remote sensing

MDPI

Article

A Deep Learning-Based Hyperspectral Object Classification Approach via Imbalanced Training Samples Handling

Md Touhid Islam [1,†], Md Rashedul Islam [1,†], Md Palash Uddin [1,2] and Anwaar Ulhaq [3,*]

[1] Department of Computer Science and Engineering, Hajee Mohammad Danesh Science and Technology University, Dinajpur 5200, Bangladesh; shourovhstu17@gmail.com (M.T.I.); rashedul_cse@hstu.ac.bd (M.R.I.); palash_cse@hstu.ac.bd or m.uddin@deakin.edu.au (M.P.U.)
[2] School of Information Technology, Deakin University, Geelong, VIC 3220, Australia
[3] School of Computing, Mathematics and Engineering, Charles Sturt University, Port Macquarie, NSW 2444, Australia
* Correspondence: aulhaq@csu.edu.au
† These authors contributed equally to this work.

Abstract: Object classification in hyperspectral images involves accurately categorizing objects based on their spectral characteristics. However, the high dimensionality of hyperspectral data and class imbalance pose significant challenges to object classification performance. To address these challenges, we propose a framework that incorporates dimensionality reduction and re-sampling as preprocessing steps for a deep learning model. Our framework employs a novel subgroup-based dimensionality reduction technique to extract and select the most informative features with minimal redundancy. Additionally, the data are resampled to achieve class balance across all categories. The reduced and balanced data are then processed through a hybrid CNN model, which combines a 3D learning block and a 2D learning block to extract spectral–spatial features and achieve satisfactory classification accuracy. By adopting this hybrid approach, we simplify the model while improving performance in the presence of noise and limited sample size. We evaluated our proposed model on the Salinas scene, Pavia University, and Kennedy Space Center benchmark hyperspectral datasets, comparing it to state-of-the-art methods. Our object classification technique achieves highly promising results, with overall accuracies of 99.98%, 99.94%, and 99.46% on the three datasets, respectively. This proposed approach offers a compelling solution to overcome the challenges of high dimensionality and class imbalance in hyperspectral object classification.

Keywords: object classification; dimensionality reduction; non-negative matrix factorization; imbalanced data handling; deep learning; batch normalization

Citation: Islam, M.T.; Islam, M.R.; Uddin, M.P.; Ulhaq, A. A Deep Learning-Based Object Classification Approach of Hyperspectral Image with Imbalanced Training Samples Handling. *Remote Sens.* **2023**, *15*, 3532. https://doi.org/10.3390/rs15143532

Academic Editor: Danfeng Hong

Received: 27 May 2023
Revised: 1 July 2023
Accepted: 12 July 2023
Published: 13 July 2023

1. Introduction

A Hyperspectral Image (HSI) is a type of image that captures spectral data across hundreds of contiguous narrow wavelength bands [1] as well as spatial information about the objects in a scene. This provides a rich and detailed representation of the objects in a scene, capturing information about the color and intensity of light as well as the spectral features of the objects [2]. As a result, HSIs provide a unique combination of spectral and spatial information, making them useful for a variety of applications, including mineral and crop mapping, environmental monitoring, and military reconnaissance [3,4]. An HSI often has a large number of spectral bands, resulting in numerous features in the data. As such, the challenge of high dimensionality is a major issue in the field of HSI. The difficulty of interpreting HSIs with a large number of dimensions (or spectral bands) is often referred to as the "curse of dimensionality" [5]. High dimensionality makes data processing and analysis difficult, increases the danger of overfitting, makes visualization difficult, and limits statistical methods. The high dimensionality of HSI demands more computational power, making it harder to process and analyze the data efficiently. To

address these challenges, it is important to choose appropriate techniques and approaches for processing and analyzing HSI and to ensure that the models used can handle the high dimensionality effectively [6].

The problems around high dimensionality in object recognition from HSIs is currently addressed through classic dimensionality reduction approaches such as feature extraction and feature selection [7,8] methods. Feature extraction approaches, such as Principal Component Analysis (PCA) [9], Minimum Noise Fraction (MNF) [10], Segmented PCA [11], Segmented MNF [12], and Non-Negative Matrix Factorization (NMF) [13], aim to identify a compact set of features that represent the most important information in the data. Feature selection approaches, such as correlation-based and mutual information-based methods, focus on identifying the most relevant features and eliminating the less important ones. However, deep learning techniques, such as Convolutional Neural Networks (CNNs) [14] and Deep Belief Networks (DBNs) [15], are promisingly being applied to handle the high dimensionality problems in HSI. The best approach for a specific application depends on its needs and requirements; by carefully considering the techniques used to process and analyze the data, high dimensionality problems in HSI can be effectively addressed. Although PCA is a popular dimensionality reduction technique that is used in many fields, including hyperspectral imaging, it can have limitations in the context of HSIs due to the unique characteristics of HSI data [16]. One of the main problems with using PCA for dimensionality reduction in HSIs is that it is a linear technique that assumes a linear relationship between the features in the data [17]. However, the spectral signatures of materials in a scene can be nonlinear, which can result in PCA losing important information during the dimensionality reduction process. Another problem with PCA for HSI is that it may not preserve the spatial information in the data. This can be a critical issue in HSI, where spatial information can be used to identify the location and distribution of materials in a scene. Segmented PCA provides a superior solution over standard PCA for dimensionality reduction in hyperspectral imaging [18]. This is because it divides the data into distinct segments and processes each segment individually, enabling more effective and efficient reduction of the data's dimensionality while preserving important spectral and spatial information. On the other hand, the standard PCA approach processes the entire dataset as one entity, which may not be suitable for handling complex relationships between features. In addition to issues with its effectiveness, standard PCA in HSI can encounter problems with preserving spectral and spatial information, as it is sensitive to noise and outliers, and is limited to linear relationships between features. The advantage of using Segmented PCA in HSI is that it enables more efficient and effective reduction of data dimensionality while retaining important spectral and spatial information through the division and individual processing of data segments.

MNF is another method for dimensionality reduction in HSI that aims to preserve the most important information in the data while reducing the dimensionality [19]. The drawback of MNF in HSI is that it can lead to the loss of important spectral information if the noise fraction is not accurately estimated, and it can be sensitive to outliers in the data. Based on the concept of segmentation, several methods have been proposed to eliminate the constraints of MNF, such as Segmented MNF and Band grouping MNF (Bg-MNF). The main difference between regular MNF and segmented MNF is that the latter involves dividing the data into segments or regions before applying the MNF transformation. The idea behind this approach is to preserve the unique spectral information in each segment and to reduce the dimensionality of the data in a more effective and efficient way. Bg-MNF is an extension of the MNF technique which is widely used for spectral data compression and feature extraction. Bg-MNF is designed to improve the performance of MNF in the presence of high levels of noise. It does this by grouping the data into smaller sub-bands, or segments, before applying the MNF transformation. This allows the algorithm to identify and suppress the background noise in each sub-band more effectively. The main advantage of Bg-MNF over standard MNF is that it can provide a more robust and stable representation of the objects in the scene even in the presence of high levels of noise. This

makes Bg-MNF a useful tool for various applications, including target detection, material classification, and environmental monitoring. By reducing the dimensionality of the data while preserving important information, Bg-MNF can improve the efficiency and accuracy of these applications. Determining the optimal number and size of the sub-bands or segments in band grouping MNF can be challenging. If the sub-bands are too large, the method may not be effective in suppressing the background noise, while if they are too small the method may become computationally impractical.

NMF is a linear algebraic technique that factorizes a non-negative matrix into two non-negative matrices to extract spectral features and reduce dimensionality in hyperspectral imaging [20]. NMF is often considered to be a better alternative to both MNF and PCA for HSI analysis, as it imposes a non-negativity constraint on the factorized matrices, which helps to preserve important spectral information and reduce the impact of noise in the data while producing factors that are non-negative and have a physical meaning. This makes NMF a more interpretable [21] and robust solution compared to PCA while providing a more computationally efficient solution compared to MNF. The main problem with standard NMF for HSI analysis is that it may not be able to effectively capture and separate the different materials in a scene, leading to spectral mixing and loss of information. Segmented NMF can address this issue by dividing the hyperspectral data into segments or clusters before applying the NMF factorization, which helps to reduce the dimensionality and improve the accuracy of the resulting factors. However, spectral redundancy may persist in the data. In light of the issue, we suggest a method for subgrouping using a correlation matrix that shifts from one set of frequencies to another; this method is termed subgrouping-based NMF. The correlation matrix image is utilized in order to partition the image into smaller groupings. Therefore, SNMF uses NMF to pull out features from each group while keeping the amount of computing needed to a minimum. Classification rankings are determined using a feature selection technique that uses correlation coefficient matrices to prioritize informative feature traits while reducing irrelevant features (mRMR). After the dimensionality of the HSI has been reduced, the next step is to recognize or classify each pixel in the image.

There are several commonly used classification methods for HSIs, including Support Vector Machine (SVMs) [22], Artificial Neural Networks (ANNs) [23], Random Forest (RF) [24], and Convolutional Neural Networks (CNNs) [25]. Deep learning algorithms are often considered more suitable for HSI classification than machine learning algorithms because they can handle both spectral and spatial variability in hyperspectral data through multiple layers of nonlinear transformations. Deep learning models, especially CNNs, have become increasingly popular in the field of object classification for hyperspectral imagery because they are capable of processing high-dimensional data and extracting hierarchical features from it. Unlike machine learning models such SVM or RF, which can struggle with high-dimensional data and require separate feature extraction and classification steps, CNNs are end-to-end learning models that can learn both the feature extraction and classification tasks simultaneously [26]. Hence, state-of-the-art algorithms such as 2D CNN, 3D CNN [27], fast and compact 3D CNN [28], and HybridSN [29] have shown great promise in achieving high accuracy and robustness in HSI classification; 2D CNNs are limited to spatial information, and may not be able to capture the spectral information that is crucial for accurate classification, while 3D CNNs can leverage both spectral and spatial information, although they have higher computational cost due to the need for 3D convolutions. Fast and compact 3D CNNs aim to reduce the computational cost of 3D CNNs while maintaining high accuracy; however, they may still be too computationally expensive for resource-constrained applications. HybridSN combines both spectral and spatial information for HSI classification, although its reliance on the assumption that the spectral and spatial features can be effectively separated may not hold in all cases. This issue can be potentially addressed through the use of optimization techniques. From the above discussion, it is evident that by integrating dimensionality reduction techniques as a preprocessing step, deep learning models can effectively target the most informative

features of the data, leading to superior classification performance with enhanced efficiency and accuracy. Even though effective methods have been employed for dimensionality reduction and constructive classification for HSI, the class imbalance problem persists. This problem is often overlooked in HSI object classification due to its complexity; however, it is crucial to address this issue, as it can greatly affect the accuracy of a classification model.

