sample. The visualization of SPE when the spatial size of the sample is $7 \times 7$ can be seen in Figure 3. The pixel in the central position is unique and most important, and the other pixels are given different position encoding depending on the distance from the center.

To flexibly represent the spatial structure in HSI, the learnable position encoding are embedded in the data:

$$Y = X + \text{spe}(P)$$  (7)

where $X$ is the HSI data, and $P$ represents the position matrix (like Figure 3) constructed according to Equation (6). $\text{spe}(\cdot)$ is a learnable array that takes the position matrix as a subscript to get the final spatial position encoding. Finally, the position encoding is added to the HSI data.

![Figure 3. SPE in a sample with the spatial size is $7 \times 7$.](image)

### 3.4. Spectral Swin-Transformer Module

The structure of the spectral swin-transformer (S-SwinT) module is shown in Figure 1. Transformer is good at processing long dependencies and lacks the ability to extract local features. Inspired by swin-transformer [43], window-based multi-head self-attention (MSA) is used in our model. Because the input of HSI is a patch which is usually small in spatial size, it cannot divide the window in space as Swin-T does. Considering the rich data of HSI in the spectral dimension, a window of spectral shift was designed for MSA, called spectral window multi-head self-attention (S-W-MSA) and spectral shifted window multi-head self-attention (S-SW-MSA). MSA within windows can effectively improve local feature capturing, and window shifting allows information to be exchanged in the neighboring windows. MSA can be expressed by the following formula:

$$Z = \text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V$$  (8)

$$\psi = \text{Concat}(Z_1, Z_2, \ldots, Z_h)W$$  (9)

$Q, K, V$ are matrices mapped from the input matrices called queries, keys and values. $d_K$ is the dimension of $K$. The attention scores are calculated from $Q$ and $K$. $h$ is the head number of MSA, $W$ denotes the output mapping matrix., and $\psi$ represents the output of MSA.

As shown in Figure 4, the size of input is assumed to be $H \times W \times C$, where $H \times W$ is the space size and $C$ is the number of spectral bands. Given that all windows’ size is set to
C/4, the window is divided uniformly for the spectral dimension. The size of each window after division is [C/4, C/4, C/4, C/4]. Then MSA is performed in each window. Next the window is moved half a window in the spectral direction. The size of each window at this point is [C/8, C/4, C/4, C/4, C/8]. MSA is again performed in each window. Wherefore, the process of S-W-MSA with m windows is:

\[ Y^{(m)} = [\psi(y^{(1)}) \oplus \psi(y^{(2)}) \oplus \cdots \oplus \psi(y^{(m)})] \]  

(10)

where \( \oplus \) means concat, \( y^{(i)} \) is the data of the \( i \)-th window.

Figure 4. The structure of (a) S(W)-MSA of SwinT and (b) S-(S)W-MSA of SSWT (ours).

Compared to SwinT, the other components of the S-SwinT module remain the same except for the design of the window, such as MLP, layer normalization (LN) and residual connections. Figure 1 describes two nearby S-SwinT modules in each stage, which can be represented by the following formula.

\[ \hat{Y}^l = S-W-MSA(LN(Y^{l-1})) + Y^{l-1} \]  

(11)

\[ Y^l = MLP(LN(\hat{Y}^l)) + \hat{Y}^l \]  

(12)

\[ \hat{Y}^{l+1} = S-SW-MSA(LN(Y^l)) + Y^l \]  

(13)

\[ Y^{l+1} = MLP(LN(\hat{Y}^{l+1})) + \hat{Y}^{l+1} \]  

(14)
where S-W-MSA and S-SW-MSA denote the spectral window based and spectral shifted window based MSA, \( \hat{Y}_l^i \) and \( Y^i_l \) are the outputs of S-(S)W-MSA and MLP in block \( l \).

4. Experiment

In this section, we conducted extensive experiments on three benchmark datasets to demonstrate the effectiveness of the proposed method, including Pavia University (PU), Salinas (SA) and Houston2013 (HU).

