

Article

An SVM-Based Nested Sliding Window Approach for Spectral–Spatial Classification of Hyperspectral Images

Jiansi Ren ^{1,2} , Ruoxiang Wang ¹ , Gang Liu ^{1,2,*}, Yuanni Wang ^{1,2} and Wei Wu ¹

¹ School of Computer Science, China University of Geosciences, Wuhan 430078, China; renjsv@cug.edu.cn (J.R.); jojo@cug.edu.cn (R.W.); wangyn@cug.edu.cn (Y.W.); wuwei_2019@cug.edu.cn (W.W.)

² Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430078, China

* Correspondence: liugang@cug.edu.cn

Abstract: This paper proposes a Nested Sliding Window (NSW) method based on the correlation between pixel vectors, which can extract spatial information from the hyperspectral image (HSI) and reconstruct the original data. In the NSW method, the neighbourhood window constructed with the target pixel as the centre contains relevant pixels that are spatially adjacent to the target pixel. In the neighbourhood window, a nested sliding sub-window contains the target pixel and a part of the relevant pixels. The optimal sub-window position is determined according to the average value of the Pearson correlation coefficients of the target pixel and the relevant pixels, and the target pixel can be reconstructed by using the pixels and the corresponding correlation coefficients in the optimal sub-window. By combining NSW with Principal Component Analysis (PCA) and Support Vector Machine (SVM), a classification model, namely NSW-PCA-SVM, is obtained. This paper conducts experiments on three public datasets, and verifies the effectiveness of the proposed model by comparing with two basic models, i.e., SVM and PCA-SVM, and six state-of-the-art models, i.e., CDCT-WF-SVM, CDCT-2DCT-SVM, SDWT-2DWT-SVM, SDWT-WF-SVM, SDWT-2DCT-SVM and Two-Stage. The proposed approach has the following advantages in overall accuracy (OA)—take the experimental results on the Indian Pines dataset as an example: (1) Compared with SVM (OA = 53.29%) and PCA-SVM (OA = 58.44%), NSW-PCA-SVM (OA = 91.40%) effectively utilizes the spatial information of HSI and improves the classification accuracy. (2) The performance of the proposed model is mainly determined by two parameters, i.e., the window size in NSW and the number of principal components in PCA. The two parameters can be adjusted independently, making parameter adjustment more convenient. (3) When the sample size of the training set is small (20 samples per class), the proposed NSW-PCA-SVM approach achieves 2.38–18.40% advantages in OA over the six state-of-the-art models.

Keywords: spatial–spectral feature integration; nested sliding window; hyperspectral remote sensing; support vector machine



Citation: An SVM-Based Nested Sliding Window Approach for Spectral–Spatial Classification of Hyperspectral Images. *Remote Sens.* **2021**, *13*, 114. <https://doi.org/10.3390/rs13010114>

Received: 6 December 2020

Accepted: 28 December 2020

Published: 31 December 2020

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Hyperspectral remote sensing images contain rich spectral features, which can provide effective information for the classification tasks and/or other tasks [1]. Hyperspectral imaging technology is widely used in the field of remote sensing, including military target detection [2], urban land planning [3], and vegetation coverage analysis [4]. In addition, it also has applications in the fields of medical and health [5] and plant disease disasters [6]. This paper mainly studies the general classification task of HSIs, which is to identify the object category of each pixel in the image. In the early studies, individual pixels were often used as the target to train the classification model [7,8]. In fact, HSIs also have spatial characteristics, which means that adjacent pixels often belong to the same category. In recent years, the research on spatial–spectral feature integration has attracted more and more attention.

In the task of HSI classification, many machine learning algorithms, such as Random Forest [9], K-Nearest Neighbor [10], Linear Discriminant Analysis [11], and Support Vector Machine (SVM) [7], are often used as classifiers. Since the Convolutional Neural Network naturally has the function of extracting spatial features, it is also often used for HSI classification [12,13]. However, when the sample size of the training set is small, CNN is difficult to train to obtain stable results. On the contrary, SVM not only converges and stabilizes, but also can be well applied to the case of a small sample size [14]. Due to the high cost of manual sample labeling of HSIs [15], it is very meaningful to explore the classification with a small sample size. In many current researches, it is also very popular to combine SVM as a classifier with the spatial-spectral feature integration algorithm [16,17]. These algorithms are mainly divided into two categories, namely “pre-processing before classification” and “processing after classification”.

The “pre-processing before classification” method refers to a pre-processing method that reconstructs data before classification. In this process, the spatial features of hyper-spectral data will be extracted and fused with spectral features. The method based on mathematical morphology is one of the methods often used to extract spatial features [18], and many researchers have made improvements on this basis. For example, Liao W et al. proposed a morphological profiles (MPs) classification method based on partial reconstruction and directional MPs [19]. On this basis, they proposed a semi-supervised feature extraction method to reduce the dimensionality of the generated MPs. Hou B et al. proposed a 3-D morphological profile (3D-MP) based method to utilize the dependence between data to improve the classification accuracy [20]. Imani M et al. pointed out that the use of fixed-shape for structural elements cannot extract the profile information effectively, so they proposed a method to extract edge patch image-based MPs [21]. Kumar B et al. used multi-shape structuring elements instead of ones with a particular shape, and then used the decision fusion method to fuse the classification results obtained based on different Extended Morphological Profiles (EMPs) [22]. In addition, Kumar B et al. also used parallel computing to improve the speed of the EMP algorithm [23].

In recent years, the spatial-spectral feature integration method based on spatial filtering has also been often used in research. Various filtering methods (e.g., Gabor Filtering, Wiener Filtering) have been applied. Gabor Filtering is often used in combination with other algorithms to improve the model. For example, Wu K et al. combined Simple Linear Iterative clustering, two-dimensional Gabor Filtering and Sparse Representation, and proposed a SP-Gabor classification method [24]. Jia S et al. proposed an extended Gabor wavelet based on morphology, combining the advantages of EMP operator and Gabor wavelet transform [25]. In addition, there are many studies that combine Gabor filtering with convolutional neural networks [26,27]. Wiener filtering is also often used in combination with other models or methods to improve classification performance. For example, in the CDCT-WF-SVM model proposed by Bazine R et al., after using Discrete Cosine Transform (DCT) to process the data, Wiener filtering is used to spatially filter the high-frequency components to further extract useful information [28]. In fact, the role of Wiener Filtering is mainly to reduce noise [29]. In the current research on denoising of HSIs, the additional photon noise related to the signal has become a research hotspot. For example, Liu X et al. used pre-whitening processing to transform the non-white noise in HSIs into white noise, and then used multidimensional Wiener filtering to denoise [30]. In addition to the above two kinds of filtering, filtering methods such as mean filtering [31,32] and edge preservation filtering [33,34] are also often used to extract spatial features of HSI.

The “processing after classification” method refers to first using a classifier to obtain the predicted probability map, and then using one or more processing methods to process the obtained probability map. Methods based on Markov Random Field (MRF) have been widely studied, and they are mainly used to further process the pixel classification results obtained by the classifier [35,36]. For example, Qing C et al. proposed a deep learning framework in which a convolutional neural network is used as a pixel classifier, and MRF is used for spatial information mining and pixel classification results [37]. Chakravarty

S et al. pointed out that the model that simply combines the SVM classifier and MRF cannot enhance the smoothness of spatial and spectral analysis. Therefore, they used fuzzy MRF to promote a smooth transition between classified pixels [38]. Xu Y et al. optimized the model by inserting the watershed algorithm in the process of combining SVM and MRF [39]. Tang B et al. proposed a classification framework based on the Spectral Angle Mapper (SAM) to obtain a more accurate classification by introducing the multi-center model and MRF into the probabilistic decision framework [40]. For the problem that shallow MRF cannot fully utilize the spatial information of HSIs, Cao X et al. proposed a cascaded MRF model, which further improved the classification performance of the model [41].

Similar to the MRF-based model, the graph segmentation (GC) based model is also used to further process the pixel classification results to effectively use the spatial information of the hyperspectral image. Wang Y et al. proposed a classification model based on joint bilateral filtering (JBF) and graph segmentation. The model first used SVM to classify pixels, and then used JBF and GC to smooth the obtained probability map [42]. Yu H et al. combined the MRF and GC algorithm, and successively processed the classification probability map obtained by SVM, which further improved the classification accuracy [43]. In addition, there are also studies that combine “pre-processing before classification” and “processing after classification”. For example, Liao W et al. proposed an adaptive Bayesian context classification model. The model first fused the spatial-spectral features of the original data based on extended morphology, and then used Markov random field to obtain the probability after classification. The feature map was processed [44]. Cao X et al. used the low-rank matrix factorization (LRMF) method based on the Gaussian mixture algorithm to extract features before classification, and used the combination of SVM and MRF to classify the data [45]. The edge retention filtering method mentioned above can actually be used in the model after classification [46,47], or used before and after classification [48].

Since HSIs are obtained by continuous imaging of the target area by the sensor, the probability that adjacent pixels belong to the same category of features is high. In the method based on image segmentation, the segmented area is considered to be composed of homogeneous pixels [49]. The spectral features of the pixels belonging to the same category should be similar, so the correlation between two similar pixels should be relatively high [50]. Based on this assumption, this paper proposed a Nested Sliding Window (NSW) method, which uses the correlation between pixel vectors to reconstruct the data in the pre-processing stage. In the NSW-PCA-SVM model, the PCA method is used to reduce the dimension of the reconstructed data, and the RBF-kernel SVM is used for classification.

Various popular methods were usually improved on the basis of some original methods or obtained by combining existing algorithms. For example, CDCT-WF-SVM model based on Discrete Cosine Transform (DCT) algorithm and Adaptive Wiener Filter (AWF) algorithm was obtained by combining existing algorithms [28]. Although the final classification accuracy can be significantly improved by improving or skillfully combining existing algorithms, we try to propose a new algorithm from a more intuitive perspective, hoping to provide additional ideas for future research.

The rest of this article is arranged as follows. Section 2 introduces the proposed model, including the structure of the model and the implementation details of the NSW method; Section 3 introduces the dataset used in this article, and gives the verification process of the model performance; Section 4 gives the experimental results and the corresponding analysis, including the measurement of model parameters and the specific classification results; Section 5 discusses the advantages and limitations of the proposed approach; Section 6 is the conclusion, which will summarize the work of this article.

2. The Proposed Approach

The proposed model consists of three parts: the reconstruction of the original data using NSW method; The dimensionality reduction of the reconstructed data by using PCA method; The classification of the data after dimensionality reduction using SVM. Figure 1 shows the structure diagram of the NSW-PCA-SVM model.

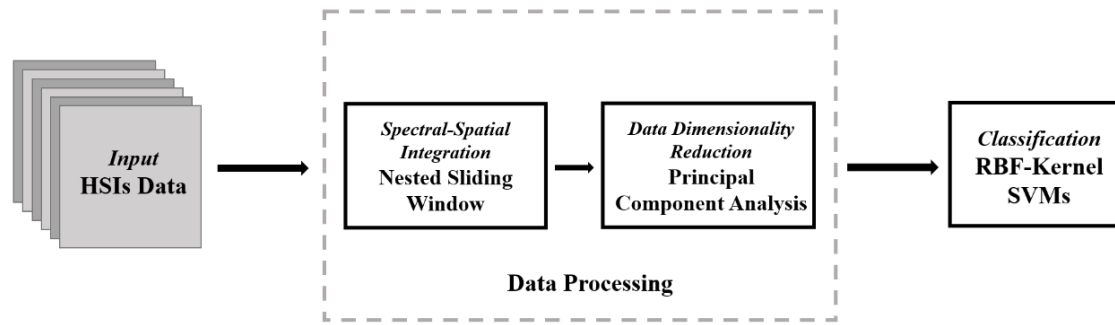


Figure 1. Structure of the proposed Nested Sliding Window (NSW)-Principal Component Analysis (PCA)-Support Vector Machine (SVM) model.

The original hyperspectral data is a three-dimensional cube, which can be expressed as $\mathbf{X} \in \mathbb{R}^{(H,W,B)}$, where $\mathbb{R}$ represents the set of real numbers and (H, W, B) represents that the image has three dimensions, including two spatial dimensions (i.e., H and W) and one spectral dimension (i.e., B). For $\mathbf{X} \in \mathbb{R}^{(H,W,B)}$, there are a total of $H \times W$ pixels, and each pixel is a B -dimensional vector. In fact, not all pixels in HSIs have a category label, and only pixels with labels will be used as target pixels for reconstruction. Therefore, the data reconstructed by NSW is transformed into a two-dimensional matrix, i.e., $\mathbf{X}' \in \mathbb{R}^{(N,B)}$, where N represents the number of pixels with labels. Dimensionality reduction using PCA maps $\mathbf{X}'$ to $\mathbf{X}'' \in \mathbb{R}^{(N,C)}$, where C represents the number of principal components, and $C \ll B$. According to the sample size of the training set specified in Section 3, a specified number of samples are randomly selected from $\mathbf{X}''$ as the training set, and the remaining samples are all as the test set, in which the training set is used to train SVM.