On the other hand, data imbalances can be easily found in HSI object recognition tasks. Imbalance occurs when one class in the data has significantly more samples than the other class, leading to a bias towards the majority class. This can result in poor performance, low sensitivity, and increased false positives for the minority class while insufficiently representing majorities in the training data [30,31]. Several solutions have been proposed to address this issue, including re-sampling techniques, cost-sensitive learning, and class weighting [32]. In statistical analysis, re-sampling techniques are frequently employed to test hypotheses and evaluate model performance by balancing the class distribution. Oversampling the minority class or undersampling the majority class are among the common methods of achieving balance [33]. Undersampling is a type of re-sampling technique that involves reducing the size of the majority class by removing samples from it. This is done in order to balance the class distribution and make the minority class more prominent in the dataset. Oversampling, on the other hand, involves increasing the size of the minority class by replicating its samples. This approach is used to mitigate the effect of class imbalance and make the minority class more representative in the dataset. There are several types of oversampling and undersampling methods in the field of machine learning, including Random Oversampling [34], Random Undersampling [35], Synthetic Minority Oversampling Technique (SMOTE) [36], Near-Miss [37], etc. In our view, the biggest drawback of oversampling is that it increases the likelihood of overfitting by creating perfect replicas of existing cases. Random oversampling generates exact copies of minority class instances, which may lead to a higher risk of overfitting. Random undersampling has the issue of potentially losing important information from the minority class. Although SMOTE is a popular oversampling method, it may generate noisy data and cause the model to overfit. While the near-miss method is a simple undersampling method, it can have limitations in handling complex data distributions, and has the problem of potentially removing important information in the majority class that could be useful for identification. In this situation, a combination of oversampling and undersampling can balance class distribution in imbalanced datasets and improve classifier performance by avoiding class imbalance biases. This reduces both the risk of overfitting from oversampling and potential loss of information from undersampling. Careful consideration of oversampling and undersampling techniques is important for an accurate representation of the data's true distribution. A well balanced combination of random oversampling and random undersampling can effectively address data imbalance, although it may result in over-generalization, overfitting, increased variance, loss of information from the majority class, and computational issues in high-dimensional datasets. Taking into account the considerations discussed above, we have developed a framework for classifying hyperspectral remote sensing images using the above literature as our source of inspiration. The proposed framework has the following key contributions:

- To address the issue of imbalanced classes, a modified combined approach was adopted using both undersampling and oversampling methods. This technique involves generating synthetic images and undersampling classes that have an excess of samples. The objective is to obtain balanced classes and datasets. The method uses a combination of undersampling and oversampling, where large classes are downsampled using one method and minority classes are oversampled with another method, resulting in a dataset that is suitably balanced.
- To streamline the classification process and reduce computational load, a technique was implemented to reduce the number of features utilized. This technique involves a new subgrouping-based method in combination with a greedy feature selection

technique, which ensures that only the most crucial spectral features are utilized while minimizing extraneous ones.

- A hybrid 3D-2D CNN network was utilized; in order to attain the highest level of accuracy possible, the 3D-CNN and 2D-CNN layers that make up the proposed model are combined to make full use of the spectral and spatial feature maps, ensuring that the model is as accurate as possible.

The remainder of this paper is structured as follows. Section 2 details our proposed approach to reducing the number of dimensions and presents our hybrid 3D-CNN/2D-CNN modified deep learning model for HSI classification along with optimization techniques. In Section 3, we provide information about the datasets, experimental setup, results, and analysis of a comparative assessment of our proposed techniques and the alternative baseline approaches. Finally, we conclude with observations and a discussion of potential future avenues for research.

2. Proposed Methodology

2.1. Motivation

Recent advances in image classification performance have been made by CNN models. CNN is one of the most popular network frameworks. The CNN model is widely used for HSI classification due to its exceptional ability to detect both spectral and spatial features, which gives it an edge over other learning models. Nonetheless, this model has limitations. For example, it is prone to the outputs collapsing to the local minimum during the gradient descent process, and a significant amount of valuable information is lost during pooling. It is well established that the preprocessing step is pivotal in object classification of HSI images. Traditional approaches such as different PCA versions, MNF, or other approaches require a great deal time, and can extract duplicate features. A technique for optimal dimension reduction is required in this case. A 2D-CNN model cannot generate distinctive feature maps from spectral dimensions without additional support. On the other hand, a deep 3D-CNN has higher computational complexity, leading to suboptimal performance for classes with comparable textures across multiple spectral bands. Thus, maximum accuracy can be achieved by combining 2D-CNN and 3D-CNN, enabling them to fully utilize both spectral and spatial feature maps.

2.2. Proposed Resampling Method

Resampling approaches are intended to modify the class distribution by adding or removing samples from the training dataset. After the class distributions are normalized, the modified datasets may be used to successfully fit the whole suite of typical machine learning classification methods. The primary flaw in the random undersampling approach is its propensity to exclude information that could be crucial to the induction procedure. In order to overcome the restrictions of non-heuristic judgments, several undersampling proposals employ heuristics to eliminate the necessity of removing data. The sample of the majority class or the minority class chosen through random sampling may introduce bias due to potential under-representation or over-representation. Random sampling entails drawbacks such as biased outcomes and the need for extensive resources and time to collect an adequately representative sample. In contrast, "Near-Miss" undersampling techniques offer an additional advantage by selecting samples based on their proximity to instances from the majority class. This approach effectively removes instances from the larger class that are near instances of the smaller class, resulting in improved balance within the dataset. The procedures for this method are as follows:

1. Calculate the distance between each instance in the majority class and each instance in the minority class;
2. Select the instances from the majority class that have the shortest distance to the instances in the minority class;
3. Store the selected instances for elimination;
4. Determine the number of instances q in the minority class;

5. Return q × p instances of the larger class as a minority class.

In contrast, SMOTE can be seen as a more advanced type of oversampling or a targeted approach to augmenting data. One advantage of SMOTE is that it generates synthetic data points that are similar to the original ones and possess slight variations, instead of simply duplicating existing data points. This effectively enlarges the minority class and creates more balanced representation of both classes.

The equation for generating a synthetic instance using SMOTE can be represented as Equation (1):

$$L = I + (Ng - I) \times Rd \tag{1}$$

where L represents the synthetic instance, I is a randomly selected minority class instance, Ng is a randomly selected neighbor from the minority class, and Rd is a random value between 0 and 1. SMOTE helps to overcome the problem of class imbalance, thereby improving the performance of machine learning algorithms, especially in situations where the minority class is underrepresented. Consequently, while each form of resampling can be beneficial independently, combining them may yield improved results [38]. The initial step in the present study involves employing a combined resampling technique to adjust the training data. The majority classes are initially reduced to match the average number of total samples using the near-miss technique. Next, each minority class is expanded through SMOTE oversampling. Finally, the resampling process concludes by ensuring that the number of samples in each class equals the average number of samples across all classes in the dataset.

2.3. Proposed Dimensionality Reduction Method

Statistical correlation analysis examines the relationship between two continuous variables. A correlation matrix is a two-dimensional matrix containing correlation coefficients. The correlation matrix shows all potential value pairs in a dataset [37]. Using this tool, a large dataset can be summarized rapidly and its trends can be detected. In contrast to conventional MI values, which are more difficult and time-intensive to compute, correlation values are standard and computationally efficient, taking values between −1 and +1. The covariance is normalized by multiplying it by the standard deviations of the correlation values to obtain the correlation. Correlation coefficients may be used to describe the relationship between two features $\boldsymbol{Q}$ and $\boldsymbol{R}$ as follows:

$$Corr(\boldsymbol{Q}, \boldsymbol{R}) = \frac{Cove(\boldsymbol{Q}, \boldsymbol{R})}{\sigma\boldsymbol{Q} \times \sigma\boldsymbol{R}}. \tag{2}$$

Here, the covariance is represented by $Cove(\boldsymbol{Q}, \boldsymbol{Q})$, while the standard deviations of $\boldsymbol{Q}$ and $\boldsymbol{Q}$ are represented by $\sigma\boldsymbol{Q}$ and $\sigma\boldsymbol{Q}$, respectively. Greater levels of similarity indicate stronger correlation. If two variables have no connection, the correlation coefficient is zero. Other varied connections may occur. A positive correlation means that one variable boosts the other. A negative correlation means that the two variables change in opposing ways [38]. The correlation matrix is determined by computing the correlation coefficient between a pair of HSI bands. Typically, HSI bands that are in close proximity exhibit higher correlation coefficient values than those that are further apart. To divide and arrange the HSI bands, an image of the correlation matrix for the original data cube can be utilized. The partitioning of the correlation matrix images is based on correlation coefficient values that exceed the specified threshold when viewed by the user. The proposed methodology involves setting a user-defined threshold and identifying the boundary in the correlation matrix images as well as the diagonal direction where the average correlation coefficient surpasses the threshold. An important factor here is the threshold value, which changes from dataset to dataset. In our case, the threshold values observed for different datasets are t_1 = 0.47 for SA, t_2 = 0.68 for PU, and t_3 = 0.39 for KSC, as obtained through a grid search. Although other methods, such as random search or Bayesian optimization, may offer enhanced parameter search efficiency, we specifically opted for grid search due to its

straightforward implementation and simplicity. A certain number of groups is established when the user-defined threshold is reached. The number of groups grows if the threshold is low, and uncorrelated bands are clustered together. The proposed dimensionality reduction approach is shown in Figure 1 and Algorithm 1. Let us suppose that $\mathbf{X} \in \mathbf{P}^{M \times N \times B}$ is the initial input, M is the width, N is the height, and B is the total number of spectral bands. In this case, we obtain a cube of $\mathbf{X}$ in the HSI dimension. While preserving the same spatial dimensions, the subgrouping-based NMF technique decreases the number of spectral bands from B to R, meaning that the data cube is given by $\mathbf{X} \in \mathbf{P}^{M \times N \times R}$.