4.1. Dataset

The three datasets that utilised in the experiments are detailed here.

(1) Pavia University: The Reflective Optics System Imaging Spectrometer (ROSIS) sensor acquired the PU dataset in 2001. It comprises 115 spectral bands with wavelengths ranging from 380 to 860 nm. Following the removal of the noise bands, there are now 103 open bands for investigation. The image measures 610 pixels in height and 340 pixels in width. The collection includes 42,776 labelled samples of 9 different land cover types.

(2) Salinas: The Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor acquired the SA dataset in 1998. The 224 bands in the original image have wavelengths between 400 and 2500 nm. 204 bands are used for evaluating after the water absorption bands have been removed. The data has 512 and 217 pixels of height and width, respectively. There are 16 object classes represented in the dataset’s 54,129 marked samples.

(3) Houston2013: The Hyperspectral Image Analysis Group and the NSF-funded Airborne Laser Mapping Center (NCALM) at the University of Houston in the US provided the Houston 2013 dataset. The 2013 IEEE GRSS Data Fusion Competition used the dataset initially for scientific research. It has 144 spectral bands with wavelengths between 0.38 and 1.05 m. This dataset contains 15 classes and measures 349 × 1905 pixels with a 2.5 m spatial resolution.

We divided the label samples in different ways for each dataset. Tables 1–3 provide specifics on the number of each class for the three dataset training, validation, and testing sets. False-color map and ground-truth map of three datasets are shown in Figures 5–7.

Figure 5. Visualization of PU Datasets. (a) False-color map. (b) Ground-truth map.
Figure 6. Visualization of SA Datasets. (a) False-color map. (b) Ground-truth map.

Figure 7. Visualization of HU Datasets. (a) False-color map. (b) Ground-truth map.

Table 1. Number of training, validation and testing samples for the PU dataset.

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>Train.</th>
<th>Val.</th>
<th>Test.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Asphalt</td>
<td>83</td>
<td>83</td>
<td>6465</td>
</tr>
<tr>
<td>2</td>
<td>Meadows</td>
<td>233</td>
<td>233</td>
<td>18,183</td>
</tr>
<tr>
<td>3</td>
<td>Gravel</td>
<td>26</td>
<td>26</td>
<td>2047</td>
</tr>
<tr>
<td>4</td>
<td>Trees</td>
<td>38</td>
<td>38</td>
<td>2987</td>
</tr>
<tr>
<td>5</td>
<td>Painted metal sheets</td>
<td>17</td>
<td>17</td>
<td>1311</td>
</tr>
<tr>
<td>6</td>
<td>Bare Soil</td>
<td>63</td>
<td>63</td>
<td>4903</td>
</tr>
<tr>
<td>7</td>
<td>Bitumen</td>
<td>17</td>
<td>17</td>
<td>1297</td>
</tr>
<tr>
<td>8</td>
<td>Self-Blocking Bricks</td>
<td>46</td>
<td>46</td>
<td>3590</td>
</tr>
<tr>
<td>9</td>
<td>Shadows</td>
<td>12</td>
<td>12</td>
<td>923</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>535</td>
<td>535</td>
<td>41,706</td>
</tr>
</tbody>
</table>
Table 2. Number of training, validation and testing samples for the SA dataset.

