OE Optical Engineering

RESEARCH PAPER

Spatially augmented guided sequence-based bidirectional encoder representation from transformer networks for hyperspectral classification studies

Yuanyuan Zhang,[a] Wenxing Bao,[a,*] Hongbo Liang,[b] and Yanbo Sun[a]
[a]North Minzu University, School of Computer Science and Engineering, Yinchuan, China
[b]Hefei University of Technology, School of Computer Science and Information Engineering, Hefei, China

ABSTRACT. In recent years, bidirectional encoder representation from transformers (BERT) models have achieved superior performance in hyperspectral images (HSIs). It can capture the long-range correlations between HSI elements, but the local space and spectral band information of HSI is insufficient. We propose a spatially augmented guided sequence BERT network for HSI classification study, referred to as SAS-BERT, which makes more effective use of HSI's spatial and spectral information by improving the BERT model. First, a spatial augmentation learning module is added in the preprocessing stage to obtain more significant spatial features before the input network and better guide the spatial sequence. Then a spectral correlation module was used to represent the spectral band features of the HSI and to establish a correlation with the spatial location of the images to obtain better classification performance. Experimental results on three datasets show that the method proposed achieves better classification performance than other state-of-the-art methods.

Keywords: bidirectional encoder representation from transformers; hyperspectral image classifification; spatially augmented; sequence

Paper 20230515G received Jun. 2, 2023; revised Oct. 3, 2023; accepted Oct. 11, 2023; published Oct. 31, 2023.

1 Introduction

With hundreds of narrow continuous spectral bands, hyperspectral images (HSIs)[1] better represent the semantic information of remotely sensed features.[2,3] Their rich spectral features provide a powerful tool for achieving accurate pixel-level classification.[4,5] HSI classification is widely used in precision agriculture,[6–8] mineral surveying,[9] anomaly detection,[10] and land cover mapping.[11] Compared with traditional visual images, HSIs have two unique features.[12] The first is that the heterogeneity of matter leads to the high-dimensional and nonlinear spectral characteristics of hyperspectral data. Second, there is a high correlation between adjacent hyperspectral spectra, and spatial correlation can supplement spectral information to provide a more accurate interpretation of surface cartography.

Classification of hyperspectral remote sensing images is fundamental to HSI processing and applications. Its ultimate aim is to assign a unique identity to each image element. Analyzing and processing the spectral and spatial characteristics of the various types of land features in the HSIs classify each image element into a category corresponding to the actual land features, thus enabling the classification of land features. Conventional methods generally use band selection and feature extraction for dimensionality reduction, compressing the original spectral image elements

*Address all correspondence to Wenxing Bao, baowenxing@nun.edu.cn into a low-dimensional space, such as principal component analysis,[13] support vector machines (SVM),[14] and random forests.[15] The properties of HSIs constrain these methods, and their classification results could be better. HSIs possess both spectral properties and spatial dependence, which implies a joint representation of spectral and spatial features. Therefore some researchers have proposed using a spatial–spectral feature extractor to extract HSI features. For example, spatial information was combined with multinomial logistic regression for HSI feature extraction.[16] Li et al.[17] proposed an edge-preserving filtering-based framework for spectral–spatial classification, significantly improving the classification accuracy of SVM with fewer computer resources. However, the above models all consist of shallow structures, such network models contain fewer hidden layers but require a more significant number of neurons, so shallow network structures do not provide a better description of HSIs.

With the development of artificial intelligence, deep-learning-based methods are widely used in remote sensing classification. For example, the spatial update super voxel stacking autoencoder method is an improved depth network. This method uses the spatial context of similar spectra within consecutive pixels for effective HSI classification.[18] The convolution neural network (CNN) has a unique advantage in processing high-dimensional HSI data among these deep-learning models. For example, Chen et al.[19] designed a 3D convolutional neural network capable of efficiently extracting spectral and spatial features with good classification performance. Li et al.[20] proposed a deep-learning framework for fully convolutional neural networks, applying deconvolution to HSI for the first time. Zhong et al.[21] proposed a deep residual network for HSI classification, which used ResNets and CNNs of different depths and widths to investigate the effect of deep-learning model size on HSI classification accuracy. Kanthi et al.[22] proposed a new 3D depth feature extraction CNN model for HSI classification using spectral and spatial information, which divides the HSI data into 3D patches and feeds them into the proposed model for depth feature extraction.

However, CNN's reliance on a geometrically fixed structure of convolutional kernels hinders long-range dependencies between features from nonlocal locations. Transformers can effectively alleviate the problem of small perceptual range and low description efficiency caused by geometrically structured convolutional filter banks like CNN. Originally proposed as a sequence-to-sequence (seq2seq) model for machine translation, the transformer has become a mainstream model in natural language processing (NLP). This simple and efficient architecture also performs well in HSI classification. He et al.[23] applied for the first time a spatial transformation network to obtain the optimal input to a CNN for HSI classification. Zhao et al.[24] proposed a convolutional transform network for the fusion of spectral information and spatial location of HSIs using central position coding. Hong et al.[25] proposed the spectral–spatial transformer to capture the relationship between spectral bands along the spectral dimension. Meanwhile, spectral-spatial transformer network (SSTN) designed a spectral–spatial transformer network consisting of a spectral association module and a spatial attention module, using a new factorized architecture search framework to determine the hierarchical operations and block order of SSTN.[26]

The bidirectional encoder representations from transformers (BERT) model[27] is based on the transformer taking its encoder part to obtain a bidirectional encoder representation model, a self-coding language model. BERT is structurally more superficial than the transformer in that it only uses the encoder part of the transformer. The BERT model was proposed in October 2018 by the Google AI Institute, and the model was initially popular in NLP. Because it is built on the transformer, BERT has powerful linguistic representation and feature extraction capabilities. However, BERT models consume substantial hardware resources so that reproducibility could be better and model convergence could be faster. RoBERTa, proposed by Liu et al.,[28] the model does not change the network structure of BERT but only modifies some pretraining methods, including dynamic masking and discarding the next sentence predict task. Chen et al. proposed ALBERT, which argued that the model parameters of BERT were too large and too resource-intensive. They proposed using word vector factorisation, cross-layer parameter sharing or sentence order prediction to reduce the model size and thus improve the training speed.[29] Cui et al.[30] proposed MacBERT, the model that proposes not using a mask, replacing the mask-tagged position with another similar word, and then allowing the model to correct errors to achieve better results automatically. Related improved models are MASS,[31] UNILM,[32] and SpanBert,[33] all of which have achieved good results.

He et al. argued that neurolinguistic models are more similar to HSIs in some aspects. They assembled multiple self-attention layers into transformer units. They converted the image elements of HSIs into sequences as input data to be applied to HSI classification tasks with satisfactory results. Their proposed HSI-BERT converts spectral image sets into sequences for characterization. This approach not only does not disrupt the inherent continuous spectral distribution of HSIs but also supports more flexible and dynamic input regions and is more easily generalized.[34] The BERT model captures richer nonlocal image element information. However, the complex distribution of spectral space in HSIs results in low efficiency of BERT model feature extraction and inconspicuous feature description in high-dimensional spectral space. Each pixel of HSI is marked by BERT sequence mode, and a multihead self-attention mechanism extracts nonlocal spatial information of HSI. Still, the representation of local spatial features of HSIs needs to be improved. Moreover, the multihead self-attention mechanism only considers the feature characterization of the spatial pixel rotation sequence. It needs to utilize the rich spectral features unique to HSIs more effectively. In order to solve the above problems based on the BERT model, this paper proposes a spatially augmented guided sequential BERT network for HSI classification studies called SAS-BERT.

The main contributions of this paper are as follows.

- This paper uses a new framework for HSI classification based on unsupervised BERT networks. The framework adds a spatial augmentation module to the BERT model to capture the local spatial information of HSI and to guide the sequence patterns. The nonlocal and local spatial information of HSI is utilized more effectively.
- In order to enhance the ability of the BERT model to describe remotely sensed features, this paper combines the rich spectral information of HSIs to extract spectral features and establish correlations-spatial locations to better account for the intraclass consistency and interclass variability between spectral bands.
- Compared with the BERT model, the network proposed in this paper aggregates the spatial augmentation module and the spectral association module to integrate and invert the spatial information of HSIs. The experimental results illustrate that the model obtains a more satisfactory classification accuracy than the state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 presents a relevant introduction to the BERT model. Section 3 presents the scheme of the proposed SAS-BERT network and its components. Section 4 presents the experimental results and analytical findings. In Sec. 5, conclusions are drawn.