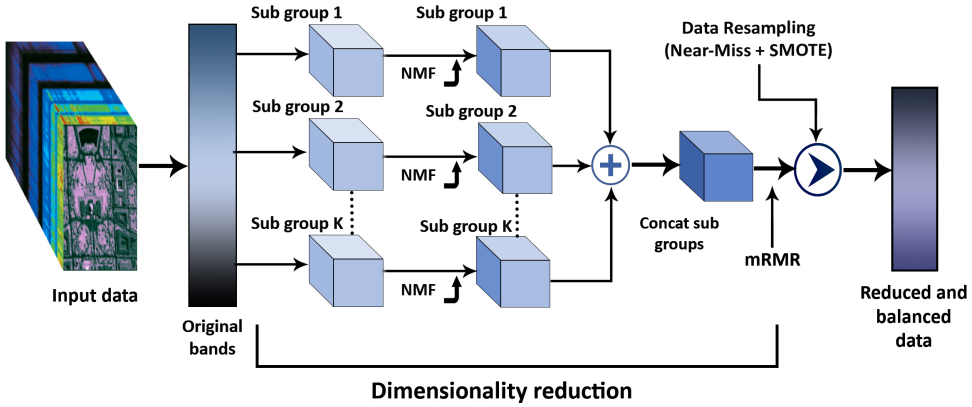


Figure 1. The architecture of proposed subgrouping-based NMF method for dimensionality reduction.

Algorithm 1 Overall proposed methodology

1: **Input** Input the original HSI dataset
2: **Reduce** dimensionality:
   a. **Apply** subgrouping NMF for spectral features
   b. **Select** features with min redundancy and max relevance
3: **Resample** the input data
   a. **Undersample** majority classes with Near-Miss to K sample, where K = Average (Total sample)
   b. **Oversample** the minority classes with SMOTE to K sample
4: **Pass** the pre-processed data to the proposed deep learning model
5: **Extract** spectral-spatial features with proposed deep 3D 2D CNN model
6: **Classification** with proposed deep learning model
7: End

In this way, only spectral bands that are essential for object recognition are eliminated, and reduction is carried out in a way that retains the intact spatial information. The HSI data cube is partitioned into tiny overlapping 3D patches, with the truth labels determined by the label of the patch's center pixel, allowing it to be used with image classification methods. To encompass the $Z \times Z$ window or geographical breadth and all R spectral bands, we generate the 3D adjacent patches $\hat{\mathbf{Q}} \in \mathbf{P}^{Z \times Z \times R}$ from $\mathbf{X}$, with their centers at (a, b). With $\mathbf{P}$ as input, $(M - Z + 1)(N - Z + 1)$ can be used to determine the overall number of 3D patches that can be created. This is why the 3D patch at coordinates (α, β) is written as $\mathbf{P}_{(\alpha,\beta)}$; it includes the space width from $\left(\frac{(\alpha-(Z-1))}{2}, \frac{(\alpha+(Z-1))}{2}\right)$ and height form $\left(\frac{(\beta-(Z-1))}{2}, \frac{(\beta+(Z-1))}{2}\right)$. In this research, we suggest a modified network that can benefit from the automated feature learning capabilities of both 2D-CNNs and 3D-CNNs.

2.4. Proposed Deep Learning Model and Classification

The proposed approach utilizes a fusion of a 3D-CNN and 2D-CNN, allowing the extraction of significant spatial and spectral feature maps from the HSI in a cost-effective manner. The

model design involves stacked 3D convolution layers followed by a decreasing dimension block that includes a Conv3D layer, reshaping operation, and another Conv3D layer. The resulting feature maps are reshaped and then processed through a Conv2D layer, which learns additional spatial features. The Conv2D layer's output undergoes normalization before being passed to the first fully connected layer subsequent to the dropout layer. The overall proposed process is depicted in Figure 2 and Algorithm 2, both of which illustrate the unique architecture of the model. The proposed model comprises seven layers, starting with 3D convolutions (C1-C3) at levels one to three for extracting spectral–spatial information, followed by 2D convolutions (C4-C5) at the fourth and fifth levels, and finally two fully connected layers (F1-F2) at the end of the model. In a fully connected layer, each neuron communicates with all others in the layer beneath it and transmits the resulting value to the classifier. Pooling layers reduce the spatial resolution of images to reduce the size of feature maps and simplify calculations. The pooling layer does not store any extra data. According to the suggested architecture, each of the first three convolution layers uses a three-dimensional convolution kernel with dimensions of $8 \times 3 \times 3 \times 3 \times 1$ (i.e., $K_1^1 = 3, K_2^1 = 3, K_3^1 = 3$), $16 \times 3 \times 3 \times 1 \times 8$ (i.e., $K_1^2 = 3, K_2^2 = 3$, $K_3^2 = 1$), and $32 \times 3 \times 3 \times 1 \times 16$ (i.e., $K_1^3 = 3, K_2^3 = 3, K_3^3 = 1$), respectively; more specifically, $32 \times 3 \times 3 \times 1 \times 16$ implies 32 3D kernels of size $3 \times 3 \times 1$ (three dimensions, one spectral and two spatial) for each of the sixteen three-dimensional input feature maps. In order to perform 2D convolution processes, the data in the fourth layer must be a 3D image. When performing 2D convolution processes, it is first necessary to resize the input size to accommodate these procedures; 4^th-layer inputs are prepared, then a 2D convolution operation with kernel dimensions of 3×3 (i.e., $K_1^4 = 3, K_2^4 = 3$) is performed on these inputs to generate 64 feature maps for the output of the fifth layer (i.e., 32 2D input feature maps are employed). The size of the following layer, which is part of the depthwise 2D convolution kernel, is $128 \times 3 \times 3 \times 32$ (i.e., $K_1^5 = 3, K_2^5 = 3$). In order to counteract the issue of overfitting and enhance the rate at which the network converges, we integrate the batch normalization optimization method after every convolution layer. This strategic inclusion of batch normalization effectively mitigates overfitting, resulting in a more stable network. The adverse effects of overfitting are minimized by normalizing the output of each layer, leading to faster convergence and improved learning efficiency during the training process. In the end, all 256 neurons are linked to one another after the fifth layer is flattened. Table 1 provides a brief explanation of the proposed model, including information on the number of parameters, types of layers, and dimensions of the resulting maps. It is evident that the first thick layer (fully connected) has the greatest number of parameters. To account for each class included in a particular dataset, the deepest hidden layer has the same number of neurons. This means that the suggested model's total number of parameters is class-specific. For the Salinas, Pavia University, and Kennedy Space Center datasets, the suggested model has 1,912,688, 1,911,785, and 1,912,430 trainable weight parameters, respectively. All weights start with a random value and are then taught via the backpropagation method, Adam optimizer, and Softmax classification.

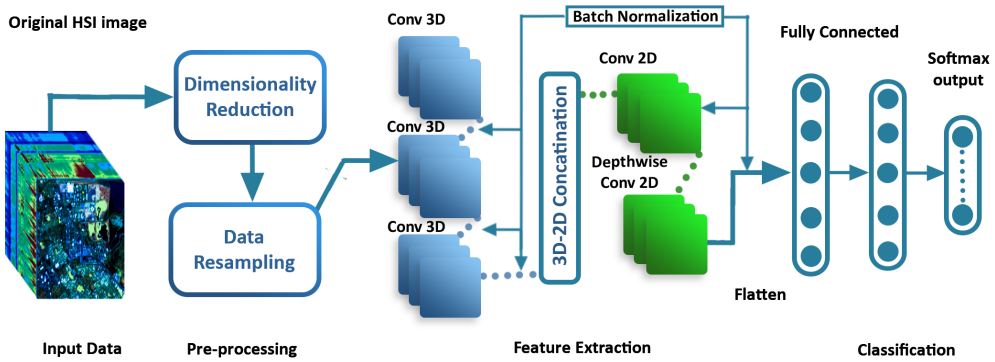


Figure 2. The architecture of the proposed optimized 3D-2D CNN model for spectral–spatial feature extraction and classification.

Algorithm 2 Dimensionality reduction.

Input: Original HSI data in $M \times N \times B$, where M, N, and B indicate image width, height, and the number of spectral bands.
Generate the correlation coefficient matrix for the input dataset
Split whole data into several subgroups based on their variance
Use threshold to separate groups
if $dataset = Salinasscene$ **then**
    $threshold = t_1$
else if $dataset = PaviaUniversity$ **then**
    $threshold = t_2$
else if $dataset = KSC$ **then**
    $threshold = t_3$
end if
Apply NMF to each subgroup and reduce features to n number $(Wheren \geq 10)$ and merge newly reduced subgroups.
Exit

Table 1. Statistics of the SC dataset, including the name and number of total samples for each class.