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>Train.</th>
<th>Val.</th>
<th>Test.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Brocoli_green_weeds_1</td>
<td>25</td>
<td>25</td>
<td>1959</td>
</tr>
<tr>
<td>2</td>
<td>Brocoli_green_weeds_2</td>
<td>47</td>
<td>47</td>
<td>3633</td>
</tr>
<tr>
<td>3</td>
<td>Fallow</td>
<td>25</td>
<td>25</td>
<td>1927</td>
</tr>
<tr>
<td>4</td>
<td>Fallow_rough_plow</td>
<td>17</td>
<td>17</td>
<td>1358</td>
</tr>
<tr>
<td>5</td>
<td>Fallow_smooth</td>
<td>33</td>
<td>33</td>
<td>2611</td>
</tr>
<tr>
<td>6</td>
<td>Stubble</td>
<td>49</td>
<td>49</td>
<td>3860</td>
</tr>
<tr>
<td>7</td>
<td>Celery</td>
<td>45</td>
<td>45</td>
<td>3490</td>
</tr>
<tr>
<td>8</td>
<td>Grapes_untrained</td>
<td>141</td>
<td>141</td>
<td>10,989</td>
</tr>
<tr>
<td>9</td>
<td>Soil_vinyard_develop</td>
<td>78</td>
<td>78</td>
<td>6048</td>
</tr>
<tr>
<td>10</td>
<td>Corn_senesced_green_weeds</td>
<td>41</td>
<td>41</td>
<td>3196</td>
</tr>
<tr>
<td>11</td>
<td>Lettuce_romaine_4wk</td>
<td>13</td>
<td>13</td>
<td>1041</td>
</tr>
<tr>
<td>12</td>
<td>Lettuce_romaine_5wk</td>
<td>24</td>
<td>24</td>
<td>1879</td>
</tr>
<tr>
<td>13</td>
<td>Lettuce_romaine_6wk</td>
<td>11</td>
<td>11</td>
<td>893</td>
</tr>
<tr>
<td>14</td>
<td>Lettuce_romaine_7wk</td>
<td>13</td>
<td>13</td>
<td>1043</td>
</tr>
<tr>
<td>15</td>
<td>Vinyard_untrained</td>
<td>91</td>
<td>91</td>
<td>7086</td>
</tr>
<tr>
<td>16</td>
<td>Vinyard_vertical_trellis</td>
<td>23</td>
<td>23</td>
<td>1762</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>676</td>
<td>676</td>
<td>52,775</td>
</tr>
</tbody>
</table>

Table 3. Number of training, validation and testing samples for the HU dataset.

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>Train.</th>
<th>Val.</th>
<th>Test.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Healthy grass</td>
<td>31</td>
<td>31</td>
<td>1188</td>
</tr>
<tr>
<td>2</td>
<td>Stressed grass</td>
<td>31</td>
<td>31</td>
<td>1191</td>
</tr>
<tr>
<td>3</td>
<td>Synthetic grass</td>
<td>17</td>
<td>17</td>
<td>662</td>
</tr>
<tr>
<td>4</td>
<td>Trees</td>
<td>31</td>
<td>31</td>
<td>1182</td>
</tr>
<tr>
<td>5</td>
<td>Soil</td>
<td>31</td>
<td>31</td>
<td>1180</td>
</tr>
<tr>
<td>6</td>
<td>Water</td>
<td>8</td>
<td>8</td>
<td>309</td>
</tr>
<tr>
<td>7</td>
<td>Residential</td>
<td>32</td>
<td>32</td>
<td>1205</td>
</tr>
<tr>
<td>8</td>
<td>Commercial</td>
<td>31</td>
<td>31</td>
<td>1182</td>
</tr>
<tr>
<td>9</td>
<td>Road</td>
<td>31</td>
<td>31</td>
<td>1189</td>
</tr>
<tr>
<td>10</td>
<td>Highway</td>
<td>31</td>
<td>31</td>
<td>1166</td>
</tr>
<tr>
<td>11</td>
<td>Railway</td>
<td>31</td>
<td>31</td>
<td>1173</td>
</tr>
<tr>
<td>12</td>
<td>Parking Lot 1</td>
<td>31</td>
<td>31</td>
<td>1171</td>
</tr>
<tr>
<td>13</td>
<td>Parking Lot 2</td>
<td>12</td>
<td>12</td>
<td>446</td>
</tr>
<tr>
<td>14</td>
<td>Tennis Court</td>
<td>11</td>
<td>11</td>
<td>407</td>
</tr>
<tr>
<td>15</td>
<td>Running Track</td>
<td>17</td>
<td>17</td>
<td>627</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>376</td>
<td>376</td>
<td>14,278</td>
</tr>
</tbody>
</table>

4.2. Experimental Setting

(1) Evaluation Indicators: To quantitatively analyse the efficacy of the suggested method and other methods for comparison, four quantitative evaluation indexes are introduced: overall accuracy (OA), average accuracy (AA), kappa coefficient (κ), and the classification accuracy of each class. A better classification effect is indicated by a higher value for each indicator.