2.1. The Nested Sliding Window Method

Due to the continuity of the hyperspectral imaging area, it can theoretically be considered that adjacent pixels have the same category label. However, at the junction of the areas where two different ground targets are located, adjacent pixels obviously do not share the same label. The nested sliding window (NSW) method proposed in this paper uses the correlation between HSI pixels to reconstruct the data, and generally uses the Pearson correlation coefficient to measure the correlation. For pixel vectors $\mathbf{x}$ and $\mathbf{y}$, the Pearson correlation coefficient is calculated as follows:

$$\text{corr}(\mathbf{x}, \mathbf{y}) = \frac{\text{Cov}(\mathbf{x}, \mathbf{y})}{\sqrt{\text{Var}(\mathbf{x}) \cdot \text{Var}(\mathbf{y})}}, \quad (1)$$

where $\text{Cov}(\cdot, \cdot)$ represents covariance, and $\text{Var}(\cdot)$ represents variance.

Denote the pixel vector in row i and column j of $\mathbf{X}$ as $\mathbf{X}_{ij}$, and assume that $\mathbf{X}_{ij}$ is the target pixel. define a neighborhood with a size of (ω, ω, B) that contains the surrounding pixels of $\mathbf{X}_{ij}$:

$$N(\mathbf{X}_{ij}) = \{\mathbf{X}_{pq} \mid p \in [i-a, i+a], q \in [j-a, j+a]\}, \quad (2)$$

where $a < i \leq H-a$, $a < j \leq W-a$, $a = (\omega-1)/2$.

Considering that Formula (2) cannot be used to obtain the neighborhood when the target pixel is at the edge position of $\mathbf{X}$ in the spatial dimension, i.e., $i \leq a$, $i > H-a$, $j \leq a$, or $j > W-a$, the data needs to be zero-padding before obtain the neighborhood. Perform zero-padding operation on $\mathbf{X}$ to get $\mathbf{X}^P \in \mathbb{R}^{(H+2a, W+2a, B)}$:

$$\begin{cases} \mathbf{X}_{ij}^P = \mathbf{0}^{(B)}, & \text{if } 1 \leq i \leq a \text{ and/or } 1 \leq j \leq a \\ \mathbf{X}_{ij}^P = \mathbf{X}_{(i-a)(j-a)}, & \text{if } a < i \leq H+a \text{ and/or } a < j \leq W+a \\ \mathbf{X}_{ij}^P = \mathbf{0}^{(B)}, & \text{if } H+a < i \leq H+2a \text{ and/or } W+a < j \leq W+2a \end{cases} \quad (3)$$

After zero-padding is performed on the raw data, the neighborhood of the target pixel $\mathbf{X}_{ij}^P$ can be obtained according to Formula (2), denoted as $N(\mathbf{X}_{ij}^P)$. Set a sliding sub-window with a size of $((a+1), (a+1))$ in the neighborhood. Using this sub-window, a smaller three-dimensional matrix $W_{mn}(\mathbf{X}_{ij}^P)$ can be divided in $N(\mathbf{X}_{ij}^P)$, where $W_{mn}(\mathbf{X}_{ij}^P) \subset N(\mathbf{X}_{ij}^P)$, m and n are used to determine the position of the sliding window, and $0 \leq m \leq a$, $0 \leq n \leq a$. $W_{mn}(\mathbf{X}_{ij}^P)$ is defined as follows:

$$W_{mn}(\mathbf{X}_{ij}^P) = \{\mathbf{X}_{pq}^P \mid p \in [i-a+m, i+m], q \in [j-a+n, j+n]\}, \quad (4)$$

where $a < i \leq H+a$ and $a < j \leq W+a$.

Within the valid range of m and n , $W_{mn}(\mathbf{X}_{ij}^P)$ always contains the target pixel $\mathbf{X}_{ij}^P$. Therefore, the correlation coefficients between $\mathbf{X}_{ij}^P$ and the pixel vectors in $W_{mn}(\mathbf{X}_{ij}^P)$ can be calculated according to Formula (1), and the calculated correlation coefficient matrix be denoted as $C_{mn}(\mathbf{X}_{ij}^P)$:

$$C_{mn}(\mathbf{X}_{ij}^P) = \{c_{pq} \mid p \in [i-a+m, i+m], q \in [j-a+n, j+n]\} \quad (5)$$

Let the mean value of the elements in the matrix $C_{mn}(\mathbf{X}_{ij}^P)$ be $\bar{c}_{mn}$. When the position of the sliding sub-window changes, i.e., when the values of m and n change, $\bar{c}_{mn}$ will change accordingly. Assuming $\bar{c}_{kl} = \max(\bar{c}_{mn})$, then the $W_{kl}(\mathbf{X}_{ij}^P)$ and $C_{kl}(\mathbf{X}_{ij}^P)$ corresponding to $\bar{c}_{kl}$ can be obtained according to Formulas (4) and (5), respectively, which would be used to reconstruct the target pixel. First, change the shape of $W_{kl}(\mathbf{X}_{ij}^P)$ from $((a+1), (a+1), B)$ to $((a+1) \times (a+1), B)$, and change the shape of $C_{kl}(\mathbf{X}_{ij}^P)$ from $((a+1), (a+1))$ to $((a+1) \times (a+1), 1)$. Second, expand the reshaped $C_{kl}(\mathbf{X}_{ij}^P)$ from a one-dimensional vector to a two-dimensional matrix, denoted as $C'_{kl}(\mathbf{X}_{ij}^P)$:

$$C'_{kl}(\mathbf{X}_{ij}^P) = \{\mathbf{x}_{:q} \mid \mathbf{x}_{:q} = C_{kl}(\mathbf{X}_{ij}^P), q \in [1, B]\} \quad (6)$$

The formula for reconstruction of $\mathbf{X}_{ij}^P$ is as follows:

$$R(\mathbf{X}_{ij}^P) = \frac{C'_{kl}(\mathbf{X}_{ij}^P) \odot W_{kl}(\mathbf{X}_{ij}^P)}{\text{Sum}(C_{kl}(\mathbf{X}_{ij}^P))}, \quad (7)$$

where $R(\mathbf{X}_{ij}^P)$ is the reconstructed pixel of the current target pixel, which is a one-dimensional vector; $C'_{kl}(\mathbf{X}_{ij}^P) \odot W_{kl}(\mathbf{X}_{ij}^P)$ represents the element-wise product (i.e., the Hadamard product) of $C'_{kl}(\mathbf{X}_{ij}^P)$ and $W_{kl}(\mathbf{X}_{ij}^P)$; $\text{Sum}(\cdot)$ represents the sum of the elements in a vector (or matrix).

Every time m and n change, a calculation of $C_{mn}(\mathbf{X}_{ij}^P)$ is performed, which actually causes a lot of repeated calculations. Therefore, it is better to calculate the correlation coefficient of $\mathbf{X}_{ij}^P$ and the pixels in the neighborhood $N(\mathbf{X}_{ij}^P)$ at the beginning. The resulting correlation coefficient matrix is as follows:

$$C(\mathbf{X}_{ij}^P) = \{c_{pq} \mid p \in [i-a, i+a], q \in [j-a, j+a]\} \quad (8)$$

where c_{pq} represents the correlation coefficient between $\mathbf{X}_{pq}^P$ and $\mathbf{X}_{ij}^P$. Therefore, $C_{mn}(\mathbf{X}_{ij}^P)$ can be obtained directly by partition in $C(\mathbf{X}_{ij}^P)$.

In fact, only a fraction of the pixels in the dataset used for the experiment have category labels. If all the pixels in the dataset are reconstructed, there will be a lot of useless computational overhead. Therefore, a pixel needs to be reconstructed based on whether it has a category label. For the three-dimensional hyperspectral dataset $\mathbf{X} \subset \mathbb{R}^{(H,W,B)}$,

there is a corresponding label matrix $L \subset \mathbb{R}^{(H,W)}$, and the elements in L correspond to the pixel vectors in X . Generally, pixels without a category label are uniformly assigned a 0-label to facilitate culling during analysis. For this reason, in the NSW method, pixels with a label of 0 will be skipped without processing; pixels with a label of non-zero will be used as the target pixels for reconstruction.

The pseudo code of NSW is shown in Algorithm 1.

Algorithm 1 Nested Sliding Window algorithm.

Input: the raw dataset: $X \subset \mathbb{R}^{(H,W,B)}$, the category label matrix: $L \subset \mathbb{R}^{(H,W)}$, the size of the neighborhood: ω

Output: the reconstructed dataset: R , the label set: I

```

1: Initialize the reconstructed dataset, which is a 2-D matrix:  $R \leftarrow \emptyset$ 
2: Initialize the label set, which is a 1-D vector:  $I \leftarrow \emptyset$ 
3: /*Obtain the neighborhoods of pixels labeled non-zero.*/
4: Initialize the neighborhood set, which is a 4-D tensor:  $N \leftarrow \emptyset$ 
5:  $X^P \leftarrow$  the matrix obtained by zero-padding on  $X$  according to formula 3
6:  $a \leftarrow (\omega - 1)/2$ 
7: for  $i \leftarrow a + 1$  to  $H + a$  do
8:   for  $j \leftarrow a + 1$  to  $W + a$  do
9:     if  $L[i, j] \neq 0$  then
10:        $N \leftarrow N \cup \{X_{pq}^P \mid p \in [i - a, i + a], q \in [j - a, j + a]\};$ 
11:        $I \leftarrow I \cup \{L[i, j]\};$ 
12:     end if
13:   end for
14: end for
15: /*According to the obtained neighborhood tensor  $N$ , reconstruct the data.*/
16:  $n \leftarrow N.shape[0]$  ▷ Get the first dimension of  $N$ .
17: for  $k \leftarrow 1$  to  $n$  do
18:    $W \leftarrow N_{k,:};$ 
19:    $C \leftarrow$  The correlation coefficient matrix calculated using Formula (8);
20:    $W_{best} \leftarrow \emptyset;$ 
21:    $C_{best} \leftarrow \emptyset;$ 
22:   for  $l \leftarrow 1$  to  $a + 1$  do
23:     for  $m \leftarrow 1$  to  $a + 1$  do
24:        $W_{temp} \leftarrow \{W_{pq} \mid p \in [l, l + a], q \in [m, m + a]\};$ 
25:        $C_{temp} \leftarrow \{C_{pq} \mid p \in [l, l + a], q \in [m, m + a]\};$ 
26:       if  $Avg(C_{best}) < Avg(C_{temp})$  then
27:          $C_{best} \leftarrow C_{temp};$ 
28:          $W_{best} \leftarrow W_{temp};$ 
29:       end if
30:     end for
31:   end for
32:    $R_k \leftarrow$  The reconstructed pixel vector calculated using Formula (7);

```

33: $R \leftarrow R \cup \{R_k\};$

34: **end for**

An example is given to graphically illustrate the general processing flow of the NSW method, as shown in Figure 2.