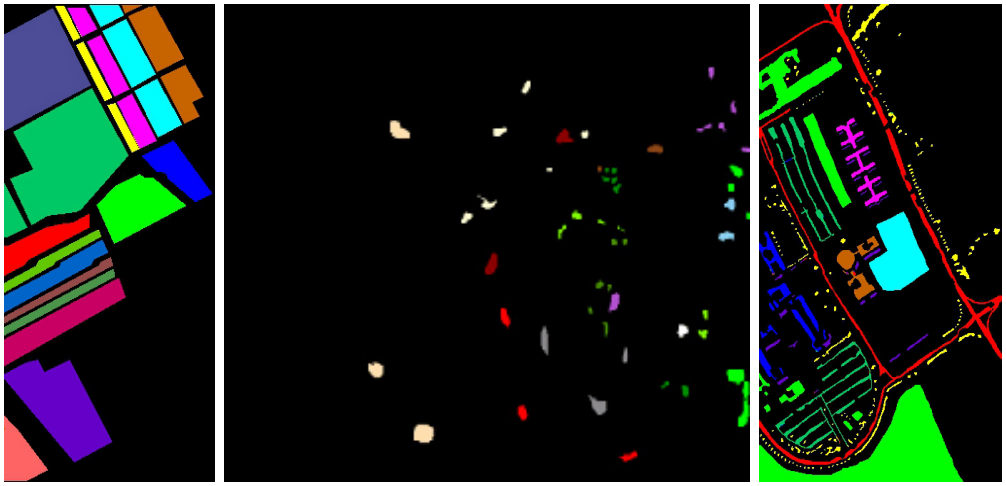
No	Class Labels	Samples
1	_Brocoli_ green_weeds_1_	2009
2	_Brocoli_ green_weeds_2_	3726
3	_Fallow__	1976
4	_Fallow_ rough_ plow_	1294
5	_Fallow_ smooth_	2678
6	_Stubble__	3959
7	_Celery__	3579
8	_Grapes_ untrained_	11,271
9	_Soil_ vinyard_ develop_	6203
10	_Corn_ senesced_ green_ weeds_	3278
11	_Lettuce_ romaine_ 4wk_	1068
12	_Lettuce_ romaine_ 5wk_	1927
13	_Lettuce_ romaine_ 6wk_	916
14	_Lettuce_ romaine_ 7wk_	1070
15	_Vinyard_ untrained_	7268
16	_Vinyard_ vertical_ trellis_	1807

3. Experiments

3.1. Dataset Description

1. Salinas Scene: the Salinas Scene (SC) dataset for HSIs was acquired in 1992 using the AVIRIS sensor, which obtained 566 images of the Salinas Valley in California. The initial image comprised 224 bands [39]. To create an HSI dataset with 204 bands, we excluded twenty water absorption bands: bands 108–112, 154–167, and 224. The spatial dimensions of the scene are 512 × 217 pixels. The scene has a total of sixteen labeled classes. Figure 3a displays ground truth images of the SC dataset, and Table 1 shows the sample distributions for the experiment.
2. Pavia University: this dataset consists of ROSIS optical sensor data collected by the University of Pavia (PU) over the city of Pavia in northern Italy. The PU dataset has a spatial resolution of 1.3 m, and has 103 spectral bands in addition to its 610 × 610 spatial resolution [39]. There are nine ground truth classes in PU. Further information on the PU experimental datasets may be accessed at [23]. Figure 3c displays the ground truth images for each of the experimental datasets. The sample distributions in the original dataset are shown in Figure 3c and Table 2.
3. Kennedy Space Center: the final dataset, referred to as the Kennedy Space Center (KSC) dataset, was acquired on March 23, 1996 using the AVIRIS sensor, which captured the KSC region in Florida. This HSI dataset is characterized by spatial

dimensions of 512 × 614 pixels and comprises 224 spectral bands [39]. Following the removal of 48 noisy bands, a total of 172 spectral bands were obtained. The dataset consists of thirteen labeled classes, as outlined in Table 3. The sample distributions in the original dataset are shown in Figure 3b and Table 3.



(**a**) (**b**) (**c**)

Figure 3. Ground truth images from the datasets: (**a**) SC, (**b**) KSC, and (**c**) PU.

Table 2. Statistics of the PU dataset, including the name and number of total samples for each class.

No	Class Labels	Samples
1	_Asphalt__	6631
2	_Meadows__	18,649
3	_Gravel__	2099
4	_Trees__	3064
5	_Painted_ metal_ sheets_	1345
6	_Bare_ Soil_	5029
7	_Bitumen__	1330
8	_Self_ Locking_ Bricks_	3682
9	_Shadows__	947

Table 3. Statistics of the KSC dataset, including the name and number of total samples for each class.

No	Class Labels	Samples
1	_Scrub__	761
2	_Willow_ swamp_	243
3	_CP_ hammock_	256
4	_Slash_ pine_	252
5	_Broadleaf__	161
6	_Hardwood__	229
7	_Swamp__	105
8	_Graminoid_ marsh_	431
9	_Spartina_ marsh_	520
10	_Cattail_ marsh_	404
11	_Salt_ marsh_	419
12	_Mud_ flat_	503
13	_Water__	927

3.2. Experimental Setup and Hyperparameter Tuning

Before discussing the specifics of the experimental results, the different configurations of the deep learning approaches used in our research are described in depth. Each dataset

contained distinct numbers of training and testing samples. Table 4 illustrates how the synthetic and real samples in the training data as well as the test samples are distributed after resampling. To achieve a fair comparison, we extracted the same spatial–spectral dimension in 3D patches of input volume for the different datasets, such as $25 \times 25 \times 5$ for all datasets. The article delves into the selection process for determining the window size of 3D patches, which has been concluded to be 25×25. This decision is supported by the results of a comprehensive window size analysis experiment, visually presented in Figure 4. Notably, this study revealed that a 25×25 window size yields the highest accuracy. Furthermore, the top features selected using various dimensionality reduction techniques are showcased in Table 5. The data were then sent to the deep learning model. The model has a total of five convolution layers (three 3D-Conv and two 2D-Conv layers) and two fully connected layers. The suggested model is briefly described in Table 6, which includes details on the number of parameters, kinds of layers, and sizes of the resulting maps. The images are then classified using a Softmax layer with n classes, where n is the number of classes used across all datasets. In order to preserve as much data as possible for each pixel, the suggested network layout does not include pooling layers. Furthermore, we utilize Adam for network training, with a mini-batch size of 256 and a learning rate of 0.001. The training of the network lasts for a total of 120 epochs. During training, no extra data are utilized in any way. Two different sample ratios were used as the foundation for the investigations.

The experimental design of the study involved two instances, in which 20% of the training data was separated from the testing data for validation purposes. The first instance consisted of a training and testing ratio of 10:90%, wherein 10% of the total data were allocated for training the model and 90% of the total data were reserved for testing the model's performance. The second instance involved a training and testing ratio of 20:80%, wherein 20% of the total data were designated for training the model and 80% of the total data were used to assess the model's performance. This approach of splitting the data into training, testing, and validation subsets helps in evaluating a machine learning model's performance on unseen data and prevents overfitting of the model on the training data. Furthermore, the validation subset helps in fine-tuning the model's parameters, enhancing the model's generalization ability, and improving its predictive performance on new and unseen data. All experiments were implemented using the Tensorflow–Keras framework on Google Colab (Colab specs in brief: graphics processing unit (1xTesla K80) running compute version 3.7 with 2496 CUDA cores and 12GB of GDDR5 Virtual RAM; memory 12.68 GB; disk 78.19 GB). Google Colab IDE's memory limits prevented us from running all the models with more than five features on this dataset.

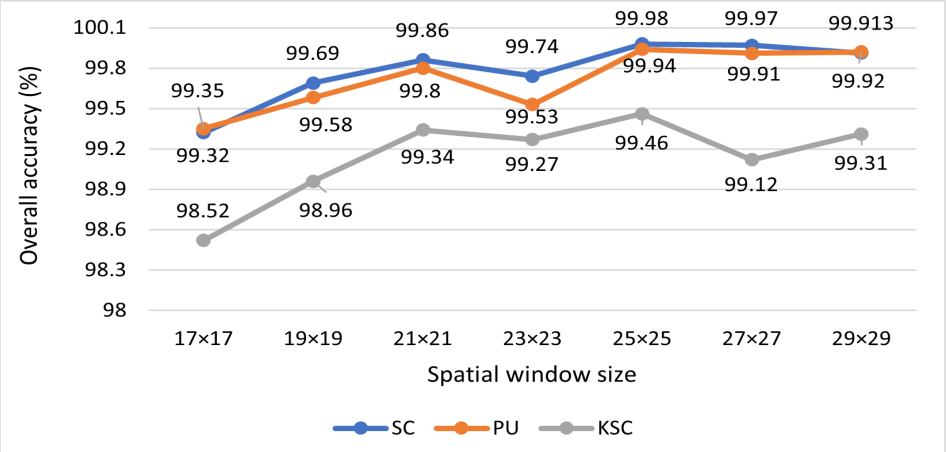


Figure 4. The effect of varying the window size of the input cube on the proposed model's accuracy for the SC, PU, and KSC datasets.

Table 4. Training and testing sample data in detail for four datasets, including sample data.

No	SC				KSC				PU			
	Train			Test	Train			Test	Train			Test
	Real	Synth	Final		Real	Synth	Final		Real	Synth	Final	
1	402	273	675	2708	152	0	80	321	1326	0	950	3803
2	745	0	676	2707	49	31	80	321	3730	0	951	3802
3	395	281	676	2707	51	29	80	321	420	531	951	3802
4	259	416	675	2708	50	30	80	321	613	337	950	3803
5	536	140	676	2707	32	48	80	321	269	682	951	3802
6	792	0	676	2707	46	34	80	321	1006	0	951	3802
7	716	0	676	2707	21	59	80	319	266	685	951	3802
8	2254	0	676	2707	86	0	81	320	736	214	950	3803
9	1241	0	675	2708	104	0	80	321	189	761	950	3802
10	656	19	675	2708	81	0	80	321				
11	214	462	676	2707	84	0	80	321				
12	385	291	676	2707	100	0	81	320				
13	183	493	676	2707	185	0	80	321				
14	214	462	676	2707								
15	1454	0	676	2707								
16	361	312	673	2695								

Table 5. Selected features from the dataset by various dimensionality reduction methods.