(2) Configuration: All verification experiments for the proposed technique were performed in the PyTorch environment using a desktop computer with an Intel(R) Core(TM) i7-10750H CPU, 16GB of RAM, and an NVIDIA Geforce GTX 1660Ti 6-GB GPU. The learning rate was initially set to $1 \times 10^{-3}$ and the Adam optimizer was selected as the initial optimizer. The size of each training batch was set to 64. Each dataset received 500 training epochs.
4.3. Parameter Analysis

4.3.1. Influence of Patch Size

Patch size is the spatial size of the input patches, which determines the spatial information that the model can utilize when classifying HSIs. Therefore, the model’s performance is influenced by the patch size. A too large patch size will increase the computational burden of the model. In this section, we compare a set of patch sizes \{3, 5, 7, 9, 11, 13\} to explore the effect of patch size on the model. The experimental results about patch size on the three datasets are shown in Figure 8. A similar trend was observed in all three datasets, OA first increased and then stabilized with increasing patch size. Specifically, the highest value of OA is achieved when the patch size is 9 in the PU and HU datasets, and the highest value of OA is achieved when the patch size is 11 in the SA dataset.

The size of patch is positively correlated with the spatial information contained in the patch. Increasing the patch means that the model can learn more spatial information, which will be beneficial to improve OA. And when the patch increases to a certain size, the distance between the pixels in the newly region and the center pixel is too far, and the spatial information that can be provided is of little value. So the improvement of OA is not much, and the OA will tend to be stable at this time.

Figure 8. Overall accuracy(%) with different patch sizes on the three datasets. The window numbers in transformer layers is set to [1, 2, 2, 4].

4.3.2. Influence of Window Number

In proposed S-SW-MSA, the number of windows is a parameter that can be set depending on the characteristics of the dataset. Moreover, the number of windows can be different for each transformer layer in order to extract multiple scales of features. We set up six sets of experiments, the model contains four transformer layers in the first four sets, and five transformer layers in the last two sets. The numbers in [] indicate the number of windows of S-SW-MSA in each transformer layer. The experimental results on the three datasets are shown in Table 4. According to the experimental results, the best OA for each dataset was obtained for different window number settings, and the best OA was obtained for the PU, SA and HU datasets in the 4th, 2nd and 6th group settings, respectively. We also found that increasing the number of transformer layers does not necessarily increase the performance of the model. For example, the best OA is achieved when the number of transformer layers is 4 for the PU and SA datasets and 5 for the HU dataset. Because the features of each dataset are different, the parameter settings will change accordingly.
Table 4. Overall accuracies (%) of proposed model with different number of windows in transformer layers on SA, PU and HU datasets. The patch size is set to 9.

<table>
<thead>
<tr>
<th>Windows Size</th>
<th>PU</th>
<th>SA</th>
<th>HU</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1, 1, 2, 2]</td>
<td>97.05</td>
<td>97.56</td>
<td>93.24</td>
</tr>
<tr>
<td>[1, 2, 2, 4]</td>
<td>97.86</td>
<td>97.80</td>
<td>93.35</td>
</tr>
<tr>
<td>[2, 2, 4, 4]</td>
<td>98.33</td>
<td>96.93</td>
<td>93.31</td>
</tr>
<tr>
<td>[2, 2, 4, 8]</td>
<td>98.37</td>
<td>97.70</td>
<td>93.58</td>
</tr>
<tr>
<td>[1, 1, 2, 4, 8]</td>
<td>98.20</td>
<td>96.25</td>
<td>93.38</td>
</tr>
<tr>
<td>[2, 2, 4, 4, 8]</td>
<td>98.25</td>
<td>96.31</td>
<td>93.69</td>
</tr>
</tbody>
</table>