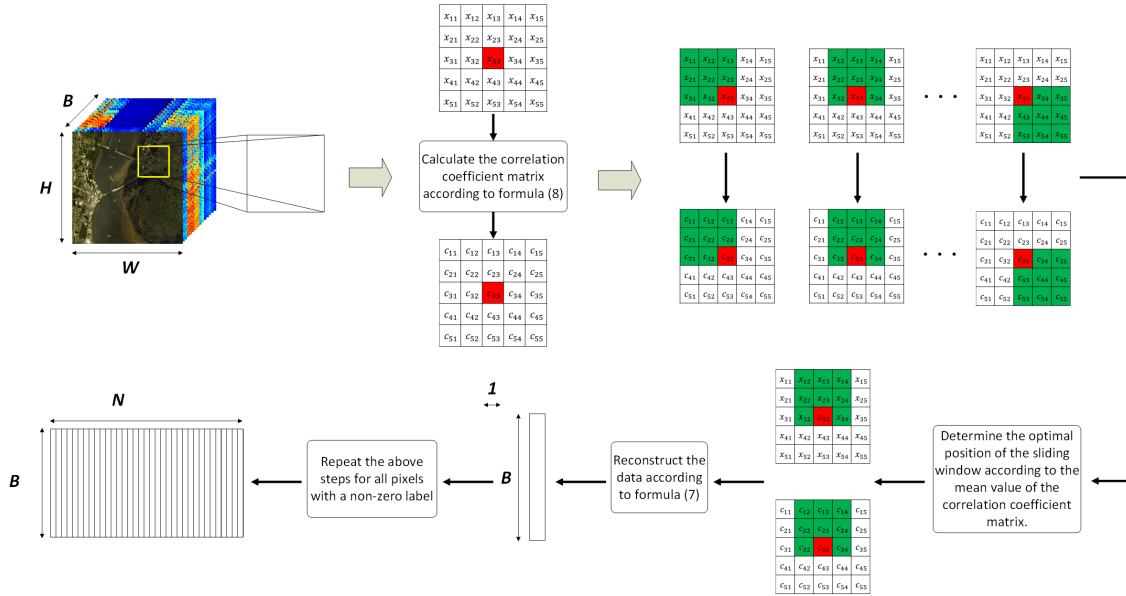


Figure 2. An example of the general flow of the NSW method.

In the example given in Figure 2, the size of the neighborhood is $(5, 5, B)$. By calculating the correlation coefficients between the target pixel and all pixels in the neighborhood, a correlation coefficient matrix of size $(5, 5)$ is obtained. Whenever the position of the sliding sub-window changes, a subset of pixels with a size of $(3, 3, B)$ is partitioned from the neighborhood, and a corresponding subset of correlation coefficients with a size of $(3, 3)$ is obtained, too. The pixel subset contains 1 target pixel (marked in red) and 8 adjacent pixels (marked in green). By comparing the mean values of different correlation coefficient matrix subsets, the maximum one corresponds to the optimal position of the sliding sub-window. Assuming that the correlation coefficient matrix subset corresponding to the optimal sub-window in Figure 2 is $C_{\text{best}} = [c_{12}, c_{13}, c_{14}, c_{22}, c_{23}, c_{24}, c_{32}, c_{33}, c_{34}]$, and the corresponding pixel subset is $X_{\text{best}} = [x_{12}, x_{13}, x_{14}, x_{22}, x_{23}, x_{24}, x_{32}, x_{33}, x_{34}]$. When performing reconstruction, first expand C_{best} from a one-dimensional vector to a two-dimensional matrix (denoted as C'_{best}). In the expanded matrix, the value of each row is the same, that is, all are C_{best} . The reconstruction is carried out according to Formula (7), namely:

$$x_{\text{recon}} = \frac{C'_{\text{best}} \odot X_{\text{best}}}{\text{Sum}(C_{\text{best}})}. \quad (9)$$

2.2. Dimensionality Reduction and Classifier

In the proposed NSW-PCA-SVM model, PCA [51] is used for dimensionality reduction of the data reconstructed by NSW, and RBF-kernel SVM [51] is used for classification. In Section 2.2, PCA and RBF-kernel SVM are introduced in two subsections, respectively.

2.2.1. Principal Component Analysis

Principal component analysis (PCA) is one of the most commonly used dimensionality reduction algorithms. It learns the low-dimensional representation of data to achieve the purpose of dimensionality reduction and denoising.

After the reconstruction of the original data using NSW method, a two-dimensional matrix with B rows and N columns is obtained, denoted as R . The matrix contains n

samples, and each sample is a B-dimensional vector. Before using PCA for dimensionality reduction, it is necessary to centralize the reconstructed data, that is, $\sum_{i=1}^n r_i = 0$, where $r_i \in R$ represents a sample in the reconstructed dataset.

Assuming that the reconstructed data needs to be reduced from B-dimension to d-dimension, the purpose of PCA is to find a two-dimensional transformation matrix with B rows and d columns, denoted as W. The variance of the projected sample points is expected to be maximized, where the projection of the sample point r_i is $W^T r_i$. Therefore, the variance of the projected sample is $\sum_{i=1}^n W^T r_i r_i^T W$, and the optimization goal is as follows:

$$\operatorname{argmax}_W \operatorname{tr}(W^T R R^T W), \quad (10)$$

where $\operatorname{tr}(\cdot)$ represents the trace of the matrix, and each column vector (denoted as w_i ($i = 1, 2, \dots, d$)) in W is an standard orthonormal basis vector, which means, $\|w_i\|_2 = 1$, $w_i^T w_j = 0$ ($i \neq j$) and $W^T W = I$.

Using Lagrange Multiplier Method for Equation (10), $R R^T w_i = \lambda_i w_i$, where λ_i represents the eigenvalue of the covariance matrix $R R^T$. Assuming that the eigenvalues are sorted: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$, then $W = \{w_1, w_2, \dots, w_d\}$, and the dimensionality reduction data is $D = W^T R$.

2.2.2. RBF-Kernel Support Vector Machine

Support Vector Machine (SVM) is one of the most influential algorithms in supervised classifiers, which is used to solve binary-classification tasks. By combining multiple binary-class SVMs, a multi-class classifier can be obtained. A common combination strategy is One-Versus-One (OVO), which implements multi-classification by designing SVMs between every two categories. Assuming classification for a dataset with k categories, the number of binary-class SVMs that OVO-SVMs need to design is $k \times (k - 1)/2$. Considering the classification results of all the binary-class SVMs, the predicted value with the highest frequency is the final classification result.

For a given training set $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, where $y_i \in \{-1, +1\}$, the binary-class SVM needs to find a hyperplane (i.e., $w^T x + b = 0$) to make the best segmentation of the sample points. Assuming that the hyperplane correctly classifies the training samples, for $(x_i, y_i) \in D$, there is

$$\begin{cases} w^T x + b > 0, y_i = +1 \\ w^T x + b < 0, y_i = -1 \end{cases} \quad (11)$$

Let

$$\begin{cases} w^T x + b \geq +1, y_i = +1 \\ w^T x + b \leq -1, y_i = -1 \end{cases} \quad (12)$$

the sum of the distances from the two heterogeneous support vectors to the hyperplane is

$$\gamma = \frac{2}{\|w\|} \quad (13)$$

where the support vector refers to the sample point closest to the hyperplane that makes Formula (12) true. The distance γ in Formula (13) is also called margin. Therefore, the goal of SVM is to find the partitioning hyperplane with the maximum margin, that is, to find the w and b that maximize γ and satisfy the constraints in Formula (12). Since maximizing γ only needs to maximize $\|w\|^{-1}$, which is equivalent to minimizing $\|w\|^2$, that is, satisfying

$$\begin{aligned} & \operatorname{argmin}_{w,b} \frac{1}{2} \|w\|^2 \\ & \text{s.t. } y_i (w^T x + b) \geq 1 \end{aligned} \quad (14)$$

This equivalent transformation makes solving Formula (14) a convex optimization problem.

Obviously, SVM solves the classification problem in a linear manner. For linearly inseparable data, the training sample can be linearly separable in the transformed eigenspace by introducing a kernel function. The Gaussian Kernel Function (also known as Radial Basis Function, RBF) is one of the commonly used kernel functions:

$$\kappa(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \quad (15)$$

where σ is the bandwidth of the Gaussian kernel (width).

3. Datasets and Model Validation

3.1. Datasets

In order to verify the validity of the proposed model, three publicly available hyperspectral remote sensing datasets were selected for experimental testing. These datasets have different scales, spatial resolutions, and spectral resolutions, and are obtained by different hyperspectral sensors. The robustness of the model can be examined by selecting such three datasets for the validation experiment, and these datasets are also commonly used in relevant studies. Among these three datasets, the Indian Pines dataset is available from the Purdue University Research Repository [52], while the Salinas and PaviaU datasets are available from the Computational Intelligence Group from the University of the Basque Country (http://www.ehu.eus/ccwintco/index.php?Title=Hyperspectral_Remote_Sensing_Scenes). In the three Sections 3.1.1–3.1.3, we will introduce these datasets in turn, mainly referring to the content provided by the Computational Intelligence Group. We will also introduce the selection rules of training samples for each dataset.

3.1.1. Indian Pines Dataset

The Indian Pines dataset was collected by the AVIRIS sensor at the test site in northwestern Indiana, USA. The dataset contains 145×145 pixels with a spatial resolution of 20 m/pixel. Since the dataset was collected by an AVIRIS sensor, it has 224 bands of 10 nm width with central wavelengths from 400 to 2500 nm. The original dataset contains 224 bands, which are reduced to 200 bands after excluding 24 water-absorbing bands: (104–108, 150–163, and 220). In the Indian Pines dataset, there are a total of 10,249 pixels with non-zero labels, which belong to 16 categories.

Table 1 shows the number of samples in each category in the Indian Pines dataset, as well as the division of training set and test set. It can be seen that the number of samples in different categories in the Indian Pines dataset is quite different. Some categories, such as Corn-notill, Soybean-mintill, and Woods, etc., have a large number of samples, while some categories, such as Alfalfa, Grass-pasture-mowed, and Oats, etc., have a relatively small number of samples. The uneven distribution of samples brings greater challenges to the classification task. In order to examine the performance of the models more comprehensively, it is necessary to use such data for detection. In the Indian Pines dataset, 20 samples of categories with a sample number greater than 40 are randomly selected to be added into the training set; for categories with a sample number less than 40, half of them are taken to be added into the training set, and the rest samples are all used to form the test set.

Table 1. The number of samples in each category in the Indian Pines dataset.

#	Class	Total Samples	Training Set	Test Set
1	Alfalfa	46	20	26
2	Corn-notill	1428	20	1408
3	Corn-mintill	830	20	810
4	Corn	237	20	217
5	Grass-pasture	483	20	463
6	Grass-trees	730	20	710
7	Grass-pasture-mowed	28	14	14
8	Hay-windrowed	478	20	458
9	Oats	20	10	10
10	Soybean-notill	972	20	952
11	Soybean-mintill	2455	20	2435
12	Soybean-clean	593	20	573
13	Wheat	205	20	185
14	Woods	1265	20	1245
15	Buildings-Grass-Trees-Drives	386	20	366
16	Stone-Steel-Towers	93	20	73
Total		10,249	304	9945

3.1.2. Salinas Dataset

The Salinas dataset is also collected by the AVIRIS sensor as the Indian Pines dataset, and the imaged scene is the Salinas Valley in California, USA. Therefore, the Salinas dataset also contains 224 bands of 10 nm width with central wavelengths from 400 to 2500 nm, and it has 204 bands remaining after excluding 20 water absorption bands: (108–112, 154–167, and 224). The dataset contains 512×217 pixels with a spatial resolution of 3.7 m/pixel. In the Salinas dataset, there are a total of 54,129 pixels with non-zero labels, which belong to 16 categories.

Table 2 shows the number of samples in each category in the Salinas dataset, as well as the division of training set and test set. Although the Salinas dataset also has some categories with a large number of samples and some categories with a relatively few samples, all categories can select 20 samples to form the training set. Therefore, the uneven sample distribution of the Salinas dataset does not actually have much impact on the classification effect of the model.

3.1.3. PaviaU Dataset

The PaviaU dataset was collected by the ROSIS sensor over several areas of the University of Pavia. The dataset contains 610×610 pixels, and its spatial resolution is 1.3 m/pixel, which is the highest among the three datasets used for testing in this paper. There is a large contiguous area in this dataset that does not contain usable information, so the actual available dataset only contains 610×340 pixels. The PaviaU dataset contains 103 bands of 6 nm width with central wavelengths from 430 to 860 nm. In this dataset, there are a total of 54,129 pixels with non-zero labels, which belong to 16 categories.

Table 3 shows the number of samples in each category in the PaviaU dataset, as well as the division of training set and test set. In the PaviaU dataset, 20 samples of each category were randomly selected to form the training set, and the remaining samples formed the test set.

Table 2. The number of samples in each category in the Salinas dataset.

#	Class	Total Samples	Training Set	Test Set
1	Brocoli green weeds 1	2009	20	1989
2	Brocoli green weeds 2	3726	20	3706
3	Fallow	1976	20	1956
4	Fallow rough plow	1394	20	1374
5	Fallow smooth	2678	20	2658
6	Stubble	3959	20	3939
7	Celery	3579	20	3559
8	Grapes untrained	11,271	20	11,251
9	Soil vineyard develop	6203	20	6183
10	Corn senesced green weeds	3278	20	3258
11	Lettuce romaine 4 wk	1068	20	1048
12	Lettuce romaine 5 wk	1927	20	1907
13	Lettuce romaine 6 wk	916	20	896
14	Lettuce romaine 7 wk	1070	20	1050
15	Vinyard untrained	7268	20	7248
16	Vinyard vertical trellis	1807	20	1787
Total		54,129	320	53,809

Table 3. The number of samples in each category in the PaviaU dataset.