2 Related Introduction

2.1 BERT Network Structure

The BERT model has been fine-tuned with good results on some downstream NLP tasks. The general framework of the BERT model is shown in Fig. 1, which is mainly based on the

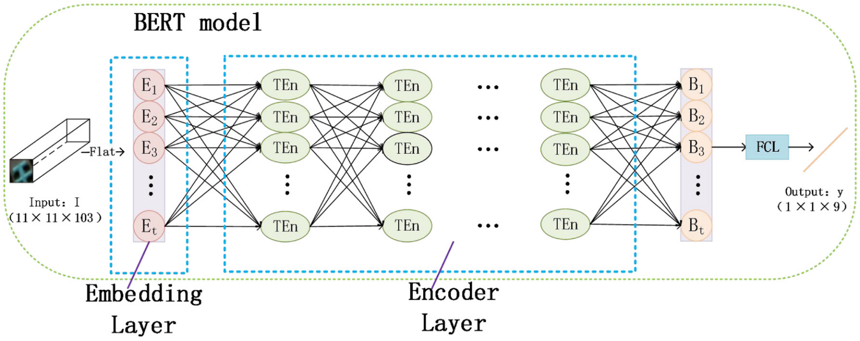


Fig. 1 Diagram of the general framework of the BERT model. The BERT model consists of an embedding layer, an encoder layer, and an FCL. The encoder layer is obtained by stacking multiple TEn.

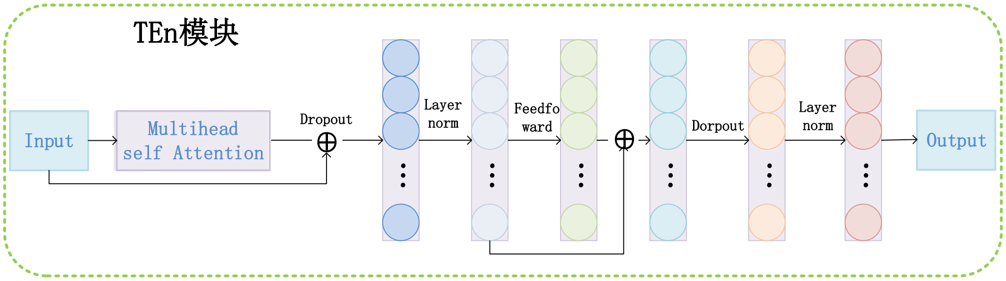


Fig. 2 Structural diagram of TEn.

transformer's encoder module, which will be referred to as TEn, as shown in Fig. 2, and the BERT model is obtained by stacking the TEn.

The BERT model consists of two main modules: embedding and encoder layers. The embedding layer includes token embedding, position embedding, and segment embedding, and segment embedding is not required for HSI classification. The token embedding implements a vector representation of the image element itself, and the position embedding learns the position properties of the image element.

The embedding layer is followed by the encoder layer, which contains the multiheaded self-attention (MHSA) module (Fig. 3). This mechanism improves the model's ability to focus on different locations and multiple focus regions simultaneously. The self-attentive mechanism requires the generation of three-word vectors **Q**, **K**, and **V** to calculate the attention of each image element, which is obtained by multiplying the image element with three training learned weight matrices. However, in multiheaded attention, multiple learned weight matrices of **Q**, **K**, and **V** are randomly initialized and independently map the input vectors to different subspaces, thus enriching the feature representation of the information. Different heads of the MHSA mechanism obtain different attentions. For the input vector, if multiple attention is used and the number of heads used is p, the input vector is divided into p independent vectors, each using self-attention to calculate the attention weight. When completed, it will be merged. Therefore, it is a parallel mechanism within a submodule. All heads work independently and in parallel. Internal to the multihead self-attention mechanism is a scaled dot product attention (Fig. 4), calculated as follows:

$$\text{attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\mathbf{V}, \tag{1}$$

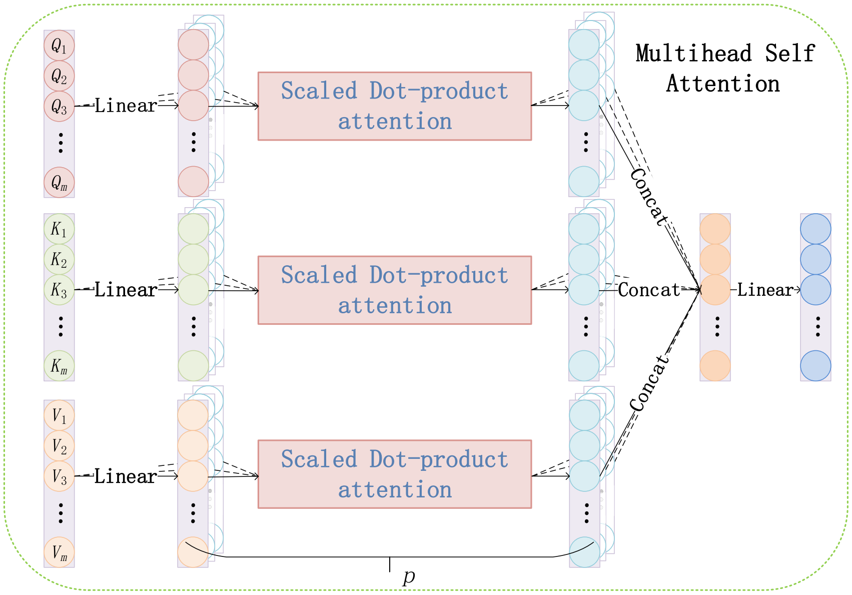


Fig. 3 Schematic diagram of the MHSA mechanism.

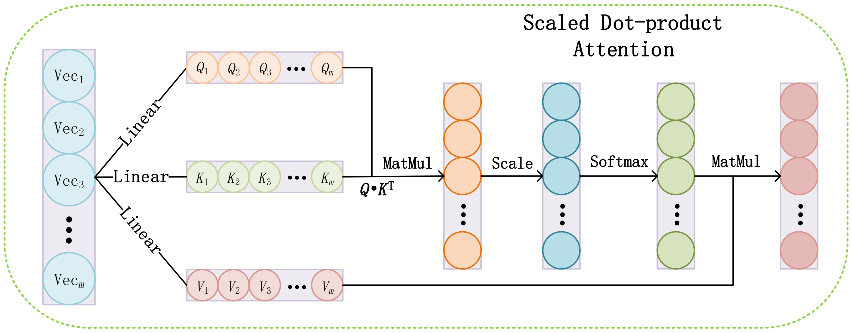

Scaled Dot-product Attention
Vec_1
Vec_2
Vec_3
Vec_m
Linear
Linear
Linear
Q_1 Q_2 Q_3 ··· Q_m
K_1 K_2 K_3 ··· K_m
V_1 V_2 V_3 ··· V_m
MatMul
$Q \cdot K^T$
Scale
Softmax
MatMul

Fig. 4 Schematic diagram of the scaled dot product attention mechanism.

where **Q**, **K**, and **V** denote query, key, and value, respectively, and d_k denotes the dimension of the vector key. After the input image is converted into a sequence, all the pixel vectors on the sequence are multiplied by three randomly initialized matrices, and three vectors, **Q**, **K**, and **V**, are obtained. As shown in Fig. 4, the attention mechanism takes **Q** as the target pixel that needs to predict the label, and the number of pixel vectors generates the number of **Q** vectors, so each pixel in the sequence can be the target pixel. **K**, as the **Q** context pixel, matches the similarity with **Q**. The similarity between the query and each key is used as the weight, and then the value of the target pixel is weighted and fused with the value of the individual pixels of the context as the output of attention.

The MHSA contains p heads $(H_1, H_2, \ldots, H_p)$:

$$\text{MHSA}(X) = \text{concat}(H_1, H_2, \ldots, H_p)W^O, \tag{2}$$

where W^O is the matrix of learned parameters, and h_i is the result of the attention structure of i. The h_i of self-attention structure can be described as follows:

$$h_i = \text{attention}(XW_i^Q, XW_i^K, XW_i^V), \tag{3}$$

where X is the target image element, and the learning parameter matrices $W^Q, W^K, W^V \in R^{d\times d/\text{p}}$ are used for the affine projection.