4.4. Ablation Experiments

To sufficiently demonstrate that proposed method is effective, we conducted ablation experiments on the Pavia University dataset. With ViT as the baseline, the components of the model are added separately: S-Swin, SPE and SFE. In total, there are 5 combinations. The experimental results are shown in the Table 5. The classification overall accuracy of ViT without any improvement was 84.43%. SPE, SFE and S-Swin are proposed improvements for the ViT backbone network, which can respectively increase classification overall accuracy of 1.69%, 7.21% and 7.87% after adding into the model. The classification overall accuracy of applying the two improvements to the model together can reach 93.78%, which is higher than baseline by 9.35%. It is considered to be a great result for the improved pure transformer, but it’s a little lower than our final result. After the SFE was added to the model, the classification overall accuracy improved by 4.59%, eventually reaching 98.37.

Table 5. Ablation experiments in PU.

<table>
<thead>
<tr>
<th>Method</th>
<th>Module (%)</th>
<th>Metric (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S-Swin</td>
<td>SPE</td>
</tr>
<tr>
<td>ViT (Baseline)</td>
<td>(\times)</td>
<td>(\times)</td>
</tr>
<tr>
<td>ViT</td>
<td>(\times)</td>
<td>(\checkmark)</td>
</tr>
<tr>
<td>ViT</td>
<td>(\times)</td>
<td>(\times)</td>
</tr>
<tr>
<td>SSWT (Ours)</td>
<td>(\checkmark)</td>
<td>(\times)</td>
</tr>
<tr>
<td>SSWT (Ours)</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
</tr>
<tr>
<td>SSWT (Ours)</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
</tr>
</tbody>
</table>

4.5. Classification Results

The proposed model’s outcomes are compared with those of the advanced deep learning models: a LSTM based network (Bi-LSTM) [51], a 3-D CNN-based deep learning network (3D-CNN) [57], a deep feature fusion network (DFFN) [53], a RSSAN [52], and some transformer based model include a Vit, Swin-transformer (SwinT) [43], a SpectralFormer (SF) [44], a Hit [46] and a SSFTT [45].

Tables 6–8 show the OA, AA, \(\kappa\) and the accuracy of each category for each model’s classification on the three public datasets. Each result is the average of repeating the experiment five times. The best results are shown in bold. As the results show, proposed SSWT performs the best. On the PU dataset, SSWT is 1.02% higher than SSFTT, 3.85% higher than Hit, 9.01% higher than SwinT and 1.51% higher than RSSAN in terms of OA. Moreover, SSWT outperforms other models in terms of AA and \(\kappa\). SSWT achieved the highest classification accuracy in 7 out of 9 categories. On the SA dataset, the advantage of SSWT is more prominent. SSWT is 3.22% higher than SSFTT, 3.99% higher than Hit, 7.10% higher than SwinT, 2.64% higher than RSSAN, and 3.01% higher than DFFN in terms of OA. The same advantage was achieved for SSWT in AA and \(\kappa\). SSWT achieved the highest classification accuracy in 11 out of 16 categories. Similar results can be observed in HU dataset, where SSWT achieved significant advantages in all three metrics of OA, AA and \(\kappa\). SSWT achieved the highest classification accuracy in 6 out of 15 categories.
<table>
<thead>
<tr>
<th>Class</th>
<th>Bi-LSTM</th>
<th>3D-CNN</th>
<th>RSSAN</th>
<th>DFFN</th>
<th>Vit</th>
<th>SwinT</th>
<th>SF</th>
<th>Hit</th>
<th>SSFTT</th>
<th>SSWT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OA(%)</td>
<td>92.65 ± 0.74</td>
<td>93.08 ± 0.14</td>
<td>92.95 ± 0.38</td>
<td>93.88 ± 0.49</td>
<td>93.05 ± 0.52</td>
<td>93.72 ± 1.44</td>
<td>93.71 ± 1.12</td>
<td>98.06 ± 0.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA(%)</td>
<td>89.62 ± 0.78</td>
<td>90.17 ± 0.51</td>
<td>89.62 ± 0.54</td>
<td>90.18 ± 0.49</td>
<td>90.22 ± 0.53</td>
<td>90.75 ± 1.34</td>
<td>90.73 ± 1.11</td>
<td>95.49 ± 0.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>κ × 100</td>
<td>89.62 ± 0.78</td>
<td>90.17 ± 0.51</td>
<td>89.62 ± 0.54</td>
<td>90.18 ± 0.49</td>
<td>90.22 ± 0.53</td>
<td>90.75 ± 1.34</td>
<td>90.73 ± 1.11</td>
<td>95.49 ± 0.24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 7. Classification results of the SA dataset.