#	Class	Total Samples	Training Set	Test Set
1	Asphalt	6631	20	6611
2	Meadows	18,649	20	18,629
3	Gravel	2099	20	2079
4	Trees	3064	20	3044
5	Painted metal sheets	1345	20	1325
6	Bare Soil	5029	20	5009
7	Bitumen	1330	20	1310
8	Self-Blocking Bricks	3682	20	3662
9	Shadows	947	20	927
Total		42,776	180	42,596

3.2. Model Validation

Since the proposed model can be regarded as an improvement based on the pixel-wise SVM classification model, the simple RBF-kernel SVM was used for comparison. Since the addition of PCA in NSW-PCA-SVM is one of the important factors that affect the final classification effect, the PCA-SVM model was also used for comparison. In order to verify the necessity of PCA, the NSW-SVM model without PCA was also used for comparison. In addition, six state-of-the-art models were selected for comparative analysis, i.e., CDCT-WF-SVM [28], CDCT-2DCT-SVM [28], SDWT-2DWT-SVM [53], SDWT-WF-SVM [53], SDWT-2DCT-SVM [53] and Two-Stage [54]. The six advanced models used for comparison come from three published papers. In these papers, the datasets used in the experiment are the same as those used in our paper. The difference is that these papers use far more samples for model training than our paper does. Among these six models, the first five models belong to the “pre-processing before classification” model, and the sixth model belongs to the “processing after classification” model.

(1) CDCT-WF-SVM: This model uses a one-dimensional Discrete Cosine Transform (1D-DCT) to obtain the spectral profile representation of the HSI, and for the high-frequency components of the representation, a two-dimensional adaptive Wiener filter (2D-AWF) is used to extract spatial information. After combining the low-frequency components with the processed high-frequency components, a one-dimensional inverse DCT is used for transformation to obtain the final pre-processed data. Finally, SVM is used for classification.

(2) CDCT-2DCT-SVM: The structure of this model is similar to CDCT-WF-SVM, except that the 2D-AWF is replaced with a 2D-DCT.

(3) SDWT-2DWT-SVM: This model uses Discrete Wavelet Transform (DWT) to decompose HSI in multiple stages in the spectral dimension, and uses 2D-DWT to extract spatial information for each stage. Then, Perform DWT reconstruction on the combined data to obtain the final pre-processed data. Finally, SVM is used for classification.

(4) SDWT-WF-SVM: The structure of this model is similar to SDWT-2DWT-SVM, except that the 2D-DWT is replaced with a 2D-WF.

(5) SDWT-2DCT-SVM: The structure of this model is similar to SDWT-2DWT-SVM, except that the 2D-DWT is replaced with a 2D-DCT.

(6) Two-Stage: This model contains two processing stages. The first stage is to use SVM to obtain a pixel-wise classification probability map, and the second stage is to use a variational denoising method to reconstruct the classification map.

The three evaluation indicators used in this paper to measure the classification performance of the models are Overall Accuracy (OA), Average Accuracy (AA) and Kappa coefficient (κ) [55]. OA reflects the proportion of correctly classified samples in the test sample. AA is obtained by calculating the classification accuracy of each category and taking the arithmetic average. When there is a large gap in the number of different samples in the dataset, the OA of the classification is easily affected by the category with a large sample size. Therefore, it is difficult to reflect the real level of the model performance using only OA. Kappa coefficient can reflect the comprehensive performance of classification model on OA and AA.

In the experimental verification, this paper repeated all the experiments for 10 times and took the average values as the final experimental results to reduce the influence of random factors. Three evaluation indicators (i.e., OA, AA and κ) can be calculated by the confusion matrix [56], whose form is as

$$C = \begin{bmatrix} c_{11} & \cdots & c_{1k} \\ \vdots & \ddots & \vdots \\ c_{k1} & \cdots & c_{kk} \end{bmatrix}, \quad (16)$$

where c_{ij} represents the number of samples whose real label is the i -th class and is classified as the j -th class, $i, j \in \{1, 2, \dots, k\}$.

The formula for calculating OA according to the confusion matrix is as follows:

$$OA = \frac{\text{tr}(C)}{\sum_{i=1}^k \sum_{j=1}^k c_{ij}} \quad (17)$$

The formula for calculating AA according to the confusion matrix is as follows:

$$AA = \frac{1}{k} \sum_{i=1}^k \frac{c_{ii}}{\sum_{j=1}^k c_{ij}} \quad (18)$$

The formula for calculating κ according to the confusion matrix is as follows:

$$\kappa = \frac{\sum_{i=1}^k \sum_{j=1}^k c_{ij} \times \sum_{i=1}^k c_{ii} - \sum_{l=1}^k \left(\sum_i c_{il} \times \sum_j c_{lj} \right)}{\left(\sum_{i=1}^k \sum_{j=1}^k c_{ij} \right)^2 - \sum_{l=1}^k \sum_i c_{il} \times \sum_j c_{lj}} \quad (19)$$

Therefore, the average OA, AA and κ can be calculated according to Formulas (17)–(19) by calculating the mean value of the confounding matrices obtained from 20 experiments.

The experiments were performed on a computer with an Intel Core i7-8750H CPU, 8 GB RAM, and all the methods were implemented with Python 3.

4. Experimental Results and Analysis

In this section, the classification performance of the proposed model was tested and analyzed experimentally on three public datasets. Firstly, the values of the two main parameters in the NSW-PCA-SVM model (i.e., neighbor size and principal component number) were measured. Secondly, the best classification results of NSW-PCA-SVM on three datasets are given, and compared with the best results of two basic pixel-wise classification models and six state-of-the-art models based on filtering. Finally, the classification results of NSW-PCA-SVM and various comparison models under different training set sample sizes are presented.

4.1. Model Validation

There are two key parameters, namely neighborhood size and principal component number, which have a significant impact on the performance of the NSW-PCA-SVM model. The value of the neighborhood size (denoted as ω) determines the value of a in Formula (2), i.e., $a = (\omega - 1)/2$. According to the description of the NSW algorithm in Section 2.1, the value of ω directly affects the number of related pixels during reconstruction. When the value of ω is large, the reconstructed pixels are more affected by the related pixels. Conversely, when the value of ω is small, the reconstructed pixels are less affected by the related pixels.

Since each dataset has a different spatial resolution, it is necessary to conduct different experiments for different datasets when determining the value of ω , where the value range of ω is $\{2n + 1 \mid n = 1, 2, \dots, 24\}$. For the NSW-SVM model, the OAs corresponding to different values of ω were obtained, and the optimal value of ω corresponded to the largest OA. Since the result of NSW-PCA-SVM is determined by two parameters, when determining the optimal value of ω , it is necessary to consider the optimal values of the principal component number corresponding to different ω .

Figure 3 shows the OAs of NSW-SVM and NSW-PCA-SVM corresponding to different values of ω on the Indian Pines, Salinas and PaviaU datasets. As shown in Figure 3a, on the Indian Pines dataset, when $\omega = 21$, both NSW-SVM and NSW-PCA-SVM achieved the highest OA. As shown in Figure 3b, on the Salinas dataset, the highest OA of NSW-SVM was obtained when $\omega = 31$, and the highest OA of NSW-PCA-SVM was obtained when $\omega = 33$. As shown in Figure 3c, on the PaviaU dataset, the highest OA of NSW-SVM was obtained when $\omega = 13$, and the highest OA of NSW-PCA-SVM was obtained when $\omega = 11$.

According to the trend of each curve in Figure 3, the following three conclusions can be drawn:

(1) All the curves basically showed a trend of increasing first and then decreasing, which is particularly obvious in Figure 3a,b. In the NSW method, the value of ω determines the number of related pixels in the neighborhood. When the value of ω is larger, the neighborhood contains relatively more related pixels, including homogeneous pixels and heterogeneous pixels. Therefore, although a larger value of ω makes the reconstructed pixels contain richer useful information, which is provided by homogeneous pixels, it also makes the reconstructed pixels contain more interference information, which is provided by heterogeneous pixels. On the contrary, a smaller value of ω makes the reconstructed pixels contain less interference information, but at the same time it makes the reconstructed pixels contain less useful information. Therefore, whether the value of ω is too large or too small will cause the classification accuracy of the NSW-PCA-SVM model to decrease.

(2) When the dataset is unchanged, the optimal values of ω in NSW-SVM and NSW-PCA-SVM are close. Therefore, the optimal value of ω in NSW-PCA-SVM can be roughly determined according to the optimal value of ω in NSW-SVM, so as to reduce the experimental testing required for parameter adjustment.

(3) When NSW-SVM and NSW-PCA-SVM achieved the best classification effect, the values of ω on the Salinas dataset were both greater than those on the Indian Pines dataset. The reason is that the spatial resolution of the Salinas dataset is larger than that of the Indian Pines dataset. When other conditions are the same or similar, the larger the

spatial resolution of HSI is, the more homogeneous pixels that can be contained in the neighborhood.

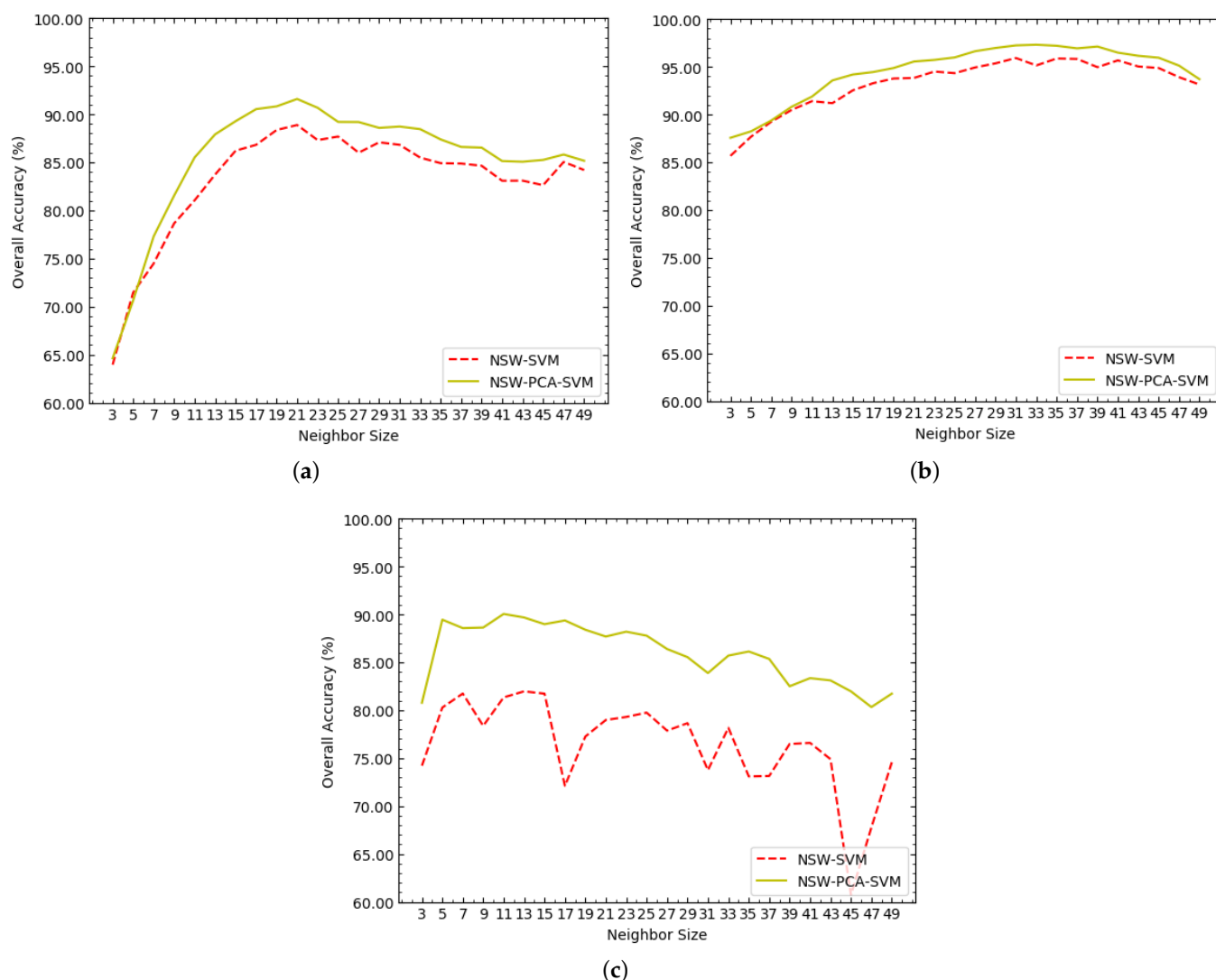


Figure 3. The overall accuracies (OAs) of NSW-SVM and NSW-PCA-SVM corresponding to different values of ω on the three datasets, where (a–c) represent the experimental results in Indian Pines, Salinas and PaviaU datasets, respectively.