3 Modified BERT Network Structure

In this paper, the BERT network is modified by adding a spatial augmentation module and a spectral correlation module. The modified network is named the spatially augmented guided sequence-based BERT network (SAS-BERT). The spatial augmentation module guides the sequence patterns and obtains local spatial information of the HSI. The spectral correlation module establishes correlations between spectral kernels and spatial locations, enhancing the description of the spectral information of the image. A schematic diagram of the SAS-BERT framework is shown in Fig. 5. Random in Fig. 5 refers to random seeds, it means that the input samples

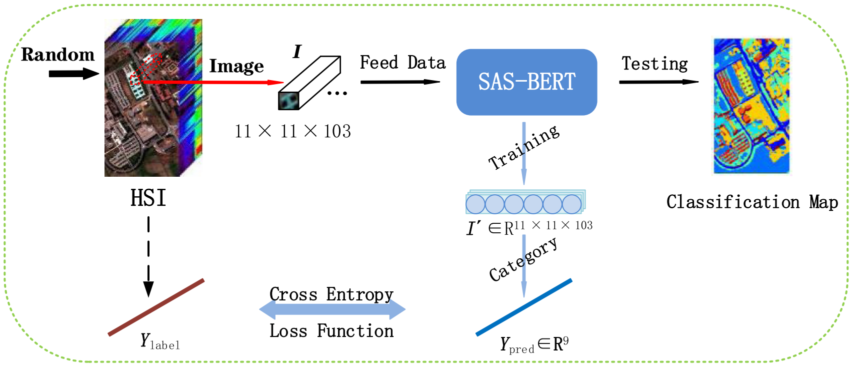


Fig. 5 Schematic diagram of the SAS-BERT framework.

for the 10 repeated experiments were randomly selected. Assume the original HSI dataset $I \in R^{H\times W\times b}$, where $H \times W$ is the spatial size and b is the spectral dimensions. When fed into the network, it forms a one-hot category vector $Y = \{y_1, y_2, \cdots, y_C\} \in R^{1\times 1\times C}$, where C is the number of land cover categories. Taking the Pavia dataset as an example, the spectral dimensions are 103, and the land cover category is 9, which is introduced explicitly in the experimental dataset part of the fourth part. The spectral size is set to 11×11. SAS-BERT consists of the spatially augmented learning module and the spe-BERT module, which contains the spectral correlation module and the multiheaded self-attentive module.

3.1 Spatially Augmented Learning Module

The MHSA mechanism of the BERT model captures rich nonlocal spatial contextual information but ignores the representation of local spatial information on HSIs. In this paper, we guide HSI's local spatial feature extraction through the spatial augmented learning module to obtain an optimal input and then send it to the BERT module for further feature extraction and classification tasks.[35] The diagram of the spatially augmented learning module is shown in Fig. 6, which contains three parts that are described below.

The first section is the localization network (F_{loc}). The localization network in Fig. 6 completes the extraction of local spatial information. The affine transformation of Eq. (4) through several hidden layers (convolution, pooling, full connection, etc.) map the coordinate point relationship between the input and output feature maps. The parameters in affine transformation are used to rotate, translate, and scale the original input image, and then the optimal input image for BERT model. The ultimate goal of using localization network is to obtain a good classification performance (as shown in red in Fig. 6). It learns the affine transformation matrix A_θ. After several convolutions or entire connection operations, the network will input image I, and then a regression layer output affine transformation matrix $A_\theta \in R^{2\times 3}$, A_θ is a learnable parameter. Four transformations are carried out by changing the parameters of A_θ: translation, scaling, rotation, and clipping. These parameters map the coordinate point relationship between the input and output feature maps. The coordinate points of the input feature map are obtained according to A_θ:

$$F_{\text{loc}}(I) = A_\theta = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}, \tag{4}$$

where I is the input image; A_θ is an affine transformation matrix; $F_{\text{loc}}(\cdot)$ denotes a function formed by a fully connected and other layers; a, b, c, d, e, f denotes different transformations by changing the values of these six parameters. For example, c, e for translation, a, e for scaling, a, b, d, e for rotation, and b, d for cropping.

The second section is the grid generator, as shown in the yellow section of Fig. 6. The position mapping relationship is based on the affine transformation matrix A_θ. The affine transformation matrix A_θ and the lattice points of the output feature map are used to find the coordinates of each pixel point of the output feature map corresponding to the input image:

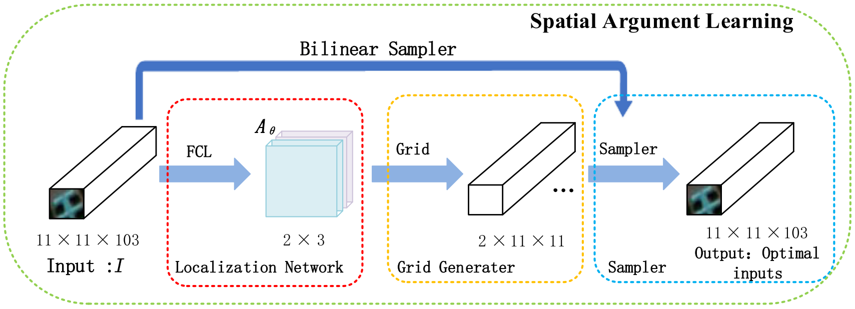


Fig. 6 Diagram of the spatially augmented learning module.

$$S_\theta(G_{(i,j)}) = \begin{pmatrix} x_i^I \\ y_j^I \end{pmatrix} = A_\theta \begin{pmatrix} x_i^O \\ y_j^O \\ 1 \end{pmatrix}, \tag{5}$$

where $S_\theta(\cdot)$ denotes the function formed by parametric grid sampling; $G_{(i,j)} = (x_i^O, y_j^O)$ denotes the individual grid coordinates of the output feature map; and (x_i^I, y_j^I) denotes the corresponding grid coordinate points of the output feature map on the input image.

The third section is the sampler, as shown in the blue section of Fig. 6. That is, the pixel value of the output image is calculated using (x_i^I, y_j^I) corresponding to the pixel value of the input image as the pixel value of the output image at (i, j). The interpolation algorithm calculates the pixel values of the output image based on the location mapping. Because the grid coordinate point corresponding to the input image in the second part may be a fractional number, the interpolation algorithm is used to correct the pixel values at that location. Bilinear interpolation is used here for interpolation:

$$V_{i,j}^l = \sum_0^H \sum_0^W I_{n,m}^l \cdot \max(0,1 - |x_i^I - m|) \max(0,1 - |y_j^I - n|), \tag{6}$$

where $V_{i,j}^l$ is the pixel value of the output feature map at channel l and coordinate (x_i^O, y_j^O), $I_{n,m}^l$ is the pixel value of the input feature map at channel l, coordinate (n, m) ((n, m) traverses all coordinate points of the input feature map). The smaller the values of $|x_i^I - m|$ and $|y_j^I - n|$ are, the closer the distance between (x_i, y_j) and (n, m) is. The larger the values of $1 - |x_i^I - m|$ and $1 - |y_j^I - n|$ are, the larger the value of max is, and the greater the weight will be. $(x_i^I, y_j^I) \rightarrow (n, m)$ is in a lattice point, and finally, four corresponding weights are obtained, summed, and output $V_{i,j}^l$.

The spatial augmentation module can be trained end-to-end with the entire convolutional network. It is inserted directly into the convolutional network and runs with good classification performance.

3.2 spe-BERT Module

The spe-BERT module contains an MHSA module and a spectral correlation module. One of the essential concepts in the original BERT model is the MHSA mechanism, as shown in Fig. 4, which can link information at different positions on the input image element to obtain nonlocal long-range dependencies. However, the MHSA mechanism only considers the spatial key to the image element's feature representation, ignoring the spectral band's characteristics for HSI interpretation. Spectral features are the key features to distinguish ground objects in HSI. So this paper adds a spectral correlation module to the BERT module to aggregate HSI's spectral and spatial features for better classification performance.