Methods	Selected Ranked Features		
	SC	PU	KSC
PCA	PC 1, 2, 3, 4, 5	PC 1, 2, 3, 4, 5	PC 1, 2, 3, 4, 5
Seg-PCA	Group: PC 4:24, 1:1, 3:14, 2:12, 2:11	Group: PC 2:12, 1:1, 2:11, 1:2, 1:3	Group: PC 1:2, 1:1, 2:4, 3:14, 2:5
MNF	PC 1, 2, 3, 4, 5	PC: 1, 2, 3, 4, 5	PC 1, 2, 3, 4, 5
Seg-MNF	Group: MNFC 4:1, 1:1, 3:14, 2:11, 1:2	Group: MNFC 1:1, 2:12, 2:11, 1:2, 1:3	Group: MNFC 1:2, 2:4, 1:1, 3:14, 2:5
Bg-MNF	Group: MNFC 1:2, 1:4, 1:6, 1:3, 1:5	Group: MNFC 1:3, 1:2, 1:4, 1:5, 1:7	Group: PC 3:18, 3:16, 3:17, 3:15, 3:19
NMF	NMFC 1, 2, 3, 4, 5	NMFC 1, 2, 3, 4, 5	NMFC 1, 2, 3, 4, 5
Proposed	Group: NMFC1:3, 4:28, 1:2, 1:4, 3:16	Group: NMFC 1:3, 2:13, 1:10, 2:15, 1:7	Group: NMFC 3:13, 2:8, 2:6, 3:16, 2:7

Table 6. A brief overview of the suggested model using a 25 × 25 window size.

Layers	Outputs	Parameters
Input	[(None, 25, 25, 5, 1)]	0
Conv3D(1)	(None, 23, 23, 3, 8)	224
Batch Normalization1	(None, 23, 23, 3, 8)	32
Conv3D(2)	(None, 21, 21, 3, 16)	1168
Batch Normalization2	(None, 21, 21, 3, 16)	64
Conv3D(3)	(None, 19, 19, 3, 32)	4640
Batch Normalization3	(None, 19, 19, 3, 32)	128
Reshape	(None, 19, 19, 96)	0
Conv2D	(None, 17, 17, 32)	27,680
Batch Normalization4	(None, 17, 17, 32)	128
Depthwise Conv2D	(None, 15, 15, 32)	320
Batch Normalization5	(None, 15, 15, 32)	128
Flatten	(None, 7200)	0
Dense	(None, 256)	1,843,456
Dropout	(None, 256)	0
Dense	(None, 128)	32,896
Dropout	(None, 128)	0
Dense	(None, Classes)	2064
Total trainable parameters: 1,912,688, non-trainable parameters: 240		

3.3. Comparison Methods

1. **SVM** [40]: Mounika et al. used Principal Component Analysis (PCA) for feature extraction in the classification of hyperspectral images using Support Vector Machine (SVM). Their paper demonstrated the effectiveness of using PCA as a preprocessing technique for feature extraction in hyperspectral image classification, providing a valuable tool for researchers in remote sensing and image analysis.
2. **3D-CNN** [27]: Li et al. proposed a 3D convolutional neural network (CNN) for spectral–spatial classification of hyperspectral images. Their paper introduced a novel method that utilizes both spectral and spatial information to improve classification performance. The application of 3D-CNNs for HSI classification can pose significant challenges in terms of computational requirements and data availability, potentially restricting its feasibility for certain practical applications.
3. **Fast 3D-CNN** [28]: Ahmad et al. introduced a fast and compact 3D-CNN for hyperspectral image classification that achieved high accuracy while using a relatively small number of parameters. This paper's contribution lies in the development of an efficient method that overcomes the challenge of processing large amounts of hyperspectral data in real-time applications. A potential disadvantage of employing 3D-CNN for hyperspectral image classification is the risk of losing relevant information during the feature extraction process, which can negatively impact classification accuracy.
4. **HybridSN** [29]: HybridSN is a deep learning architecture proposed for efficient hyperspectral image classification. It combines a deep 2D CNN with a shallow 3D CNN to capture both spatial and spectral features of hyperspectral data. The architecture may pose a challenge in terms of hyperparameter tuning, as its performance can be highly dependent on the careful selection and adjustment of these parameters to achieve the best results.
5. **SpectralNET** [41]: this paper proposes a deep learning architecture that combines the spectral–spatial features of hyperspectral data with the multi-resolution analysis capabilities of wavelet transforms, with the aim of improving classification performance.
6. **MDBA** [42]: this paper's main contribution lies in the development of a multibranch 3D Dense Attention Network (MDBA) that incorporates attention mechanisms to capture spatial and spectral information effectively. While the proposed MDBA offers promising results in hyperspectral image classification, it is important to note that the network architecture and attention mechanisms add complexity to the model, requiring careful implementation and large computational resources for practical deployment.
7. **Hybrid 3D-CNN/2D-CNN** [43]: In this paper, the authors proposed a hybrid CNN architecture combining 3D-CNN and 2D-CNN and using depthwise separable convolutions and group convolutions to improve the classification performance of hyperspectral images. However, the lack of comparison with state-of-the-art methods on larger or more diverse hyperspectral datasets makes it difficult to assess the generalizability and scalability of the proposed method beyond the specific conditions used in the study.

3.4. Complexity Analysis

To analyze the complexity of the proposed model, we can consider the number of parameters. Our developed model utilizes a customized hybrid CNN architecture. It consists of several convolutional layers followed by batch normalization, reshaping, and dense layers. The number of parameters in a CNN model depends on the architecture and the size of the layers. We can break down the number of parameters in each layer, using the example of the Salinas Scene dataset:

1. **Conv3D Layer:** the number of parameters in a Conv3D layer can be calculated as $filters \times kernel_Size^3 \times inputChannels + filters$. The first Conv3D layer has eight filters and a kernel size of $(3, 3, 3)$, and the input has one channel; thus, the number of parameters in this layer is $(8 \times 3^3 \times 1) + 8 = 224$ parameters.
2. **BatchNormalization Layer:** the number of parameters in a BatchNormalization layer is equal to four times the number of filters in the previous layer. The first BatchNormalization layer after Conv3D has eight filters; thus, it has $4 \times 8 = 32$ parameters.
3. **Conv2D Layer:** the number of parameters in a Conv2D layer can be calculated as $filters \times kernelSize^2 \times inputChannels + filters$. The Conv2D layer has 32 filters and a kernel size of $(3, 3)$, and the input has 32 channels; thus, the number of parameters in this layer is $(32 \times 3^2 \times 32) + 32 = 29,056$ parameters.
4. **DepthwiseConv2D Layer:** the number of parameters in a DepthwiseConv2D layer can be calculated as $kernelSize^2 \times inputChannels + inputChannels$. The DepthwiseConv2D layer has a kernel size of $(3, 3)$ and 32 input channels; thus, the number of parameters in this layer is $3^2 \times 32 + 32 = 320$ parameters.
5. **Dense Layer:** the number of parameters in a Dense layer can be calculated as $units \times inputDim + units$. The first Dense layer has 256 units and an input dimension of 7200; thus, the number of parameters in this layer is $256 \times 7200 + 256 =$ 1,849,856 parameters.

By summing up the number of parameters in each layer, we can find the total number of trainable parameters in the model, which is 1,912,688 parameters.

In comparison to the two most similar methods, "HybridSN" and "Hybrid 3D-2D CNN", there are both similarities and differences. Both methods utilize a combination of 3D and 2D convolutions for feature extraction. However, the exact dimensions and configurations of the convolutional kernels differ. Regarding the "HybridSN" method, the convolutional kernels have dimensions of $8 \times 3 \times 3 \times 7 \times 1$, $16 \times 3 \times 3 \times 5 \times 8$, and $32 \times 3 \times 3 \times 3 \times 16$ in the subsequent layers. Additionally, a 2D convolution with a kernel dimension of $64 \times 3 \times 3 \times 576$ is applied before the flatten layer. The model consists of a total of 5,122,176 trainable weight parameters. The number of classes in the dataset determines the total number of parameters. On the other hand, the "Hybrid 3D-2D CNN" method comprises seven layers, starting with 3D convolutions and followed by 2D convolutions. The dimensions of the 3D convolution kernels are $8 \times 3 \times 3 \times 3 \times 1$, $16 \times 3 \times 3 \times 38$, and $32 \times 3 \times 3 \times 3 \times 16$. The output of the third layer is resized to fit the input size of the fourth layer, where a 2D convolution with a kernel dimension of 3×3 is applied. A depthwise 2D convolution with a kernel dimension of $128 \times 3 \times 3 \times 64$ is then performed. The model has a total of 1,033,728 trainable weight parameters.

When comparing the proposed model with the above two methods, it can be observed that the proposed model has a larger number of total and trainable parameters. This might indicate both higher model complexity and greater capacity to capture more intricate patterns and features. Additionally, the proposed model has a larger input shape $(None, 25, 25, 5, 1)$, which could allow it to capture more spatial and spectral information from the data. These factors might contribute to the proposed model achieving better results compared to the other two methods, despite their lower number of trainable parameters. However, it is important to consider other factors, such as dataset characteristics, training settings, and evaluation metrics, when drawing a comprehensive comparison between the models.