For any non-adaptive dimensionality reduction algorithm, the dimension of the data after dimensionality reduction needs to be determined in advance. The principal component number of PCA is the dimension of the dataset after dimensionality reduction, denoted as c . When performing experimental determination on the value of c , it is also necessary to consider the optimal value of ω corresponding to different values of c .

It can be seen from Figure 4 that the three curves generally show a trend of first increasing and then decreasing. Therefore, the possible range of the optimal value of c can be investigated in a sparse value interval, such as $[1, 10, 20, \dots]$. According to Figure 4, the optimal value of c in NSW-PCA-SVM on the three datasets can be determined: on the Indian Pines dataset, $c = 16$; on the Salinas dataset, $c = 12$; on the PaviaU dataset, $c = 15$.

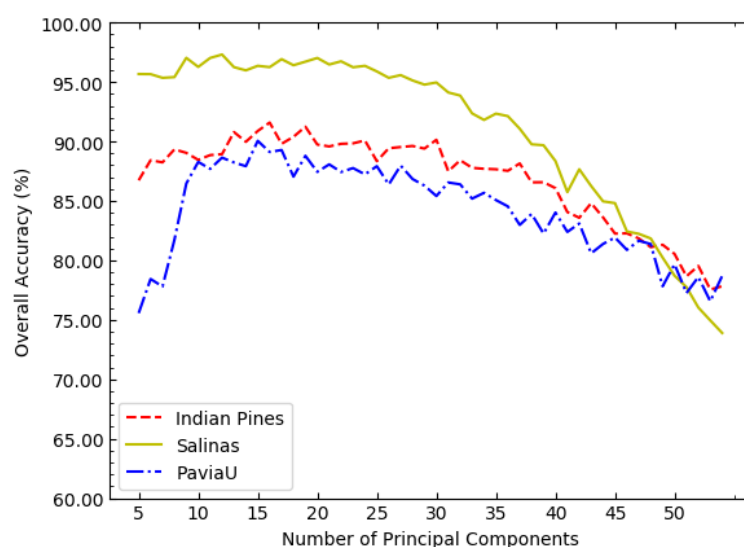


Figure 4. The OAs of NSW-PCA-SVM corresponding to different values of c on the three datasets.

According to the experimental results, the parameter settings of NSW-SVM and NSW-PCA-SVM on the three datasets are obtained, as shown in Table 4.

Table 4. Parameter settings of NSW-SVM and NSW-PCA-SVM on three datasets.

Model	Parameter	Indian Pines	Salinas	PaviaU
NSW-SVM	ω	21	31	13
NSW-PCA-SVM	ω	21	33	11
	c	16	12	15

The experimental results of the five state-of-the-art comparison models are obtained according to the codes provided by the authors of the original papers, so the parameters of each comparison model are set according to the original papers. For the basic SVM model and PCA-SVM model, the best parameters were determined based on experiments.

SVM: The kernel function is RBF, the kernel coefficient is 0.125, and the regularization parameter is 200. The above parameter settings are applicable to the experiments on the three datasets, and are applicable to the SVM part of all models in this paper.

PCA-SVM: The principal component number of PCA is 6 on the Indian Pines dataset, 4 on the Salinas dataset, and 11 on the PaviaU dataset.

4.2. Classification Results on the Indian Pines Dataset

In Section 4.2, the classification performances of NSW-SVM and NSW-PCA-SVM were tested on the Indian Pines dataset. Table 5 shows the specific classification results of each model, including the classification accuracy of each class, overall classification accuracy (OA), average classification accuracy (AA) and Kappa coefficient (κ), where the sample size of the training set is shown in Table 1.

Table 5. Specific classification results of each model on the Indian Pines dataset, where Model-1, Model-2, ..., Model-10 represents SVM, PCA-SVM, CDCT-WF-SVM, CDCT-2DCT-SVM, SDWT-2DWT-SVM, SDWT-WF-SVM, SDWT-2DCT-SVM, Two-Stage, NSW-SVM and NSW-PCA-SVM, respectively, and the same in the other figures and tables.

Class	Model-1	Model-2	Model-3	Model-4	Model-5	Model-6	Model-7	Model-8	Model-9	Model-10
1	12.35	18.93	97.31	97.69	95.38	98.08	95.00	100	83.33	69.27
2	39.37	48.30	65.44	79.95	69.81	73.69	64.39	76.59	71.66	87.26
3	45.73	40.51	69.59	82.95	71.17	75.90	70.60	87.56	82.19	90.40
4	16.40	32.69	89.86	87.05	85.85	82.63	80.83	99.95	75.10	74.48
5	74.20	65.94	90.52	92.18	93.02	89.94	90.04	93.17	96.48	97.82
6	88.33	81.21	96.44	92.56	94.30	96.14	97.96	98.73	99.36	99.97
7	77.69	24.61	98.75	95.00	88.75	96.25	100	100	45.60	42.94
8	65.83	95.49	97.12	99.45	97.75	96.97	95.50	100	100	99.61
9	40.74	25.00	100	99.00	96.00	100	100	100	36.23	38.91
10	53.91	53.58	71.70	76.96	69.71	75.67	71.74	87.27	82.17	84.48
11	70.15	69.88	54.21	72.10	63.00	69.44	59.75	84.60	94.03	94.28
12	21.88	32.03	73.63	75.13	71.10	68.62	56.70	93.73	72.79	75.69
13	89.87	78.45	99.03	97.41	98.97	98.32	96.59	100	99.62	99.13
14	90.35	90.08	92.94	96.55	92.47	88.30	92.84	93.03	99.09	99.70
15	35.42	32.91	80.74	91.45	81.69	85.77	80.46	99.59	95.21	95.42
16	91.43	68.31	95.48	96.30	97.12	96.03	97.95	100	94.94	94.59
OA	53.29	58.44	74.47	83.26	76.91	79.36	74.55	89.02	87.35	91.40
AA	57.10	53.62	85.80	89.48	85.38	86.98	84.40	94.64	82.99	84.00
κ	47.94	53.45	71.25	81.07	73.94	76.69	71.29	87.55	85.61	90.20

According to Table 5, on the Indian Pines dataset, NSW-PCA-SVM achieved the best classification result ($\kappa = 90.20\%$). The Kappa coefficient in the classification results of each model is mainly determined by OA, which is the reason why NSW-PCA-SVM (OA = 91.40%) achieved a high Kappa value. However, the classification result of the proposed model performed worse on AA than the six state-of-the-art comparison models. By analyzing the classification results of various classes of samples, it can be seen that NSW-PCA-SVM achieved low classification accuracy in the classification of samples of category 1, 7 and 9. According to Table 1, the Indian Pines dataset has a small number of samples in categories 1, 7 and 9, which makes it difficult for NSW method to obtain effective spatial information and results in poor final classification results.

By comparing the classification results with the basic SVM model and the PCA-SVM model, it can be seen that 8 models utilizing the spatial information of HSIs were significantly improve the classification accuracy. Figure 5 shows the classification maps of each model on the Indian Pines dataset. According to Figure 5, the classification maps of the two pixel-based classification models (SVM and PCA-SVM) contained a large amount of classification noise, while the classification maps of the models utilizing the spatial information contained much less classification noise. In addition, by observing the classification maps of NSW-SVM and NSW-PCA-SVM, it can be seen that the classification maps of the two models contained less noise for the categories with more samples. On the contrary, they contain more noise for the categories with less samples.

In order to investigate the impact of the size of the training set on the classification accuracy of the models, this paper tested the OAs of each model under different sizes of the training set, as shown in Figure 6. According to Figure 6, with the increase of the sample size of the training set, the increase amplitudes of the OAs of NSW-SVM and NSW-PCA-SVM were smaller than that of other comparison models. With the increase of the sample size of the training set, the OAs of the six state-of-the-art classification models were getting closer to NSW-PCA-SVM. In particular, when the number of samples in the training set exceeds 60 samples/class, the overall classification accuracy of NSW-PCA-SVM is worse than that of the Two-Stage model. Figure 6 can show the advantage of NSW-PCA-SVM under a small sample size of the training set.

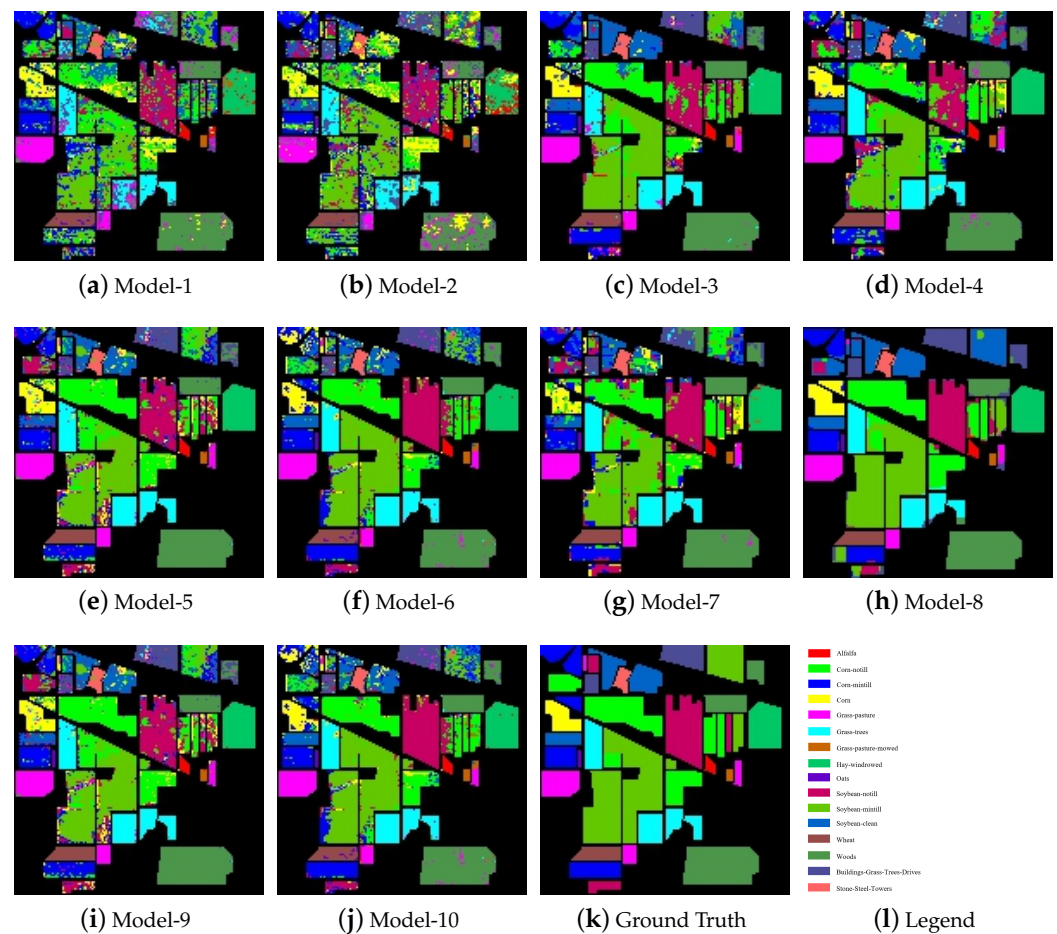


Figure 5. Classification maps of the models on the Indian Pine dataset.