The schematic diagram of the spectral combined module is shown in Fig. 7. It comprises a multibranch network structure that uses two-dimensional convolution to integrate and invert the spatial information of the image element sequences along the spectral dimension. The sequenced

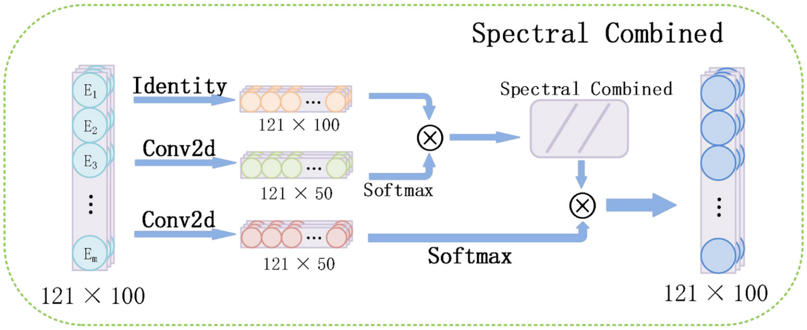


Fig. 7 Schematic diagram of the spectral combined module.

feature cube X feed into the spectral correlation module and its spectral correlation kernel is calculated as follows:

$$M_1 = \sigma(\zeta(X; W_{M_1})) \in R^{mn \times k}, \tag{7}$$

$$\text{Asso} = M_1^{\mathrm{T}} \cdot X^{\mathrm{T}} = (X \cdot M_1)^{\mathrm{T}} \in R^{k \times l}, \tag{8}$$

where $\zeta(\cdot)$ denotes a convolution operation and generates a tensor $mn \times k$; $\sigma(\cdot)$ is the softmax function, which assigns independent weights to each image element. M_1 is the generated mask, which integrates the spatial information in conjunction with the input features X to obtain a feature map of the spectral kernel concerning the spatial location. The output of the spectral correlation kernel is as follows:

$$M_2 = \sigma(\zeta(X; W_{M_2})) \in R^{mn \times k}, \tag{9}$$

$$\text{Spe}_{\text{Asso}} = \text{Asso}^{\mathrm{T}} \cdot M_2^{\mathrm{T}} = (M_2 \cdot \text{Asso})^{\mathrm{T}} \in R^{m \times n \times l}, \tag{10}$$

where M_2 is another generated mask and $\zeta(\cdot; W_{M_1})$ shares the training parameters with $\zeta(\cdot; W_{M_2})$.

The spectral correlation module establishes the correlation between spectral kernels and spatial locations, complementing the BERT model for HSIs with missing spectral information.

3.3 SAS-BERT

In this paper, we have chosen the Pavia University dataset of dimensions 610 × 340 × 103 as an example to illustrate the designed SAS-BERT model. The Pavia University dataset is a selection of hyperspectral data from images taken in 2003 of the Italian city of Pavia, containing nine feature classes. Figure 8 details the SAS-BERT algorithm flow. The entire network consists of a spatial augmentation module, a spe-BERT module, and a fully connected layer (FCL). The spe-BERT module, in turn, contains the MHSA mechanism module (a module of the BERT model itself) and the spectral correlation module.

SAS-BERT uses a cross-entropy loss function to minimise losses:

$$L_{\text{ce}} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{C} y_{i,j}\,\log(y'_{i,j}), \tag{11}$$

where y and y' denote the actual and predicted one-hot label vectors, respectively. B is the number of samples in a batch, and C is the number of categories. $y_{i,j}$ denotes the scalar of the j'th category for the i'th sample.

Algorithm 1 describes the detailed training process of SAS-BERT in detail.

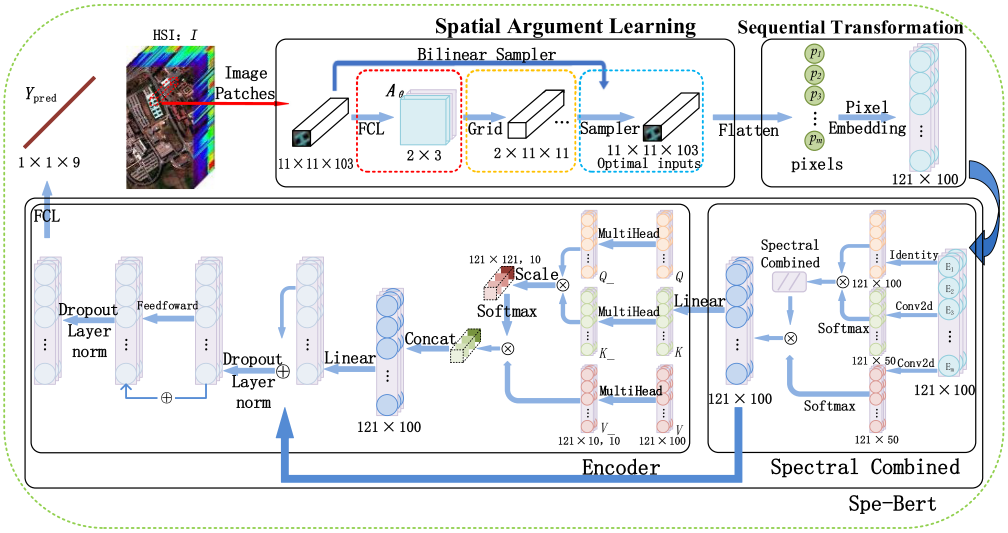


Fig. 8 Detailed framework diagram of SAS-BERT.

Algorithm 1 SAS-BERT algorithm.

```
Input: Hyperspectral image cubes $I$, model depth $d$, attention to the number of heads $h$, training batch $b$, training epoch $e$.
Output: Classification evaluation results for $X_{\mathrm{test}}$ and predicted classification charts.
Begin
Input hyperspectral image cubes $I$;
for $i = 0$ to epoch $e$ do
  for $j = 0$ to batch $b$ do
    Generate the affine transformation matrix $A_\theta$ using the localization network;
    The grid generator calculates the coordinates of each pixel point of the output feature map corresponding to the input image according to $A_\theta$;
    As shown in Eq. (6), calculate the pixel values of the output image using bilinear interpolation to obtain a spatially augmented input image;
    Spatially augmented images for sequence conversion;
    for $k = 1$ to depth $d$ do
      The sequenced image is fed into the spectral correlation module to obtain a spectral kernel relation to the spatial location of the feature map $Spe_{\mathrm{Asso}}$;
      Divide the last dimension of the feature map into $h$ attention heads.
      The attentional characteristics of each attentional head are obtained by Eq. (1);
      Concat up the attentional features of $h$ attentional heads;
      Combined with the spectral correlation of the relational feature map $Spe_{\mathrm{Asso}}$;
      $k++$;
    end for
    j++;
  end for
  $i++$;
end for
Load the model and feed $X_{\mathrm{test}}$ into the model prediction;
Obtaining classification evaluation results and predicted classification maps.
```

4 Experimental Results

All experiments in this paper are implemented on an Ubuntu 18.04 system using a GeForce RTX 2080 GPU and TensorFlow with CUDA 9.0 and Python 3.6. All subsequent training and testing experiments were conducted based on this environment.

4.1 Experimental Datasets

In this paper, three classical real datasets, Indian Pines (IN), Pavia University (PU), and Houston (HOU), are used for the experiments.

- *Indian Pines*. IN was acquired by the airborne visible, infrared imaging spectrometer to image a patch of Indian pine trees in Indiana, United States, in 1992. They then captured a 145×145 size image for annotation as an HSI classification study. The spatial resolution was 20 m and contained 16 vegetation classes. Bands 104 to 108, 150 to 163, and 220, which cannot reflect by water, were removed, and the remaining 200 bands retain.

- *Pavia University*. The PU image by an airborne reflectance optical spectroscopic imager of Pavia, Italy, was then selected for HSI classification studies after annotating images with dimensions 610 × 340 in size. The spatial resolution is 1.3 m and contains nine classes of land features. Twelve bands were affected by noise removal, leaving 103 bands.
- *Houston*. The HOU data acquired by the ITRES CASI-1500 sensor. Initially used in the IEEE GRSS Data Fusion Competition 2013, it was provided by the Hyperspectral Image Analysis Group and the NSF-funded Center for Airborne Laser Mapping (NCALM) at the University of Houston, USA, with a size of 349 × 1905 and containing 15 feature classes. Contains 144 spectral bands in the range of 364 to 1046 nm.

4.2 Evaluation Indicators

In this paper, three classification evaluation metrics, overall accuracy (OA), average accuracy (AA), and kappa coefficient (Kappa), are used as evaluation metrics to validate the experimental performance of SAS-BERT. OA indicates the percentage of correctly classified pixels to the total number of pixels; AA indicates the average percentage of the sum of the ratio of the number of correctly classified pixels in each category to the overall number of pixels in that category. The Kappa coefficient indicates the percentage of good or bad classification of the image as a whole. Higher OA, AA, and Kappa values indicate better classification results, whereas AA measures the category's excellent or lousy classification results.