3.5. Classification Performance

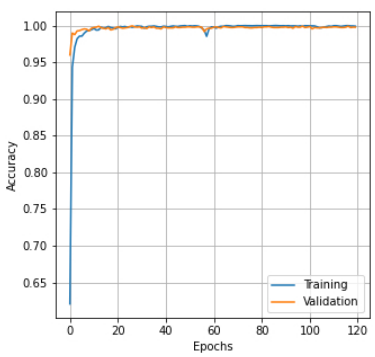
To assess the effectiveness of the proposed model, we selected three sets of freely accessible HSI data. We analyzed the impact of different dimensionality reduction methods on the accuracy of the proposed model, including PCA, Segmented PCA, MNF, Segmented MNF, Bg-MNF, NMF, and the dimensionality reduction method proposed in this paper. For the purpose of comparative analysis, we used state-of-the-art methods such as SVM [44], 3D CNN [27], Fast 3D CNN [28], HybridSN [29], SpectralNET [41], Hybrid 3D 2D CNN [43], etc. Our proposed approach demonstrates superior performance compared to other dimensional-

ity reduction methods when evaluated on these deep learning model. Figures 5–10 provide a detailed analysis and comparison of our method with existing approaches, highlighting its advantages in terms of accuracy and loss metrics for both training and validation data. Additionally, we employed performance measurements such as the overall accuracy (OA), average accuracy (AA) for each class, and Kappa coefficient (Kappa). The OA measure considers the ratio of properly categorized images in the testing dataset to the total number of samples, the AA metric indicates the mean of the accuracy of the image class, and the Kappa metric indicates the weighting of the observed accuracy. Both metrics are used to quantify image quality. Precision, recall, and f-1 score are other kinds of commonly used performance metrics, with their weighted averages being used here as the weighted average can provide more reliable results than a simple average. To calculate the weighted averages, the value of each data point is first multiplied by the weight associated with it, then the resulting total is divided by the number of datapoints.

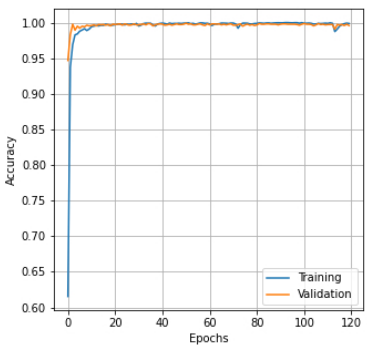
On the basis of the different datasets, we describe the experimental results in different domains of analysis. We note that the OA of the suggested SNC model is greater than 99% in every dataset, resulting in the best performance. In each classification table, the most significant values are bolded; additionally, according to the findings of our experiment, the classification of HSI requires a greater emphasis on spatial context than on spectral correlations.

3.5.1. Results for the SC

The structure of the deep learning models in this experiment were quite similar to those mentioned in the experimental setup, with the only variation being the number of parameters that were altered following parameter tuning, such as the threshold value for segmentation and the number of spatial features after reduction. On the basis of the SC dataset and using training data of 10% and 20%, Table 7 shows the experimental outcomes and effects of several resampling techniques. We used a number of different dimensionality reduction techniques; this analysis is shown in Table 8 along with the results for the proposed method. Table 9 illustrates how the size of the spatial window affects the overall accuracy and training time. Finally, the experimental outcomes of all state-of-the-art strategies tested on the SC dataset are summarized in Table 10. The SNC model's OA is 99.98%. Figures 5 and 6 show the categorization outcomes for each comparative technique as accuracy and loss curves, respectively. The accuracy and loss curves shown here were produced with both training samples and validation samples,. Figures 5g and 6g demonstrate the optimum performance.

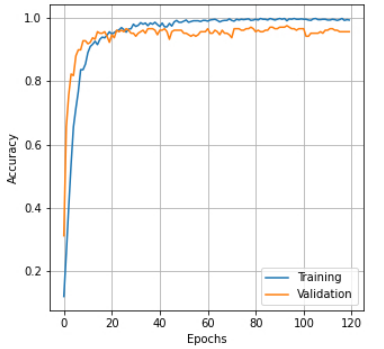


(**a**)

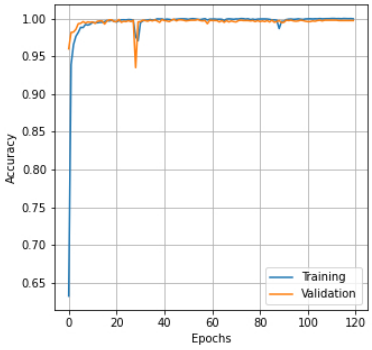


(**b**)

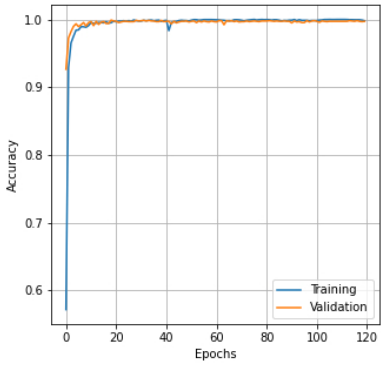
Figure 5. *Cont.*



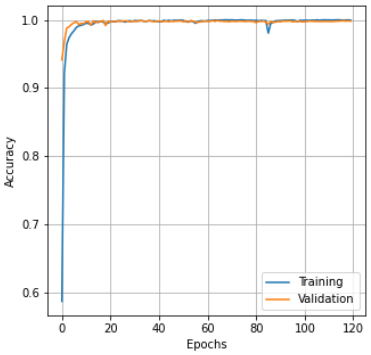
(**c**)



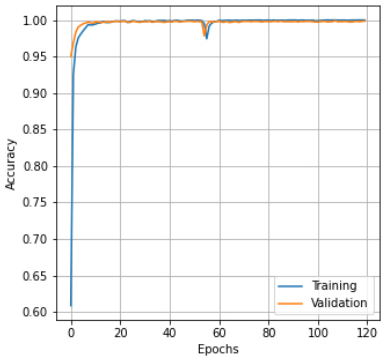
(**d**)



(**e**)



(**f**)



(**g**)

Figure 5. Accuracy curves for training and validation data of SC dataset obtained with different dimensionality reduction methods: (**a**) PCA, (**b**) Segmented PCA, (**c**) MNF, (**d**) Segmented MNF, (**e**) Bg-MNF, (**f**) NMF, (**g**) proposed method (SNC).

3.5.2. Results for the PU

The results of the experiments as well as the impacts of the various resampling methods are presented in Table 11. In addition to this, we employed a variety of methods for dimensionality reduction; the analysis is presented in Table 12 alongside the findings obtained using the suggested approach. Table 13 provides a concise summary of the results of the experiments conducted on the state-of-the-art techniques that were evaluated using the PU dataset. The OA of the SNC model is 99.98%. Accuracy and loss curves illustrating the results of categorization are presented in Figures 7 and 8, respectively, for each of the comparing techniques. The accuracy and loss curves were generated using both training data and validation samples. The best possible optimal performance in terms of accuracy and loss is depicted in Figures 7g and 8g, respectively. Table 14 demonstrates how the overall accuracy and the amount of time required for training can be affected by the size of the spatial window.

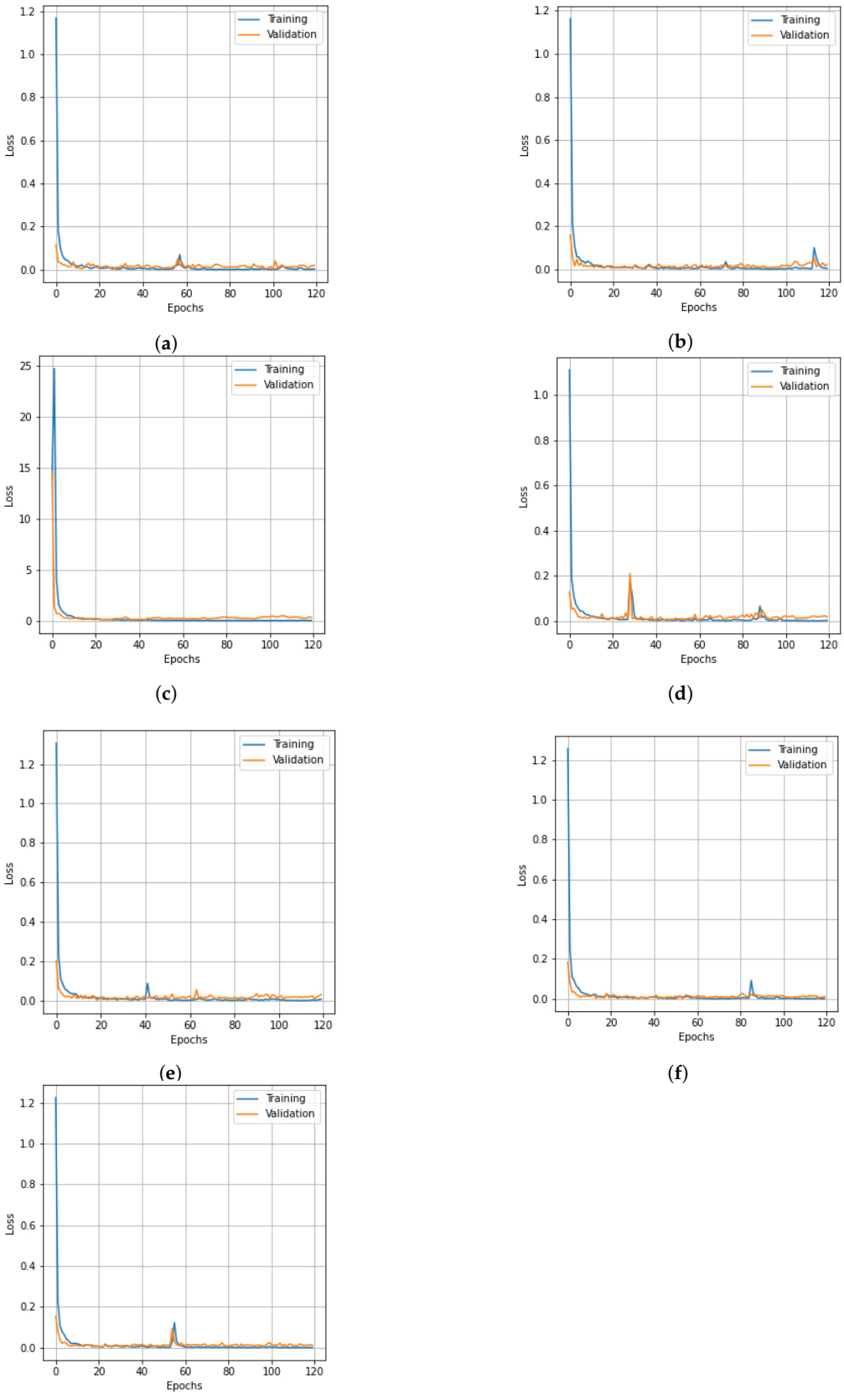


Figure 6. Loss curve for training and validation data of SC dataset obtained from different dimensionality reduction methods: (**a**) PCA, (**b**) Segmented PCA, (**c**) MNF, (**d**) Segmented MNF, (**e**) Bg-MNF, (**f**) NMF, (**g**) proposed method (SNC).