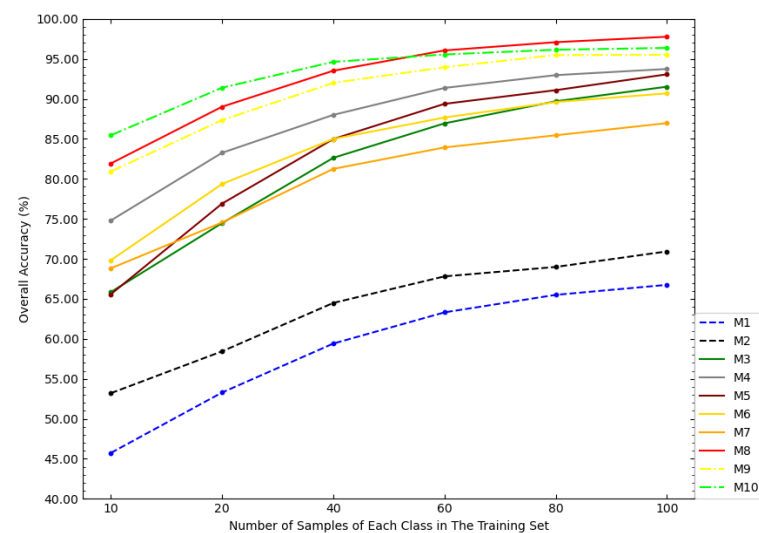


Figure 6. The OAs of each model in the Indian Pine dataset under the partitioning of training sets of different sizes.

4.3. Classification Results on the Salinas Dataset

In Section 4.3, the classification performances of NSW-SVM and NSW-PCA-SVM were tested on the Salinas dataset. Table 6 shows the specific classification results of each model, including classification accuracy of each category, overall classification accuracy (OA),

average classification accuracy (AA) and Kappa coefficient (κ), where the sample size of the training set is shown in Table 2.

Table 6. Specific classification results of each model on the Salinas dataset.

Class	Model-1	Model-2	Model-3	Model-4	Model-5	Model-6	Model-7	Model-8	Model-9	Model-10
1	98.87	97.83	98.50	98.64	97.97	98.52	98.54	100	100	100
2	99.27	97.78	98.89	96.07	98.98	98.53	99.02	100	99.30	100
3	91.07	86.83	95.74	94.99	98.02	94.97	97.84	99.92	99.15	98.06
4	97.61	96.95	99.24	99.38	99.42	99.49	98.68	99.83	93.12	95.97
5	98.83	97.69	96.24	95.28	96.31	95.36	94.82	98.56	99.75	99.70
6	100	99.90	98.87	98.73	99.38	99.41	99.37	99.99	99.97	99.97
7	98.95	97.32	99.11	99.12	99.40	99.26	99.52	100	99.91	99.89
8	71.26	74.50	62.65	86.08	84.60	78.26	76.43	85.27	92.71	98.26
9	99.30	99.35	97.93	98.71	97.67	98.99	98.17	99.99	99.18	99.53
10	77.36	81.86	82.98	88.72	89.94	90.25	89.89	96.25	92.59	93.93
11	85.83	80.07	96.62	94.80	96.80	96.44	92.92	99.56	92.98	97.05
12	94.38	96.56	99.79	99.64	99.62	99.23	99.59	100	99.47	99.44
13	96.41	94.96	98.29	96.10	98.23	97.02	97.88	99.29	95.07	94.00
14	90.83	87.63	93.92	93.23	94.14	92.66	93.76	98.50	86.55	84.38
15	53.16	57.79	71.78	85.99	84.05	81.99	82.27	83.92	84.13	91.25
16	86.90	93.16	94.97	97.42	97.08	96.72	97.94	98.74	98.83	99.36
OA	84.08	85.51	86.10	93.07	92.94	91.25	90.91	94.36	94.89	97.24
AA	90.00	90.01	92.85	95.18	95.73	94.82	94.79	97.49	95.80	96.92
κ	82.33	83.89	84.57	92.29	92.15	90.28	89.91	93.72	94.32	96.93

According to Table 6, on the Salinas dataset, NSW-PCA-SVM achieved the best classification result ($\kappa = 96.93\%$), while NSW-SVM achieved the sub-optimal result ($\kappa = 94.32\%$). Different from the Indian Pines dataset, the AAs of the two proposed models on the Salinas dataset were better than that of the comparison models, except for the Two-Stage model. According to Table 2, although the number of samples in each category of the Salinas dataset were not similar, there is no case where the number of samples in several categories were particularly small.

Figure 7 shows the classification maps of each model on the Salinas dataset. According to Figure 6, the classification noises contained in the classification maps of NSW-SVM and NSW-PCA-SVM were far less than other classification models, especially the two basic models. According to Figure 7k, it can be seen that the distribution of similar samples was very concentrated, which enables the NSW method to effectively extract the spatial information of the Salinas dataset to improve the classification effect.

Figure 8 shows the OAs of each model on the Salinas dataset under different sample sizes of the training set. According to Figure 8, when the sample size of the training set was small, the OA of NSW-PCA-SVM was much higher than that of the comparison models. However, as the sample size of the training set increased, the advantage of the proposed model over the comparison models was gradually weakening, especially when the sample size of the training set is 100 samples/class, the OA of SDWT-2DCT-SVM was very close to NSW-PCA-SVM and the OA of Two-Stage was higher than NSW-PCA-SVM. Figure 8 also reflected the superiority of NSW-PCA-SVM under a small sample size of the training set.

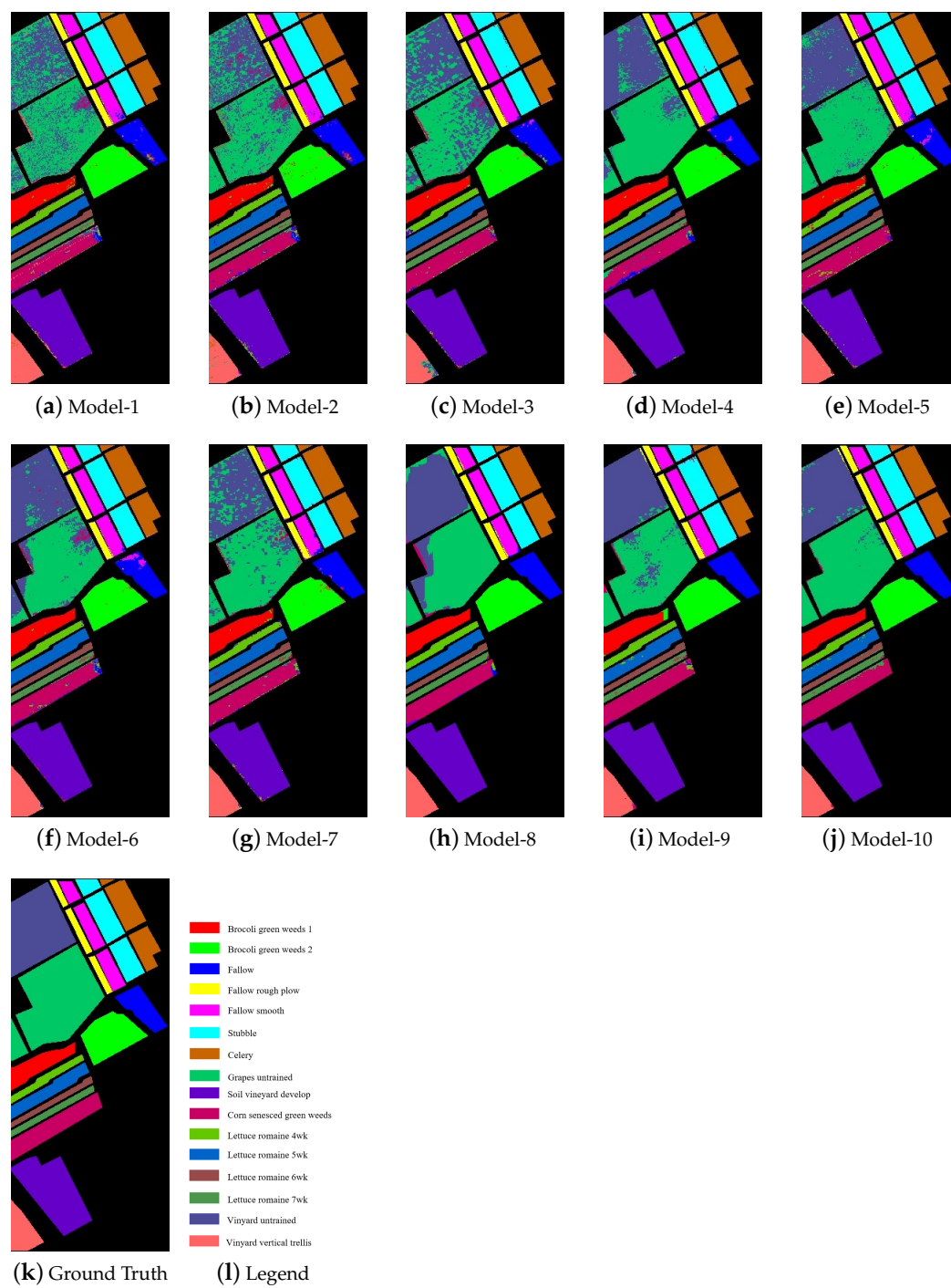


Figure 7. Classification maps of the models on the Salinas dataset.

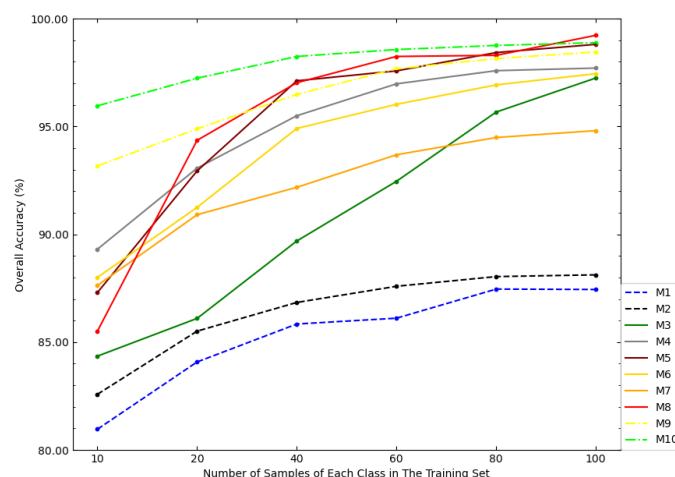


Figure 8. The OAs of each model in the Salinas dataset under the partitioning of training sets of different sizes.

4.4. Classification Results on the PaviaU Dataset

In Section 4.4, the classification performances of NSW-SVM and NSW-PCA-SVM were tested on the PaviaU dataset. Table 7 shows the specific classification results of each model, including the classification accuracy of each category, overall classification accuracy (OA), average classification accuracy (AA) and Kappa coefficient (κ), where the sample size of the training set is shown in Table 3.

Table 7. Specific classification results of each model on the PaviaU dataset.

Class	Model-1	Model-2	Model-3	Model-4	Model-5	Model-6	Model-7	Model-8	Model-9	Model-10
1	92.98	86.77	75.80	80.07	80.79	74.60	82.10	87.86	80.05	92.13
2	87.20	90.42	82.82	87.33	77.30	81.93	88.07	81.64	94.32	97.15
3	50.72	48.46	81.69	84.02	79.75	75.47	76.93	93.92	73.85	83.16
4	62.82	62.84	88.89	90.96	91.82	91.28	89.33	88.68	79.34	75.15
5	78.18	97.49	99.20	99.28	99.29	99.18	99.51	99.95	97.69	99.68
6	41.66	47.89	80.47	87.19	81.82	79.24	91.43	97.62	64.53	85.46
7	48.58	40.54	80.91	92.12	91.08	88.96	86.79	99.52	56.93	81.65
8	80.02	67.42	73.16	74.71	76.55	77.79	72.31	97.26	85.17	77.57
9	100	99.60	99.92	99.87	99.89	99.90	99.20	93.10	77.17	92.98
OA	71.41	72.16	81.83	85.99	81.07	81.62	86.29	88.30	82.38	89.81
AA	71.35	71.27	84.76	88.39	86.48	85.37	87.30	93.28	78.78	87.21
κ	64.00	64.81	76.56	81.82	75.80	76.33	82.17	85.04	77.25	86.70

According to Table 7, on the PaviaU dataset, NSW-PCA-SVM also achieved the best classification results ($\kappa = 86.70\%$). In addition, although NSW-PCA-SVM achieved the best OA, it was slightly lower than Two-Stage on AA. Figure 9 shows the classification maps of each model on the PaviaU dataset. According to Figure 9, the classification map of NSW-PCA-SVM contained less noise, which corresponds to the classification results in Table 7. According to Figure 9k, it can be seen that except for a few categories, the sample distribution of the same categories was scattered. Therefore, it difficult for the NSW method to effectively extract the spatial information of the PaviaU dataset.