4.3 Parameter Adjustment

In order to improve the efficiency of the SAS-BERT network, this paper sets the neighbourhood size to 11 × 11, the learning rate to 3×10^{-4}, the batch size to 16, the training times to 200, the dropout rate to 0.3, and samples 5%, 1% and 2% for IN, PU, and HOU, respectively, for training. This paper experiments with several factors affecting SAS-BERT's representation of HSIs.

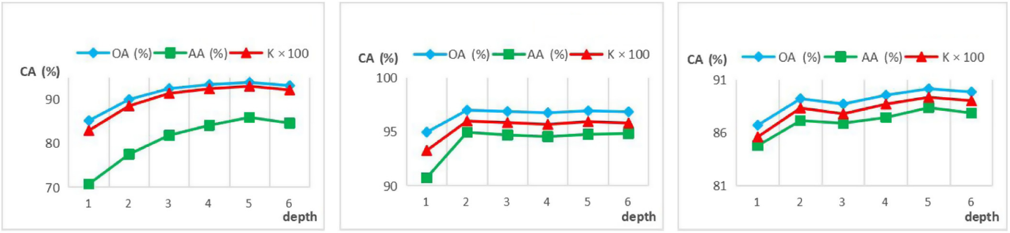
4.3.1 *TEn layers*

Evaluating the number of TEn layers at different depths in BERT: the effects of TEn layers on HSIs classification were evaluated. The effect of TEn layer depth on classification accuracy verifies for the BERT network for the three datasets above. Letting the number of attention heads $h = 10$, the classification results for different TEn layer depths on the IN, PU, and HOU datasets show in Fig. 9.

As shown in Fig. 9, the highest evaluation results for HSI classification accuracy were obtained on the IN and HOU datasets when the TEn layer depth was 5. When the TEn layer depth is 2, the highest evaluation result is received on the PU dataset. Therefore, the TEn layer depths selected in this paper are 5, 2, and 5 for the IN, PU, and HOU datasets, respectively.

4.3.2 *Number of self-attended heads*

Evaluation of the number of heads for the MHSA mechanism: different numbers in the MHSA mechanism in the BERT model significantly impact classification performance, and this paper



(a) (b) (c)

Fig. 9 Line graphs of classification accuracy for different TEn layer depths on the (a) IN, (b) PU, and (c) HOU datasets.

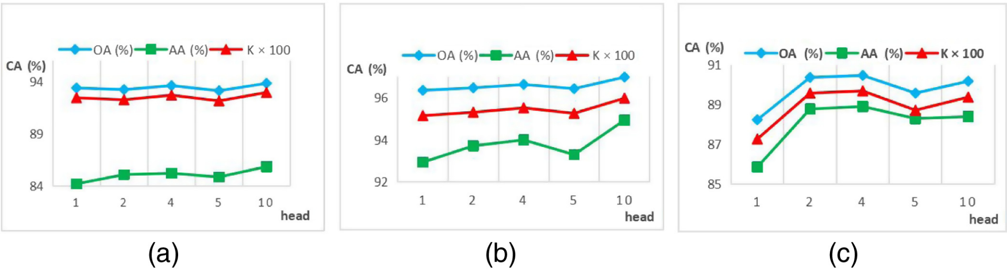


Fig. 10 Line graphs of classification accuracy for different numbers of attention heads on the (a) IN, (b) PU, and (c) HOU datasets.

evaluates the effect of other numbers of attention heads on the classification results. Let the TEn layer depths d of the IN, PU, and HOU datasets be 5, 2, and 5, respectively. The classification accuracies of different numbers of attention heads on the IN, PU, and HOU datasets show in Fig. 10.

As shown in Fig. 10, different attention heads are set for the proposed model. The highest evaluation results were obtained when the number of attention heads was 10 on the IN and PU datasets. When the number of attention heads is 4, the highest evaluation result is received on the HOU dataset. Therefore, the number of attention heads selected in this paper for the IN, PU, and HOU datasets are 10, 10, and 4, respectively.

4.4 Experiments Related to the Spatial Augmentation Learning Module and the Spectral Correlation Module

4.4.1 *Affine transformation*

Evaluation of various affine transformations of the spatial augmentation learning module: since the affine transformation of the spatial augmentation module includes translation, rotation, and scaling, the results of each shift may significantly impact the experiments. In this paper, several experiments of mapping, translation, scaling, and process of the image elements were evaluated, such that the TEn layer depth d was 5, 2, and 5 for the IN, PU, and HOU datasets, respectively; the number of attention heads selected for the IN, PU, and HOU datasets were 10, 10, and 4, respectively. The classification accuracies of the different transformations on the IN, PU, and HOU datasets are shown in Table 1. Let the classification accuracy be as CA and the method in this paper as SAL.

The affine transformations from left to right in Table 1 are direct mapping (dm), translation (tra), doubling (dou), halving (hal), rotation by 30 deg, rotation by 45 deg, and rotation by 60 deg, with rotation being a clockwise rotation by an angle around the origin. The IN dataset gives higher results for the transformations doubling the image element and rotating it by 60 deg, with the AA for the 60 deg rotation being almost 2% less accurate than the experimental results for doubling the image element and the overall higher classification accuracy for doubling the image element when considered together. The Pavia dataset sees the results of the doubled transformation outperforming the other transformations. The HOU dataset shows better classification results when the image elements reduce by half than the other transformations.

The classification accuracy is better than the original BERT module for most affine transformations when only the spatial augmentation module adds. The OA and Kappa for the IN dataset are smaller than the original BERT module for translation and halving. Still, the classification results for AA are higher than the direct use of the BERT module. The results for the PU dataset were all better than the original BERT module classification accuracy, or the classification accuracy was not significantly different. The classification results of the HOU dataset and PU dataset are basically the same, and the classification accuracy of the HOU dataset is slightly worse than that of the original BERT module in translation transformation.

The effect of using the spatial augmentation and spectral correlation modules together needs to be clarified, combined with affine transformations separately with the spectral correlation module in this paper, as explained later.

Table 1 Classification accuracy of different affine transformations on the IN, PU, and HOU datasets.

Datasets	CA	SAL (dm)	SAL (tra)	SAL (dou)	SAL (hal)	SAL (30 deg)	SAL (45 deg)	SAL (60 deg)
IN	OA (%)	95.08 ± 1.10	93.63 ± 0.97	**94.83 ± 0.87**	92.82 ± 1.01	94.93 ± 0.82	95.01 ± 1.00	95.13 ± 0.70
	AA (%)	87.73 ± 1.11	89.92 ± 2.46	**89.30 ± 3.01**	87.27 ± 1.22	87.18 ± 2.35	88.04 ± 2.23	87.35 ± 1.91
	$k \times 100$	94.38 ± 1.26	92.73 ± 1.10	**94.11 ± 0.99**	91.80 ± 1.16	94.22 ± 0.93	94.31 ± 1.14	94.45 ± 0.79
PU	OA (%)	97.12 ± 0.41	96.85 ± 0.56	**97.19 ± 0.50**	96.21 ± 0.60	96.91 ± 0.88	96.86 ± 0.73	97.18 ± 0.42
	AA (%)	94.54 ± 1.30	94.60 ± 1.16	**95.36 ± 0.88**	93.51 ± 1.06	94.67 ± 1.12	94.22 ± 1.50	95.07 ± 0.68
	$k \times 100$	96.17 ± 0.55	95.81 ± 0.74	**96.27 ± 0.66**	94.96 ± 0.79	95.89 ± 1.17	95.83 ± 0.98	96.25 ± 0.56
HOU	OA (%)	90.91 ± 0.93	88.23 ± 1.32	91.09 ± 0.91	**91.12 ± 1.57**	90.34 ± 0.98	90.95 ± 0.57	90.41 ± 1.04
	AA (%)	89.75 ± 1.27	88.15 ± 2.02	90.78 ± 1.07	**91.20 ± 1.62**	89.54 ± 1.12	90.77 ± 0.69	89.71 ± 1.69
	$k \times 100$	90.16 ± 1.00	87.27 ± 1.43	90.37 ± 0.99	**90.40 ± 1.70**	89.55 ± 1.06	90.21 ± 0.62	89.63 ± 1.13

Note: SAL(dou) has the best experimental results in Indian and Pavia datasets, which are marked in bold. SAL(hal) has the best experimental results in the Houston dataset and are highlighted in bold.

Table 2 Classification accuracy of the spectral correlation module on the IN, PU, and HOU datasets.