<table>
<thead>
<tr>
<th>Class</th>
<th>Bi-LSTM</th>
<th>3D-CNN</th>
<th>RSSAN</th>
<th>DFFN</th>
<th>Vit</th>
<th>SwinT</th>
<th>SF</th>
<th>Hit</th>
<th>SSFTT</th>
<th>SSWT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OA(%)</td>
<td>92.65 ± 0.74</td>
<td>93.08 ± 0.14</td>
<td>92.95 ± 0.38</td>
<td>93.88 ± 0.49</td>
<td>93.05 ± 0.52</td>
<td>93.72 ± 1.44</td>
<td>93.71 ± 1.12</td>
<td>98.06 ± 0.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA(%)</td>
<td>89.62 ± 0.78</td>
<td>90.17 ± 0.51</td>
<td>89.62 ± 0.54</td>
<td>90.18 ± 0.49</td>
<td>90.22 ± 0.53</td>
<td>90.75 ± 1.34</td>
<td>90.73 ± 1.11</td>
<td>95.49 ± 0.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>κ × 100</td>
<td>89.62 ± 0.78</td>
<td>90.17 ± 0.51</td>
<td>89.62 ± 0.54</td>
<td>90.18 ± 0.49</td>
<td>90.22 ± 0.53</td>
<td>90.75 ± 1.34</td>
<td>90.73 ± 1.11</td>
<td>95.49 ± 0.24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 8. Classification results of the HU dataset.

<table>
<thead>
<tr>
<th>Class</th>
<th>Bi-LSTM</th>
<th>3D-CNN</th>
<th>RSSAN</th>
<th>DFFN</th>
<th>Vit</th>
<th>SwinT</th>
<th>SF</th>
<th>Hit</th>
<th>SSFTT</th>
<th>SSWT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OA(%)</td>
<td>92.65 ± 0.74</td>
<td>93.08 ± 0.14</td>
<td>92.95 ± 0.38</td>
<td>93.88 ± 0.49</td>
<td>93.05 ± 0.52</td>
<td>93.72 ± 1.44</td>
<td>93.71 ± 1.12</td>
<td>98.06 ± 0.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA(%)</td>
<td>89.62 ± 0.78</td>
<td>90.17 ± 0.51</td>
<td>89.62 ± 0.54</td>
<td>90.18 ± 0.49</td>
<td>90.22 ± 0.53</td>
<td>90.75 ± 1.34</td>
<td>90.73 ± 1.11</td>
<td>95.49 ± 0.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>κ × 100</td>
<td>89.62 ± 0.78</td>
<td>90.17 ± 0.51</td>
<td>89.62 ± 0.54</td>
<td>90.18 ± 0.49</td>
<td>90.22 ± 0.53</td>
<td>90.75 ± 1.34</td>
<td>90.73 ± 1.11</td>
<td>95.49 ± 0.24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We visualized the prediction results of each model on the samples to compare the performance of the models, and the visualization results of each model on the three datasets are shown in Figures 9–11. Proposed SWSST has less noise in all three datasets compared to other models, and the classification result of SWSST is closest to the ground truth. In the
PU dataset, the blue area in the middle is misclassified by many models, and the SSWT result in the fewest errors. In the SA dataset, the pink area and the green area on the top left show a number of errors in the classification results of other models, and the SSWT classification results are the smoothest. A similar situation is observed in the HU dataset. The superiority of proposed model is further demonstrated.