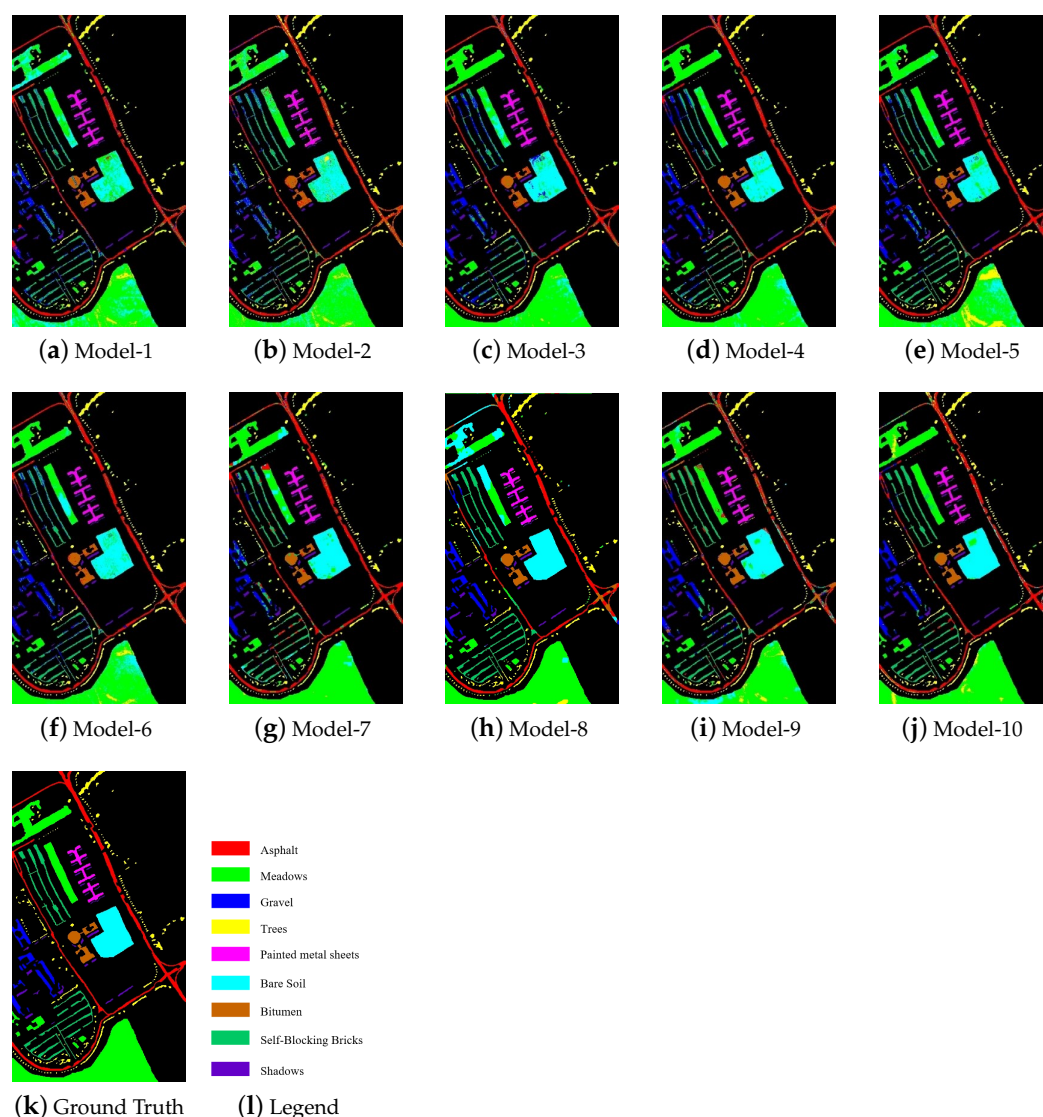


Figure 9. Classification maps of the models on the PaviaU dataset.

Figure 10 shows OAs of each model on the PaviaU dataset under different sample sizes of the training set. It can be seen from Figure 10 that the classification effect of NSW-PCA-SVM compared to the six state-of-the-art models achieved greater advantages when the sample size of the training set was small. However, as the sample size of the training set increased, the advantage was gradually weakening. The result was similar to the those on the first two datasets. For example, when the sample size of the training set exceeds 60 samples/class, the OA of the Two-Stage model exceeded that of NSW-PCA-SVM.

Table 8 shows the running time of the proposed NSW-SVM, NSW-PCA-SVM and the eight models used for comparison. It can be seen that the running time of the original SVM model or the PCA-SVM model is very short. The reason is that the SVM algorithm and PCA algorithm called in the program are actually developed based on the C Programming Language. Compared with the comparison models, the two models based on NSW require more computational time. The main computational cost lies in the calculation of correlation coefficient matrices according to formula (6). In fact, after the neighbourhoods are divided using formula (2), the reconstruction of each target pixel is independent of each other. Therefore, parallel computing techniques can be used to improve the calculation speed of the NSW algorithm. If the GPU is used for running acceleration, the running time of the NSW algorithm can be greatly reduced.

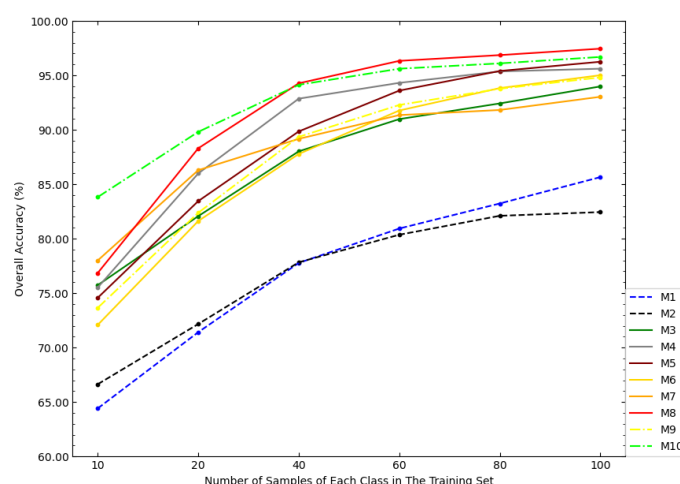


Figure 10. The OAs of each model in the PaviaU dataset under the partitioning of training sets of different sizes.

Table 8. Running time (seconds) for the classification for Indian Pines, Salinas, PaviaU datasets by various models.

	Model-1	Model-2	Model-3	Model-4	Model-5	Model-6	Model-7	Model-8	Model-9	Model-10
Indian Pines	1.13	0.62	6.76	7.09	27.94	20.25	25.56	3.17	113.43	112.58
Salinas	5.05	2.46	34.98	37.52	145.38	104.06	133.63	13.19	725.35	704.77
PaviaU	1.62	1.25	59.68	60.92	282.33	193.86	238.74	11.18	148.99	113.51

5. Discussion

In the experimental part, we conducted experimental tests on the proposed NSW-PCA-SVM model and the models for comparison on three datasets. The performance of all models on the three datasets is relatively consistent, so the following analysis is based on the experimental results on the Indian Pines dataset. By analysing the experimental results, the following conclusions can be drawn:

(1) The two basic models, SVM and PCA-SVM, only consider the spectral information of the HSI, their classification accuracy is worse than other models that additionally consider spatial information. The classification accuracy of the NSW-SVM model (OA = 87.35%) is much higher than that of the SVM model (OA = 53.29%), and the classification accuracy of the NSW-PCA-SVM model (OA = 91.40%) is also much higher than that of the PCA-SVM model (OA = 58.44%), which shows that the NSW method can effectively extract the spatial information of HSI and significantly improve the classification performance of the models.

(2) The classification accuracy of the five comparison models, i.e., CDCT-WF-SVM, CDCT-2DCT-SVM, SDWT-2DWT-SVM, SDWT-WF-SVM and SDWT-2DCT-SVM, has been significantly improved compared with the SVM model and the PCA-SVM model. Among them, the CDCT-2DCT-SVM model (OA = 83.26%) achieved the highest classification accuracy. In fact, the CDCT-2DCT-SVM model and the other four models only perform spatial filtering on the noisy part from the spectral filter, while a part of the reconstructed data does not contain spatial information. However, the NSW method considers all spectral bands when extracting the spatial information of HSIs. Therefore, the classification results of the NSW-PCA-SVM model (OA = 91.40%, AA = 84.00%, Kappa = 90.20%) were better than those of the CDCT-2DCT-SVM model (OA = 83.26%, AA = 89.48%, Kappa = 81.07%) and the other four models.

(3) In our comparative experiment, we chose a “processing after classification” model, namely Two-Stage. Since the main classification stage of the Two-Stage model is based on pixel-wise classification, when the sample size of the training set is small, the OA of Two-Stage (OA = 89.02%) is lower than that of NSW-PCA-SVM (OA = 91.04%). The Two-Stage model uses a variational denoising method to restore the classification map, which allows the model to ignore the uneven sample distribution, so the AA of Two-Stage (AA = 94.64%)

is higher than the NSW-PCA-SVM model (AA = 84.00%). It can also be seen from Figure 5 that the classification map obtained by using the Two-Stage model contains less noise, but there are cases where all pixels in a certain area are classified incorrectly.

(4) When the number of samples in the training set is small, such as 20 samples/class, the advantage of the NSW-PCA-SVM model in OA can reach 2.38–38.11% over the comparison models. According to Figures 6, 8 and 10, further reducing the number of training set samples, such as 10 samples/class, would further expand the advantage of the NSW-PCA-SVM model. On the contrary, as the number of samples in the training set increases, the advantage of the NSW-PCA-SVM model decreases continuously. According to Table 8, the NSW-PCA-SVM model requires more computational time than other models, although it can be further optimized. Therefore, only when the number of samples in the training set is small, such as 10 samples/class, 20 samples /class or 40 samples/class, etc., the use of NSW-PCA-SVM for classification can achieve higher returns.

Another disadvantage of the NSW-PCA-SVM model is that when the distribution of homogeneous samples of the HSI in the spatial dimension is scattered, it is difficult for the NSW method to extract spatial information very effectively. For example, on the PaviaU dataset, the classification accuracy of NSW-PCA-SVM did not have a great advantage. According to Table 7, the advantage of the NSW-PCA-SVM model in OA only achieves 1.61–18.40% over the comparison models.

6. Conclusions

Based on the correlation between pixels, a Nested Sliding Window (NSW) method is proposed to extract the spatial features of HSIs. The NSW method can be used to reconstruct the pixels of HSIs, and the reconstructed pixels contain the information of the original pixels and the pixels that are in a spatially adjacent relationship with them. For the reconstructed data, PCA is used for dimensionality reduction to further eliminate the noise in the spectral dimension. Finally, the RBF-kernel SVM is used to classify the processed data. The NSW-PCA-SVM model has been tested experimentally on three public datasets. By comparing with the SVM model and the PCA-SVM model that only consider the spectral information of HSIs, the proposed model based on the NSW method can significantly improve the classification accuracy. Compared with five filter-based comparison approaches, NSW can extract spatial features for all spectral bands of HSIs. Compared with the “processing after classification” model, i.e., the Two-Stage model, the NSW-PCA-SVM model also has obvious advantages when the training set sample size is small. Therefore, our main contribution is to propose an effective classification model with limited training samples.

The limitations of NSW-PCA-SVM are: when the number of training set samples is large, it is difficult for NSW-PCA-SVM to obtain the advantages in classification accuracy, especially the NSW method requires more computational time; on datasets where homogeneous samples are closely adjacent in spatial relationship, such as Indian Pines and Salinas datasets, NSW-PCA-SVM has greater advantages, while for the datasets with dispersed spatial relations of homogeneous samples, such as PaviaU dataset, NSW-PCA-SVM are difficult to achieve significant advantages. For the improvement of the NSW method, we will carry out the following goals in future work: consider using more reasonable parameters to measure the degree of correlation between homogeneous samples; consider more reasonable reconstruction methods to avoid heterogeneous pixels participating in reconstruction; consider different neighborhood division methods, including the division with irregular shapes.

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

Funding: The work described in this paper was supported by the National Natural Science Foundation of China (U1711267), Hubei Province Innovation Group Project (2019CFA023), Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan) (CUGCJ1810), and the Open Fund of Hubei Key Laboratory of Intelligent Geo-Information Processing (KLIGIP2018 and ZRIGIP-201801).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: http://www.ehu.eus/ccwintco/index.php?Title=Hyperspectral_Remote_Sensing_Scenes.

Acknowledgments: We wish to thank the authors of the papers [28,53,54] for providing us with their experimental source code, which facilitates our comparison with their models in our experiments.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; and in the decision to publish the results.