Datasets	CA	HSI-BERT	spe-BERT
IN	OA (%)	93.36 ± 0.88	**94.95 ± 0.89**
	AA (%)	85.01 ± 1.90	**90.47 ± 2.25**
	$k \times 100$	92.42 ± 1.02	**94.24 ± 1.02**
PU	OA (%)	96.38 ± 0.58	**97.41 ± 0.54**
	AA (%)	93.41 ± 0.90	**95.24 ± 1.71**
	$k \times 100$	95.19 ± 0.77	**96.56 ± 0.72**
HOU	OA (%)	89.59 ± 2.10	**91.12 ± 0.69**
	AA (%)	87.55 ± 3.17	**91.06 ± 0.83**
	$k \times 100$	88.74 ± 2.28	**90.40 ± 0.75**

Note: Bold values represent the optimal results under the same evaluation index.

4.4.2 *Spectral correlation modules*

Experiments on spectral correlation modules: the results of the experiments with the addition of a separate spectral correlation module are shown in Table 2.

As shown in Table 2, including the spectral correlation module resulted in a corresponding improvement in the classification accuracy of the IN, PU, and HOU datasets compared to when the spectral module was not added, with a significant increase in the AA accuracy of the IN dataset in particular. Therefore, the classification effect of the spectral correlation module is noticeable.

4.4.3 *Experiments combining spatial augmentation and spectral correlation modules*

Experiments on spatial augmentation and spectral correlation modules were used together, and the results are shown in Table 3.

As shown in Table 3, comparing the results of the experiments, we found that the classification accuracy of the IN, PU, and HOU datasets was the best when combined with the spectral correlation module at a rotation of 60 deg. The classification accuracy was also the best when comparing the spatial augmentation and the spectral correlation module separately. The two modules of the HOU dataset were the same as the separate experiments, so the effect of using the spatial augmentation and spectral correlation module separately on the HOU dataset was not significant.

4.5 Comparison with Various Algorithms

The SAS-BERT algorithm proposed in the paper is experimentally compared with other deep-learning methods. There are two types of deep-learning methods for comparison, CNN-based algorithms: CNN,[19] SSRN,[36] HybridSN,[37] and LS^{2}CM,[38] and transformer-based methods: CTN,[24] SSTN,[26] and HSI-BERT.[34]

In these experiments, the SAS-BERT input hyperspectral cube has a spatial size of 11 × 11. The Adam optimizer optimizes 200 epochs. The learning rate set to 3×10^{-4}, the batch size to 16, and the dropout rate to 0.3. Various other settings for the comparison algorithms follow the original paper. The classification accuracy for the comparison experiments is shown in Table 4.

Table 4 shows the classification results of the various classification methods. It can be analyzed that SAS-BERT outperforms the compared CNN and transformer methods on the IN, PU, and HOU datasets, significantly improving the HSI-BERT method in particular. Overall, the SAS-BERT classification method performs better than the current state-of-the-art methods.

Table 3 Classification accuracy on the IN, PU, and HOU datasets using the combined spatial augmentation and spectral correlation modules.

	IN			PU			HOU		
Datasets	OA (%)	AA (%)	$k \times 100$	OA (%)	AA (%)	$k \times 100$	OA (%)	AA (%)	$k \times 100$
SAL (dm) + spe	95.35 ± 0.59	90.56 ± 2.13	94.70 ± 0.68	97.24 ± 0.28	95.49 ± 0.47	96.33 ± 0.37	90.80 ± 0.96	90.50 ± 1.12	90.06 ± 1.04
SAL (tra) + spe	94.26 ± 0.65	90.39 ± 2.46	93.45 ± 0.74	97.08 ± 0.62	94.78 ± 1.25	96.12 ± 0.83	89.29 ± 1.06	89.53 ± 0.71	88.41 ± 1.14
SAL (dou) + spe	95.15 ± 0.37	90.55 ± 2.27	94.47 ± 0.42	97.22 ± 0.29	94.84 ± 0.93	96.30 ± 0.38	90.83 ± 0.96	90.86 ± 1.00	90.09 ± 1.04
SAL (hal) + spe	93.99 ± 1.39	89.37 ± 1.73	93.15 ± 1.57	96.62 ± 0.51	94.73 ± 0.67	95.50 ± 0.68	90.58 ± 0.97	90.72 ± 1.02	89.81 ± 1.05
SAL (30 deg) + spe	95.08 ± 0.55	90.36 ± 2.02	94.39 ± 0.63	96.92 ± 0.50	94.95 ± 0.82	95.92 ± 0.66	90.95 ± 0.65	91.22 ± 0.64	90.21 ± 0.71
SAL (45 deg) + spe	95.47 ± 0.91	91.49 ± 2.15	94.84 ± 1.04	97.20 ± 0.64	94.71 ± 1.48	96.28 ± 0.85	90.80 ± 1.12	90.83 ± 1.15	90.05 ± 1.21
SAL (60 deg) + spe	**95.64 ± 0.85**	**91.38 ± 1.95**	**95.03 ± 0.97**	**97.53 ± 0.44**	**95.87 ± 0.93**	**96.71 ± 0.58**	**91.20 ± 0.85**	**91.21 ± 0.75**	**90.48 ± 0.92**
spe-BERT	94.95 ± 0.89	90.47 ± 2.25	94.24 ± 1.02	97.41 ± 0.54	95.24 ± 1.71	96.56 ± 0.72	91.12 ± 0.69	91.06 ± 0.83	90.40 ± 0.75
SAL	94.83 ± 0.87	89.30 ± 3.01	94.11 ± 0.99	97.19 ± 0.50	95.36 ± 0.88	96.27 ± 0.66	91.12 ± 1.57	91.20 ± 1.62	90.40 ± 1.70
HSI-BERT	93.36 ± 0.88	85.01 ± 1.90	92.42 ± 1.02	96.38 ± 0.58	93.41 ± 0.90	95.19 ± 0.77	89.59 ± 2.10	87.55 ± 3.17	88.74 ± 2.28

Note: Bold values represent the optimal results under the same evaluation index.

Table 4 Comparison of classification accuracy of algorithms on IN, PU, and HOU datasets.

	IN			PU			HOU		
Methods	OA (%)	AA (%)	$k \times 100$	OA (%)	AA (%)	$k \times 100$	OA (%)	AA (%)	$k \times 100$
CNN	85.24 ± 1.14	87.54 ± 1.53	83.16 ± 1.31	78.63 ± 2.37	78.53 ± 1.63	70.72 ± 3.91	74.23 ± 2.91	79.90 ± 1.46	72.13 ± 3.15
SSRN	84.64 ± 9.88	80.16 ± 4.98	82.67 ± 10.81	94.89 ± 3.33	91.38 ± 6.41	93.23 ± 4.41	84.89 ± 4.94	86.52 ± 4.80	83.68 ± 5.32
HybridSN	86.80 ± 0.67	90.73 ± 0.56	84.89 ± 0.75	94.76 ± 1.56	93.15 ± 2.07	93.01 ± 2.11	88.78 ± 0.90	89.80 ± 0.68	87.86 ± 0.97
LS^{2}CM	92.21 ± 2.01	91.31 ± 4.34	91.12 ± 2.30	96.04 ± 2.13	94.61 ± 2.47	94.76 ± 2.80	86.87 ± 4.09	89.50 ± 2.84	85.80 ± 4.42
CTN	93.57 ± 1.09	88.07 ± 3.44	92.66 ± 1.24	95.08 ± 0.63	93.36 ± 0.87	93.47 ± 0.83	80.36 ± 2.33	81.66 ± 2.30	78.73 ± 2.53
SSTN	94.54 ± 0.60	77.53 ± 1.37	93.77 ± 0.69	96.37 ± 0.84	91.07 ± 2.79	95.17 ± 1.13	85.45 ± 1.17	82.96 ± 1.06	84.24 ± 1.27
HSI-BERT	93.36 ± 0.88	85.01 ± 1.90	92.42 ± 1.02	96.38 ± 0.58	93.41 ± 0.90	95.19 ± 0.77	89.59 ± 2.10	87.55 ± 3.17	88.74 ± 2.28
SAS-BERT	**95.49 ± 0.51**	**91.14 ± 1.05**	**94.86 ± 0.58**	**97.41 ± 0.32**	**95.63 ± 0.90**	**96.57 ± 0.43**	**91.23 ± 0.69**	**91.48 ± 0.54**	**90.52 ± 0.75**

Note: Bold values represent the optimal results under the same evaluation index.