Table 7. Classification accuracy (%) for the SC dataset using different resampling techniques with the proposed SNC model.

Classes	Random O.S.	Random U.S.	Near-Miss U.S.	SMOTE O.S.	Random U.S. & O.S.	Random U.S. & SMOTE	Proposed
1	100.0	99.59	98.77	100.0	100.0	99.70	**100.0**
2	99.29	99.72	99.86	100.0	99.29	99.92	**100.0**
3	9970	100.0	100.0	99.51	99.70	99.85	**100.0**
4	99.81	99.59	99.72	99.88	99.81	99.85	**100.0**
5	100.0	99.86	99.72	99.77	100.0	99.96	**99.83**
6	99.88	100.0	100.0	100.0	99.88	99.88	**100.0**
7	100.0	99.72	99.72	99.93	100.0	99.96	**100.0**
8	99.96	99.72	99.59	99.96	99.96	99.92	**100.0**
9	100.0	100.0	100.0	99.95	100.0	99.96	**100.0**
10	99.92	97.95	99.04	99.77	99.92	99.88	**100.0**
11	99.96	99.72	99.45	99.88	99.96	100.0	**99.76**
12	99.96	100.0	100.0	99.92	99.96	99.96	**100.0**
13	99.92	99.31	99.45	100.0	99.92	99.81	**100.0**
14	99.88	99.31	99.45	100.0	99.88	99.96	**100.0**
15	99.81	99.86	99.86	99.84	99.81	99.85	**100.0**
16	100.0	100.0	100.0	100.0	100.0	100.0	**100.0**
OA	99.88	99.64	99.66	99.90	99.87	99.90	**99.98**
Kappa	99.87	99.62	99.59	99.91	99.88	99.90	**99.98**
AA	99.88	99.65	99.61	99.90	99.88	99.90	**99.97**

Table 8. Effects of different dimensionality reduction methods on the proposed deep learning model.

Dimensionality Reduction Methods	Before Re-Sampling						After Re-Sampling					
	10% Training Data			20% Training Data			10% Training Data			20% Training Data		
	OA	Kappa	AA	OA	Kappa	AA	OA	Kappa	AA	OA	Kappa	AA
PCA	99.14	99.11	99.02	99.49	99.44	99.27	99.53	99.43	99.51	99.93	99.89	99.92
Segmented PCA	99.30	99.24	99.27	99.54	99.49	99.31	99.52	99.49	99.55	99.88	99.85	99.86
MNF	99.32	99.21	99.22	99.89	99.87	99.84	99.49	99.42	99.48	99.42	99.40	99.39
Segmented MNF	99.34	99.29	99.23	99.31	99.24	99.61	99.58	99.55	99.56	99.91	99.90	99.91
Bg-MNF	99.23	99.20	99.26	99.84	99.82	99.69	99.53	99.39	99.51	99.90	99.83	99.88
NMF	99.42	99.35	99.38	99.49	99.43	99.81	99.60	99.52	99.65	99.93	99.91	99.92
Proposed (SNC)	**99.44**	**99.39**	**99.45**	**99.76**	**99.74**	**99.71**	**99.61**	**99.58**	**99.6**	**99.98**	**99.98**	**99.97**

Table 9. Effects of different window sizes on the overall accuracy and training time for the SC dataset.

	Window Size (Spatial)						
#	17 × 17	19 × 19	21 × 21	23 × 23	**25 × 25**	27 × 27	29 × 29
Accuracy	99.32	99.69	99.86	99.74	**99.98**	99.97	99.913
Time (s)	44.93	46.39	60.21	82.63	**83.91**	87.214	92.15

Table 10. Classification accuracy (in percentage) of the proposed model compared to state-of-the-art methods on the SC dataset.

Comparison Method	10% Training Data						20% Training Data					
	OA	Kappa	AA	Precision	Recall	f-1 Score	OA	Kappa	AA	Precision	Recall	f-1 Score
SVM	87.76	87.40	87.64	89.56	90.21	89.88	93.56	92.96	93.26	93.56	92.96	93.34
2D CNN	96.46	96.04	95.88	98.45	97.88	98.16	98.89	98.85	98.87	98.89	98.85	99.54
3D CNN	99.12	99.02	99.14	99.56	99.34	99.45	99.63	99.64	99.63	99.63	99.64	99.23
Fast 3D CNN	99.17	99.02	99.13	99.62	99.59	99.6	99.77	99.81	99.79	99.77	99.81	99.93
HybridSN	99.26	99.21	99.22	99.66	99.65	99.65	99.88	99.91	99.89	99.88	99.91	99.97

Table 10. *Cont.*

Comparison Method	10% Training Data						20% Training Data					
	OA	Kappa	AA	Precision	Recall	f-1 Score	OA	Kappa	AA	Precision	Recall	f-1 Score
Spectral-NET	99.35	99.27	99.34	99.77	99.76	97.78	99.69	99.55	99.64	100.0	99.93	99.97
Hybrid 3D-2D CNN	99.49	99.41	99.42	100.0	99.94	99.97	99.77	99.68	99.75	100.0	100.0	100.0
MBDA	99.42	99.36	99.39	100.0	99.93	99.96	99.62	99.54	99.60	100.0	99.97	100.0
Proposed (SNC)	**99.61**	**99.58**	**99.60**	**100.0**	100.0	**100.0**	**99.98**	**99.98**	**99.97**	**100.0**	**100.0**	100.0

Table 11. Classification accuracy (%) for the PU Dataset using different resampling techniques with the proposed SNC model.

Classes	Random O.S.	Random U.S.	Near-Miss U.S.	SMOTE O.S.	Random U.S. & O.S.	Random U.S. & SMOTE	Proposed
1	99.08	99.83	99.79	100.0	100	99.13	100.0
2	99.87	99.98	99.91	99.89	99.97	99.21	100.0
3	100.0	99.84	99.42	99.92	99.94	98.85	99.92
4	99.58	97.53	98.61	99.92	99.88	99.73	99.87
5	97.36	99.82	99.5	99.97	99.81	99.15	100.0
6	99.60	100.0	100.0	99.82	99.95	99.32	99.89
7	99.62	99.92	99.95	99.97	99.72	99.41	99.95
8	99.74	98.58	99.08	99.03	99.83	99.39	99.79
9	100.0	99.67	99.27	99.97	99.76	99.62	99.87
OA	99.48	99.52	99.56	99.85	99.89	99.38	99.94
Kappa	99.43	99.42	99.51	99.79	99.86	99.29	99.92
AA	99.42	99.46	99.50	99.83	99.87	99.31	99.92

Table 12. Effects of different dimensionality reduction methods on the proposed deep learning model for PU dataset.