References

- Wang, C.L.; Ren, J.; Wang, H.W.; Zhang, Y.; Wen, J. Spectral–spatial classification of hyperspectral data using spectral-domain local binary patterns. *Multimed. Tools Appl.* **2018**, *77*, 29889–29903. [CrossRef]
- Tiwari, K.C.; Arora, M.K.; Singh, D. An assessment of independent component analysis for detection of military targets from hyperspectral images. *Int. J. Appl. Earth Obs. Geoinf.* **2011**, *13*, 730–740. [CrossRef]
- Heiden, U.; Heldens, W.; Roessner, S.; Segl, K.; Esch, T.; Mueller, A. Urban structure type characterization using hyperspectral remote sensing and height information. *Landsc. Urban Plan.* **2012**, *105*, 361–375. [CrossRef]
- Im, J.; Jensen, J.R.; Jensen, R.R.; Gladden, J.; Waugh, J.; Serrato, M. Vegetation cover analysis of hazardous waste sites in Utah and Arizona using hyperspectral remote sensing. *Remote Sens.* **2012**, *4*, 327–353. [CrossRef]
- Akbari, H.; Kosugi, Y.; Kojima, K.; Tanaka, N. Detection and analysis of the intestinal ischemia using visible and invisible hyperspectral imaging. *IEEE Trans. Biomed. Eng.* **2010**, *57*, 2011–2017. [CrossRef]
- Wahabzada, M.; Mahlein, A.K.; Bauckhage, C.; Steiner, U.; Oerke, E.C.; Kersting, K. Plant phenotyping using probabilistic topic models: Uncovering the hyperspectral language of plants. *Sci. Rep.* **2016**, *6*, 1–11. [CrossRef]
- Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
- Bandos, T.V.; Bruzzone, L.; Camps-Valls, G. Classification of hyperspectral images with regularized linear discriminant analysis. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 862–873. [CrossRef]
- Belgiu, M.; Drăguț, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 24–31. [CrossRef]
- Ma, L.; Crawford, M.M.; Tian, J. Local manifold learning-based k-nearest-neighbor for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 4099–4109. [CrossRef]
- Shahdoosti, H.R.; Mirzapour, F. Spectral–spatial feature extraction using orthogonal linear discriminant analysis for classification of hyperspectral data. *Eur. J. Remote. Sens.* **2017**, *50*, 111–124. [CrossRef]
- Mei, S.; Ji, J.; Hou, J.; Li, X.; Du, Q. Learning sensor-specific spatial–spectral features of hyperspectral images via convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4520–4533. [CrossRef]
- Zhang, M.; Li, W.; Du, Q. Diverse region-based CNN for hyperspectral image classification. *IEEE Trans. Image Process.* **2018**, *27*, 2623–2634. [CrossRef]
- Zhao, C.; Liu, W.; Xu, Y.; Wen, J. A spectral–spatial SVM-based multi-layer learning algorithm for hyperspectral image classification. *Remote Sens. Lett.* **2018**, *9*, 218–227. [CrossRef]
- Dópido, I.; Li, J.; Marpu, P.R.; Plaza, A.; Dias, J.M.B.; Benediktsson, J.A. Semisupervised self-learning for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 4032–4044. [CrossRef]
- Wang, Y.; Duan, H. Classification of hyperspectral images by SVM using a composite kernel by employing spectral, spatial and hierarchical structure information. *Remote Sens.* **2018**, *10*, 441. [CrossRef]
- Liao, J.; Wang, L.; Zhao, G.; Hao, S. Hyperspectral image classification based on bilateral filter with linear spatial correlation information. *Int. J. Remote Sens.* **2019**, *40*, 6861–6883. [CrossRef]
- Benediktsson, J.A.; Palmason, J.A.; Sveinsson, J.R. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 480–491. [CrossRef]
- Liao, W.; Bellens, R.; Pizurica, A.; Philips, W.; Pi, Y. Classification of hyperspectral data over urban areas using directional morphological profiles and semi-supervised feature extraction. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2012**, *5*, 1177–1190. [CrossRef]

20. Hou, B.; Huang, T.; Jiao, L. Spectral–spatial classification of hyperspectral data using 3-D morphological profile. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2364–2368.
21. Imani, M.; Ghassemian, H. Edge patch image-based morphological profiles for classification of multispectral and hyperspectral data. *IET Image Process.* **2016**, *11*, 164–172. [[CrossRef](#)]
22. Kumar, B.; Dikshit, O. Hyperspectral image classification based on morphological profiles and decision fusion. *Int. J. Remote Sens.* **2017**, *38*, 5830–5854. [[CrossRef](#)]
23. Kumara, B.; Dikshit, O. Parallel Implementation of Morphological Profile Based Spectral–Spatial Classification Scheme for Hyperspectral Imagery. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *7*. [[CrossRef](#)]
24. Wu, K.; Jia, S. 2D Gabor-Based Sparse Representation Classification for Hyperspectral Imagery. In Proceedings of the 2018 Digital Image Computing: Techniques and Applications (DICTA), Canberra, Australia, 10–13 December 2018; pp. 1–8.
25. Jia, S.; Xie, H.; Deng, X. Extended morphological profile-based Gabor wavelets for hyperspectral image classification. In Proceedings of the 2018 24th International Conference on Pattern Recognition, Beijing, China, 20–24 August 2018; pp. 782–787.
26. Chen, Y.; Zhu, L.; Ghamisi, P.; Jia, X.; Li, G.; Tang, L. Hyperspectral images classification with Gabor filtering and convolutional neural network. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2355–2359. [[CrossRef](#)]
27. Hanbay, K. Hyperspectral image classification using convolutional neural network and two-dimensional complex Gabor transform. *J. Fac. Eng. Archit. Gazi Univ.* **2020**, *35*, 443–456.
28. Bazine, R.; Wu, H.; Boukhechba, K. Spatial Filtering in DCT Domain-Based Frameworks for Hyperspectral Imagery Classification. *Remote Sens.* **2019**, *11*, 1405. [[CrossRef](#)]
29. Liu, X.; Bourennane, S.; Fossati, C. Nonwhite noise reduction in hyperspectral images. *IEEE Geosci. Remote Sens. Lett.* **2011**, *9*, 368–372. [[CrossRef](#)]
30. Liu, X.; Feng, Y.; Li, Y.; Liu, X.; Zhao, W.; Fu, M. Denoising hyperspectral images with non-white noise based on tensor decomposition. In Proceedings of the 2016 IEEE International Conference on Digital Signal Processing, Beijing, China, 16–18 October 2016; pp. 437–441.
31. Nair, P.; Chaudhury, K.N. Fast high-dimensional bilateral and nonlocal means filtering. *IEEE Trans. Image Process.* **2018**, *28*, 1470–1481. [[CrossRef](#)]
32. Qian, Y.; Shen, Y.; Ye, M.; Wang, Q. 3-D nonlocal means filter with noise estimation for hyperspectral imagery denoising. In Proceedings of the 2012 IEEE International Geoscience and Remote Sensing Symposium, Munich, Germany, 22–27 July 2012; pp. 1345–1348.
33. Kang, X.; Xiang, X.; Li, S.; Benediktsson, J.A. PCA-based edge-preserving features for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 7140–7151. [[CrossRef](#)]
34. Tian, W.; Xu, L.; Chen, Z.; Shi, A. Multiple feature learning based on edge-preserving features for hyperspectral image classification. *IEEE Access* **2019**, *7*, 106861–106872. [[CrossRef](#)]
35. Tarabalka, Y.; Fauvel, M.; Chanussot, J.; Benediktsson, J.A. SVM-and MRF-based method for accurate classification of hyperspectral images. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 736–740. [[CrossRef](#)]
36. Yu, H.; Gao, L.; Li, J.; Li, S.S.; Zhang, B.; Benediktsson, J.A. Spectral–spatial hyperspectral image classification using subspace-based support vector machines and adaptive Markov random fields. *Remote Sens.* **2016**, *8*, 355. [[CrossRef](#)]
37. Qing, C.; Ruan, J.; Xu, X.; Ren, J.; Zabalza, J. Spatial–spectral classification of hyperspectral images: A deep learning framework with markov random fields based modelling. *IET Image Process.* **2018**, *13*, 235–245. [[CrossRef](#)]
38. Chakravarty, S.; Banerjee, M.; Chandel, S. Spectral–spatial classification of hyperspectral imagery using support vector and fuzzy-MRF. In Proceedings of the International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, Vancouver, BC, Canada, 25–27 October 2017; pp. 151–161.
39. Xu, Y.; Wu, Z.; Wei, Z. Markov random field with homogeneous areas priors for hyperspectral image classification. In Proceedings of the 2014 IEEE Geoscience and Remote Sensing Symposium, Quebec City, QC, Canada, 13–18 July 2014; pp. 3426–3429.
40. Tang, B.; Liu, Z.; Xiao, X.; Nie, M.; Chang, J.; Jiang, W.; Zheng, C. Spectral–spatial hyperspectral classification based on multi-center SAM and MRF. *Opt. Rev.* **2015**, *22*, 911–918. [[CrossRef](#)]
41. Cao, X.; Wang, X.; Wang, D.; Zhao, J.; Jiao, L. Spectral–Spatial Hyperspectral Image Classification Using Cascaded Markov Random Fields. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 4861–4872. [[CrossRef](#)]
42. Wang, Y.; Song, H.; Zhang, Y. Spectral–spatial classification of hyperspectral images using joint bilateral filter and graph cut based model. *Remote Sens.* **2016**, *8*, 748. [[CrossRef](#)]
43. Yu, H.; Gao, L.; Li, J.; Zhang, B. Spectral–spatial classification based on subspace support vector machine and Markov random field. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium, Beijing, China, 10–15 July 2016; pp. 2783–2786.
44. Liao, W.; Bellens, R.; Pižurica, A.; Philips, W.; Pi, Y. Classification of hyperspectral data over urban areas based on extended morphological profile with partial reconstruction. In Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems, Brno, Czech Republic, 4–7 September 2012; pp. 278–289.
45. Cao, X.; Xu, Z.; Meng, D. Spectral–spatial hyperspectral image classification via robust low-rank feature extraction and Markov random field. *Remote Sens.* **2019**, *11*, 1565. [[CrossRef](#)]
46. Cui, B.; Ma, X.; Zhao, F.; Wu, Y. A novel hyperspectral image classification approach based on multiresolution segmentation with a few labeled samples. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1729881417710219. [[CrossRef](#)]

47. Kang, X.; Li, S.; Benediktsson, J.A. Spectral–spatial hyperspectral image classification with edge-preserving filtering. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 2666–2677. [[CrossRef](#)]
48. Cui, B.; Ma, X.; Xie, X.; Ren, G.; Ma, Y. Classification of visible and infrared hyperspectral images based on image segmentation and edge-preserving filtering. *Infrared Phys. Technol.* **2017**, *81*, 79–88. [[CrossRef](#)]
49. Jiang, J.; Ma, J.; Chen, C.; Wang, Z.; Cai, Z.; Wang, L. SuperPCA: A superpixelwise PCA approach for unsupervised feature extraction of hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4581–4593. [[CrossRef](#)]
50. Chen, L.; Liu, D.; Huang, S. Decorrelate hyperspectral images using spectral correlation. In Proceedings of the 27th International Congress on High-Speed Photography and Photonics, Xi'an, China, 11 January 2007; Volume 6279, p. 627937.
51. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016.
52. Baumgardner, M.F.; Biehl, L.L.; Landgrebe, D.A. 220 band aviris hyperspectral image dataset: June 12, 1992 indian pine test site 3. *Purdue Univ. Res. Repos.* **2015**, *10*, R7RX991C.
53. Bazine, R.; Wu, H.; Boukhechba, K. Spectral DWT multilevel decomposition with spatial filtering enhancement preprocessing-based approaches for hyperspectral imagery classification. *Remote Sens.* **2019**, *11*, 2906. [[CrossRef](#)]
54. Chan, R.H.; Kan, K.K.; Nikolova, M.; Plemmons, R.J. A two-stage method for spectral–spatial classification of hyperspectral images. *J. Math. Imaging Vis.* **2020**, *62*, 790–807. [[CrossRef](#)]
55. McHugh, M.L. Interrater reliability: The kappa statistic. *Biochem. Medica Biochem. Medica* **2012**, *22*, 276–282. [[CrossRef](#)]
56. Story, M.; Congalton, R.G. Accuracy assessment: A user's perspective. *Photogramm. Eng. Remote Sens.* **1986**, *52*, 397–399.