Figures 11–13 show the classification mapping of the comparison experiments on the Indian, Pavia, and Houston datasets. From these figures, the SAS-BERT algorithm shows more accurate classification mapping results with smoother and clearer edges, and SAS-BERT has better classification results.

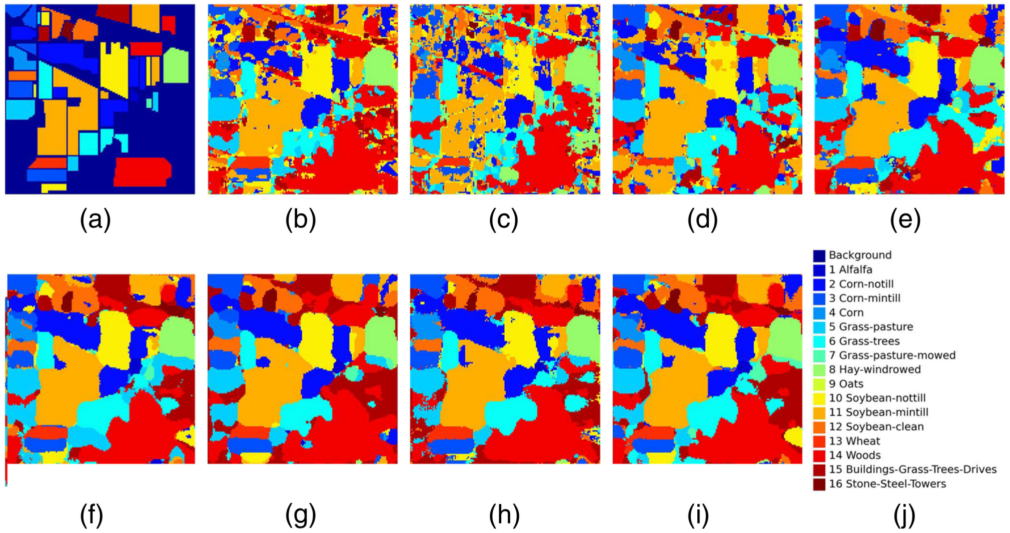


Fig. 11 Classification maps on the Indian dataset: (a) a reference map of the real ground category on the Indian dataset; a classification map of (b) CNN; (c) SSRN; (d) HybridSN; (e) LS^{2}CM; (f) CTN; (g) SSTN; (h) HSI-BERT; (i) SAS-BERT; and (j) each ground cover category marker colour for the Indian dataset.

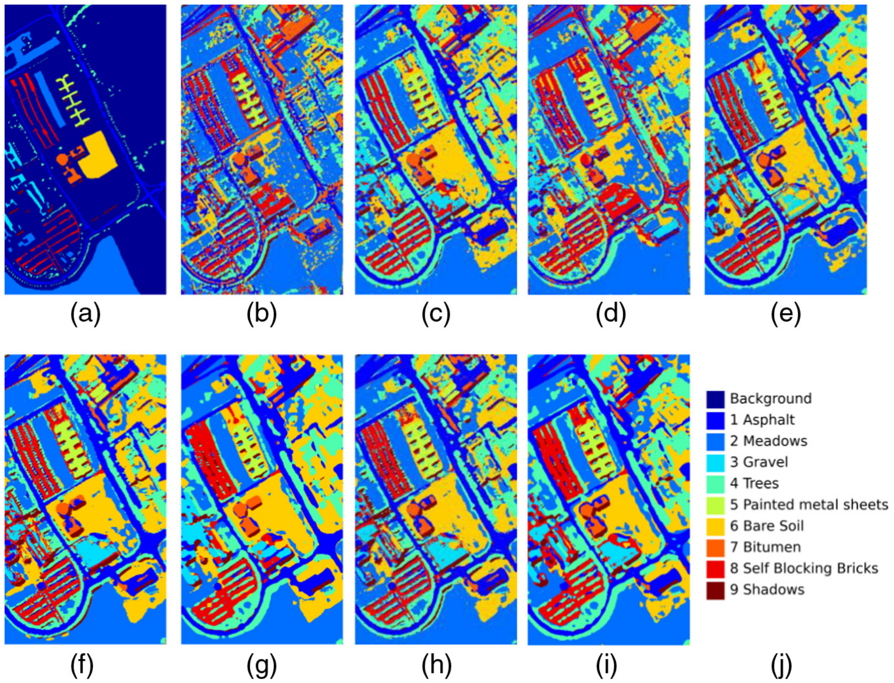


Fig. 12 Classification maps on the Pavia dataset: (a) a reference map of the real ground category on the Pavia dataset; a classification map of (b) CNN; (c) SSRN; (d) HybridSN; (e) LS^{2}CM; (f) CTN; (g) SSTN; (h) HSI-BERT; (i) SAS- BERT; and (j) each ground cover category marker colour for the Pavia dataset.

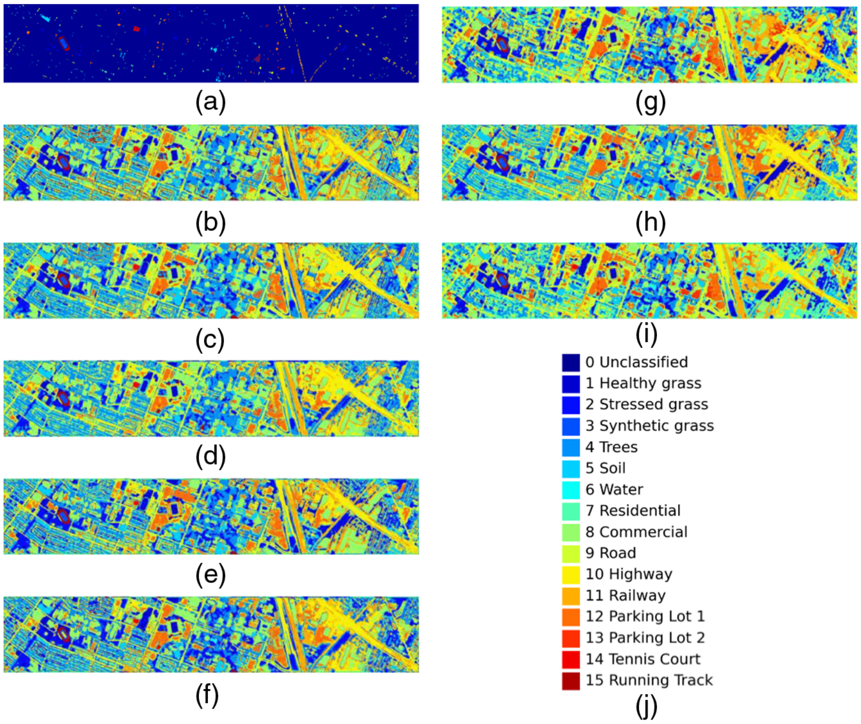


Fig. 13 Classification maps on the Houston dataset: (a) a reference map of the real ground category on the Houston dataset; a classification map of (b) CNN; (c) SSRN; (d) HybridSN; (e) LS^{2}CM; (f) CTN; (g) SSTN; (h) HSI-BERT; (i) SAS-BERT; and (j) each ground cover category marker colour for the Houston dataset.

5 Conclusion

This paper proposes a SAS-BERT method for HSI classification based on the BERT model for feature extraction of spatial and spectral information. The method improves the performance of BERT model-based classification by aggregating augmented spatial features and spectral features to represent HSI features. It improves the representative characteristics of the local spatial information using a spatial augmentation module. The module transforms the input image so that distinct representational features characterise this input. It allows for better performance in classification tasks, which helps to minimize the overall cost of the network during training. In addition, it uses the spectral properties of HSIs so that they fully reflect the internal physical structure of matter. And correlation with spatial location is established, which significantly improves the interpretation of HSIs through feature description. Results from experiments on three widely used datasets show that the SAS-BERT model outperforms the current state-of-the-art CNN and transformer network classification models.

Code, Data, and Materials Availability

The data presented in this paper are publicly available in Ref. 39. The archived version of the code described in this manuscript can be freely accessed through Github: https://github.com/zyy1234aiyou/SAS-BERT.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Project No. 62201438), the Ningxia Key R&D Program of China (Project No. 2021BEG03030), and the Image and Intelligence Information Processing Innovation Team of the National Ethnic Affairs Commission of China.