Figure 9. Classification maps of different methods in PU dataset. (a) Bi-LSTM. (b) 3D-CNN. (c) RSSAN. (d) DFFN. (e) Vit. (f) SwinT. (g) SF. (h) Hit. (i) SSFTT. (j) Proposed SSWT.

Figure 10. Cont.
4.6. Robustness Evaluation

In order to evaluate the robustness of the proposed model, we conducted experiments with the proposed model and other models under different numbers of training samples. Figure 12 shows the experimental results on three datasets, we selected 0.5%, 1%, 2%, 4%, and 8% of the samples in turn as training data for the PU and SA dataset, while 2%, 4%, 6%, 8% and 10% for the HU dataset. It can be observed that the proposed SSWT is performing best in every situation, especially in the case of few training samples. The robustness
of proposed SSWT and its superiority in the case of small samples can be demonstrated. Taking the PU dataset as an example, most of the models achieve high accuracy at 8% of the training percent, with SSWT having a small advantage. And as the training percent decreases, SSWT has higher accuracy compared to other models. Similar results were found on the SA and HU datasets, where SSWT showed excellent performance for all training percents.

Figure 12. Classification results in different training percent of samples on the three datasets. (a) PU. (b) SA. (c) HU.
5. Conclusions

In this paper, we summarize the shortcomings of the existing ViT for HSI classification tasks. For the lack of ability to capture local contextual features, we use the self-attentive mechanism of shifted windows. The corresponding design is made for the characteristics of HSI, i.e., the spectral shifted window self-attention, which effectively improves the local feature extraction capability. For the insensitivity of ViT to spatial features and structure, we designed the spatial feature extraction module and spatial position encoding to compensate. The superiority of the proposed model has been verified by experimental results across three public HSI datasets.

In future work, we will improve the calculation of S-SW-MSA to reduce its time complexity. In addition, we will continue our research based on the transformer and try to achieve higher performance with a model of pure transformer structure.

Author Contributions: All the authors made significant contributions to the work. Y.P., J.R. and J.W. designed the research, analyzed the results, and accomplished the validation work. M.S. provided advice for the revision of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

17. Navarro, A.; Nicastro, N.; Costa, C.; Pentangelo, A.; Cardarelli, M.; Ortenzi, L.; Pallottino, F.; Cardi, T.; Pane, C. Sorting biotic and abiotic stresses on wild rocket by leaf-image hyperspectral data mining with an artificial intelligence model. *Plant Methods* 2022, 18, 45. [CrossRef] [CrossRef] [PubMed]


33. Zhou, F.; Hang, R.; Liu, Q.; Yuan, X. Hyperspectral image classification using spectral-spatial LSTMs. *Neurocomputing* 2019, 328, 39–47. [CrossRef] [CrossRef]


40. Li, W.; Chen, H.; Liu, Q.; Liu, H.; Wang, Y.; Gui, G. Attention mechanism and depthwise separable convolution aided 3DCNN for hyperspectral remote sensing image classification. *Remote Sens.* 2022, 14, 2215. [CrossRef] [CrossRef]


49. Hu, X.; Yang, W.; Wen, H.; Liu, Y.; Peng, Y. A lightweight 1-D convolution augmented transformer with metric learning for hyperspectral image classification. Sensors 2021, 21, 1751. [CrossRef] [CrossRef] [PubMed]


56. Gong, H.; Li, Q.; Li, C.; Dai, H.; He, Z.; Wang, W.; Li, H.; Han, F.; Tuniyazi, A.; Mu, T. Multiscale information fusion for hyperspectral image classification based on hybrid 2D-3D CNN. Remote Sens. 2021, 13, 2268. [CrossRef] [CrossRef]


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.