References

1. M. Imani and H. Ghassemian, “An overview on spectral and spatial information fusion for hyperspectral image classification: current trends and challenges,” *Inf. Fusion* **59**, 59–83 (2020).
2. X. Xu et al., “Multisource remote sensing data classification based on convolutional neural network,” *IEEE Trans. Geosci. Remote Sens.* **56**(2), 937–949 (2017).
3. M. Zhang, W. Li, and Q. Du, “Diverse region-based CNN for hyperspectral image classification,” *IEEE Trans. Image Process* **27**(6), 2623–2634 (2018).
4. M. Ahmad et al., “Hyperspectral image classification-traditional to deep models: a survey for future prospects,” *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **15**, 968–999 (2022).
5. B. Tu et al., “Spectral–spatial hyperspectral classification via structural-kernel collaborative representation,” *IEEE Geosci. Remote Sens. Lett.* **18**(5), 861–865 (2021).
6. Z. Xia et al., “Crop classification based on feature band set construction and object-oriented approach using hyperspectral images,” *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **9**(9), 4117–4128 (2016).
7. K. R. Manjunath, S. S. Ray, and D. Vyas, “Identification of indices for accurate estimation of anthocyanin and carotenoids in different species of flowers using hyperspectral data,” *Remote Sens. Lett.* **7**(10), 1004–1013 (2016).
8. E. M. Paoletti et al., “Capsule networks for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.* **57**(4), 2145–2160 (2019).
9. L. Ni, H. Xu, and X. Zhou, “Mineral identification and mapping by synthesis of hyperspectral VNIR/SWIR and multispectral TIR remotely sensed data with different classifiers,” *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **13**, 3155–3163 (2020).
10. J. Zhou et al., “A novel cluster kernel RX algorithm for anomaly and change detection using hyperspectral images,” *IEEE Trans. Geosci. Remote Sens.* **54**(11), 6497–6504 (2016).
11. C. Shang et al., “Spectral-spatial generative adversarial network for super-resolution land cover mapping with multispectral remotely sensed imagery,” *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **16**, 522–537 (2023).
12. H. Liang et al., “Spectral–spatial attention feature extraction for hyperspectral image classification based on generative adversarial network,” *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **14**, 10017–10032 (2021).
13. R. Sunkara, A. K. Singh, and G. R. Kadambi, “Class information-based principal component analysis algorithm for improved hyperspectral image classification,” in *Int. Conf. Mach. Intell. for GeoAnalytics and Remote Sens. (MIGARS)*, Vol. 1, pp. 1–4 (2023).
14. G. Liu et al., “Hyperspectral image classification based on fuzzy nonparallel support vector machine,” in *Global Conf. Rob., Artif. Intell. and Inf. Technol. (GCRAIT)*, pp. 242–246 (2022).
15. J. Xia et al., “Random forest ensembles and extended multiextinction profiles for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.* **56**(1), 202–216 (2017).
16. J. Li, J. M. Bioucas-Dias, and A. Plaza, “Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields,” *IEEE Trans. Geosci. Remote Sens.* **50**(3), 809–823 (2012).
17. X. Kang, S. Li, and J. A. Benediktsson, “Spectral–spatial hyperspectral image classification with edge-preserving filtering,” *IEEE Trans. Geosci. Remote Sens.* **52**(5), 2666–2677 (2013).
18. A. Mughees and L. Tao, “Hyper-voxel based deep learning for hyperspectral image classification,” in *IEEE Int. Conf. on Image Process. (ICIP)*, IEEE, pp. 840–844 (2017).
19. Y. Chen et al., “Deep feature extraction and classification of hyperspectral images based on convolutional neural networks,” *IEEE Trans. Geosci. Remote Sens.* **54**(10), 6232–6251 (2016).
20. J. Li et al., “Classification of hyperspectral imagery using a new fully convolutional neural network,” *IEEE Geosci. Remote Sens. Lett.* **15**(2), 292–296 (2018).
21. Z. Zhong et al., “Deep residual networks for hyperspectral image classification,” in *IEEE Int. Geosci. and Remote Sens. Symp. (IGARSS)*, IEEE, pp. 1824–1827 (2017).
22. M. Kanthi, T. H. Sarma, and C. S. Bindu, “A 3D-deep CNN based feature extraction and hyperspectral image classification,” in *IEEE India Geosci. and Remote Sens. Symp. (InGARSS)*, IEEE, pp. 229–232 (2020).
23. X. He and Y. Chen, “Optimized input for CNN-based hyperspectral image classification using spatial transformer network,” *IEEE Geosci. Remote Sens. Lett.* **16**(12), 1884–1888 (2019).
24. H. W. Z. Zhao, D. Hu, and X. Yu, “Convolutional transformer network for hyperspectral image classification,” *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5 (2022).
25. D. Hong et al., “Spectralformer: rethinking hyperspectral image classification with transformers,” *IEEE Trans. Geosci. Remote Sens.* **60**, 1–15 (2021).
26. Z. Zhong et al., “Spectral–spatial transformer network for hyperspectral image classification: a factorized architecture search framework,” *IEEE Trans. Geosci. Remote Sens.* **60**, 1–15 (2021).
27. J. Devlin et al., “BERT: pre-training of deep bidirectional transformers for language understanding,” arXiv:1810.04805 (2018).

28. Y. Liu et al., "RoBERTa: a robustly optimized BERT pretraining approach," arXiv:1907.11692 (2019).
29. Z. Lan et al., "ALBERT: a lite BERT for self-supervised learning of language representations," arXiv:1909.11942 (2019).
30. Y. Cui et al., "Revisiting pre-trained models for Chinese natural language processing," arXiv:2004.13922 (2020).
31. K. Song et al., "MASS: masked sequence to sequence pre-training for language generation," arXiv:1905.02450 (2019).
32. L. Dong et al., "Unified language model pre-training for natural language understanding and generation," in *Adv. in Neural Inf. Process. Syst.*, Vol. 32 (2019).
33. M. Joshi et al., "SpanBERT: improving pre-training by representing and predicting spans," *Trans. Assoc. Comput. Linguist.* **8**, 64–77 (2020).
34. H. Ji et al., "HSI-BERT: hyperspectral image classification using the bidirectional encoder representation from transformers," *IEEE Trans. Geosci. Remote Sens.* **58**(1), 165–178 (2019).
35. M. Gong et al., "A spectral and spatial attention network for change detection in hyperspectral images," *IEEE Geosci. Remote Sens.* **60**, 1–14 (2022).
36. Z. Zhong et al., "Spectral-spatial residual network for hyperspectral image classification: a 3D deep learning framework," *IEEE Trans. Geosci. Remote Sens.* **56**(2), 847–858 (2017).
37. S. K. Roy et al., "HybridSN: exploring 3D–2D CNN feature hierarchy for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.* **17**(2), 277–281 (2019).
38. Z. Meng et al., "A lightweight spectral-spatial convolution module for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5 (2021).
39. M. Graña, M. A. Veganzons, and B. Ayerdi, "Hyperspectral Remote Sensing Scenes," Grupo de Inteligencia Computacional (GIC), http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes (2021).

Yuanyuan Zhang is currently pursuing her MEng degree from the School of Computer Science and Engineering, North Minzu University, Yinchuan, China. She received her BS degree in software engineering from North Minzu University, Yinchuan, China, in 2020. Her current research interests include hyperspectral image classification, image processing, artificial intelligence, and deep learning.

Wenxing Bao received his BEng degree in industrial automation from Xidian University, Xi'an, China, in 1993, and his MSc degree in electrical engineering and his PhD in electronic science and technology from Xi'an Jiaotong University, Xi'an, China, in 2001 and 2006, respectively. He is currently a professor and a vice president of North Minzu University, Yinchuan, China. His research interests include digital image processing, remote sensing image classification, and fusing.

Hongbo Liang received his BS degree in computer science and technology from North Minzu University, Yinchuan, China, in 2018 and his MEng degree from the School of Computer Science and Engineering, North Minzu University, Yinchuan, China, in 2021. He is currently pursuing his PhD in communication engineering from Hefei University of Technology, Hefei, China. His research interests include hyperspectral image processing, SAR image processing, remote sensing image classification, computer vision, and deep learning.

Yanbo Sun is currently pursuing his MEng degree at the School of Computer Science and Engineering, North Minzu University, Yinchuan, China. He received his BS degree in software engineering from Nanyang Normal University, Nanyang, China, in 2022. His current research interests include hyperspectral image change detection, hyperspectral image classification, and deep learning.