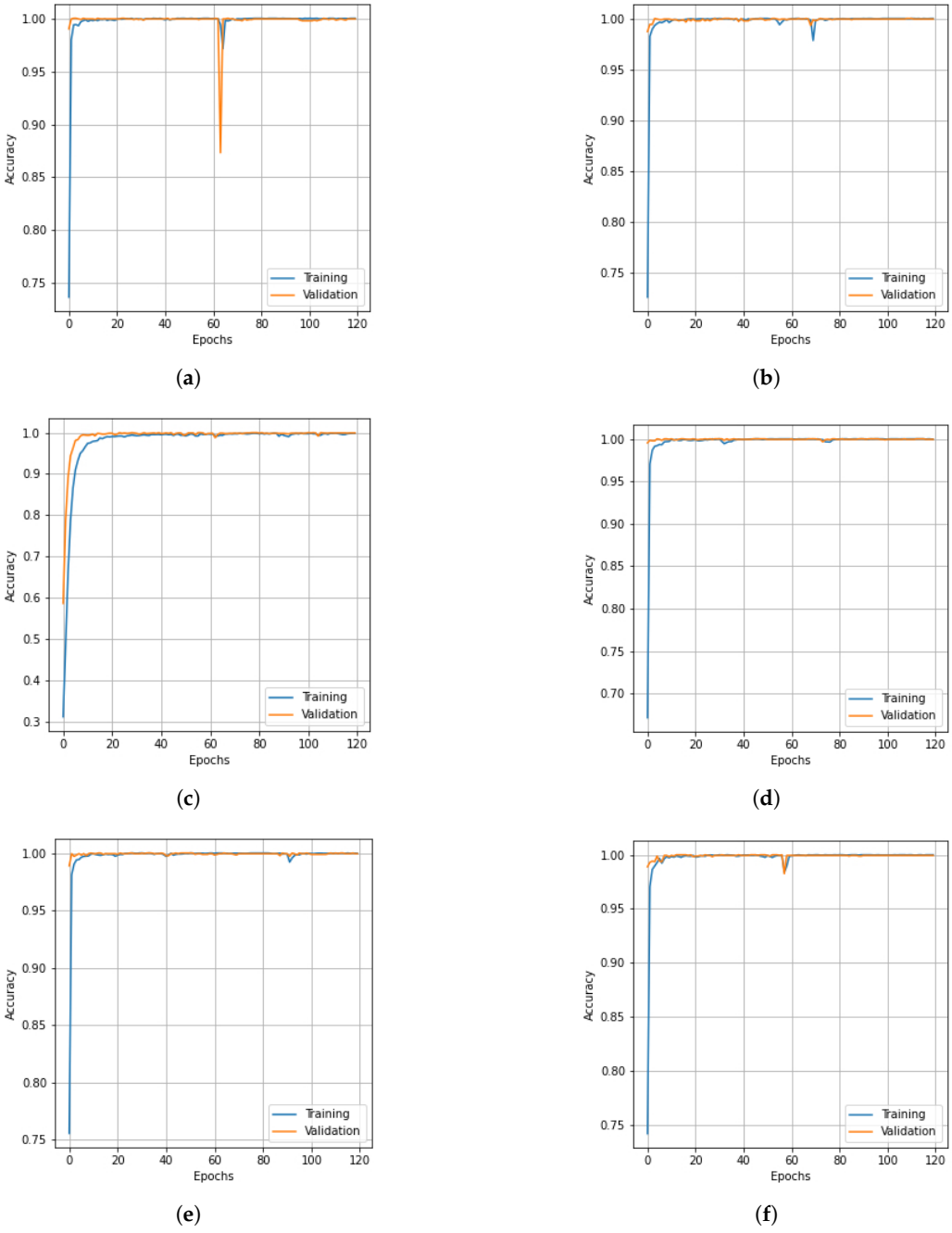
Dimensionality Reduction Methods	Before Re-Sampling						After Re-Sampling					
	10% Training Data			20% Training Data			10% Training Data			20% Training Data		
	OA	Kappa	AA	OA	Kappa	AA	OA	Kappa	AA	OA	Kappa	AA
PCA	98.65	98.61	98.63	99.33	99.32	98.35	99.54	99.48	99.52	99.9	99.89	99.89
Segmented PCA	99.11	98.96	99.04	99.37	99.32	99.33	99.53	99.52	99.51	99.91	99.90	99.91
MNF	99.21	99.13	99.19	99.44	99.41	99.42	99.32	99.26	99.25	99.60	99.55	99.57
Segmented MNF	99.26	99.22	99.25	99.5	99.43	99.47	99.68	99.57	99.63	99.92	99.9	99.91
Bg-MNF	99.23	99.14	99.20	99.47	99.44	99.43	99.57	99.55	99.55	99.91	99.89	99.91
NMF	99.24	99.22	99.19	99.56	99.49	99.52	99.65	99.54	99.59	99.92	99.90	99.92
Proposed (SNC)	99.35	99.31	99.34	99.59	99.48	98.54	99.67	99.61	99.63	99.94	99.92	99.92

Table 13. Classification accuracy (in percentage) of the proposed model compared to state-of-the-art methods on the PU dataset.

Comparison Method	10% Training Data						20% Training Data					
	OA	Kappa	AA	Precision	Recall	f-1 Score	OA	Kappa	AA	Precision	Recall	f-1 Score
SVM	85.36	84.19	86.11	89.45	97.76	87.30	88.35	86.98	92.98	88.40	87.54	88.11
2D CNN	95.43	95.24	95.17	96.32	98.02	97.34	96.78	96.43	98.75	98.14	97.46	97.20
3D CNN	99.06	99.01	99.05	99.10	99.21	99.14	99.24	99.21	99.17	99.70	99.79	99.74
Fast 3D CNN	99.08	99.04	98.99	99.3	99.43	99.36	99.31	99.26	99.29	99.84	99.89	99.86
Hybrid SN	99.12	98.97	99.09	99.42	99.29	99.35	99.62	99.54	99.58	100.0	100.0	100.0
Spectral-NET	99.36	99.31	99.37	100.0	99.87	99.93	99.79	99.65	99.74	100.0	100.0	100.0
Hybrid 3D-2D CNN	99.19	99.03	99.14	99.78	99.65	99.71	99.53	99.45	99.51	100.0	100.0	100.0
MBDA	99.21	99.13	99.17	99.83	99.87	99.92	99.61	99.57	99.59	99.78	99.89	99.85
Proposed (SNC)	**99.67**	**99.61**	**99.63**	**100.0**	**100.0**	**100.0**	**99.94**	**99.92**	**99.92**	**100.0**	**100.0**	**100.0**

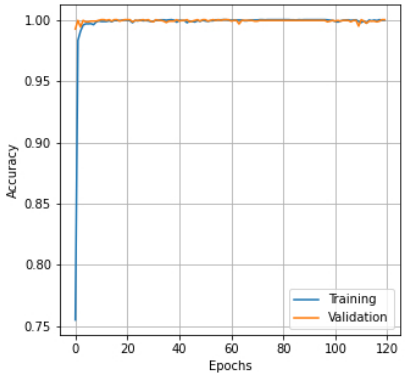
Table 14. Effects of different window sizes on the proposed method's overall accuracy and training time for the PU dataset.

	Window Size (Spatial)						
#	17 × 17	19 × 19	21 × 21	23 × 23	**25 × 25**	27 × 27	29 × 29
Accuracy	99.35	99.58	99.53	99.80	**99.94**	99.91	99.91
Time (s)	42.79	45.23	62.83	80.48	**82.81**	84.58	89.75



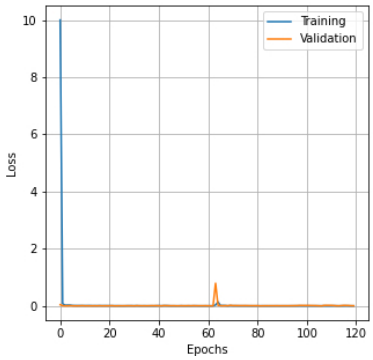
(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 7. *Cont.*

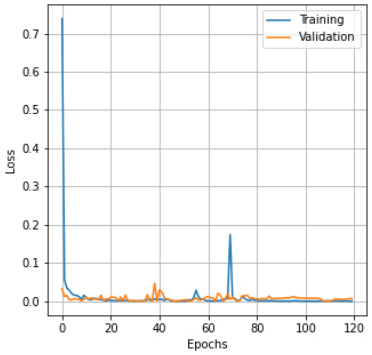


(**g**)

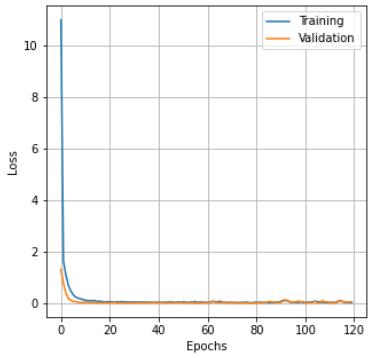
Figure 7. Accuracy curves for training and validation data of the PU dataset obtain obtained with different dimensionality reduction methods: (**a**) PCA, (**b**) Segmented PCA, (**c**) MNF, (**d**) Segmented MNF, (**e**) Bg-MNF, (**f**) NMF, (**g**) proposed method (SNC).



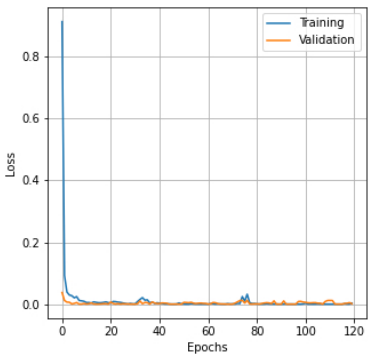
(**a**)



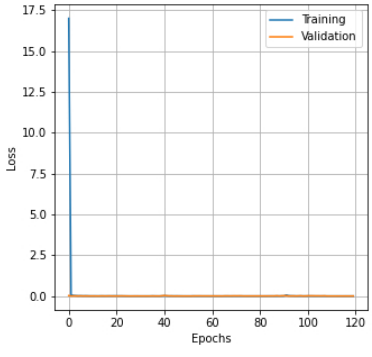
(**b**)



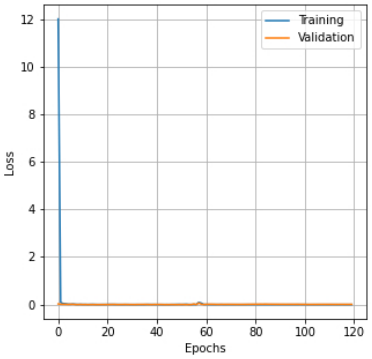
(**c**)



(**d**)

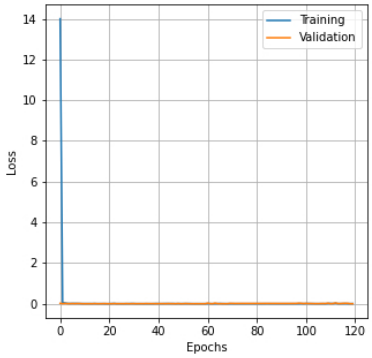


(**e**)



(**f**)

Figure 8. *Cont.*



(**g**)

Figure 8. Loss curves for training and validation data of the PU dataset obtained with different dimensionality reduction methods. (**a**) PCA, (**b**) Segmented PCA, (**c**) MNF, (**d**) Segmented MNF, (**e**) Bg-MNF, (**f**) NMF, (**g**) proposed method (SNC).

3.5.3. Results for the KSC

As in the earlier scenes, we only modified a few parameters from the deep learning model discussed in the experimental configuration. Table 15 presents the experimental results of all the methods based on the KSC dataset without resampling and after data resampling. The impacts of different dimensionality reduction techniques and window sizes of cube patches are shown in Table 16 and Table 17, respectively. The experimental findings of the evaluations of the state-of-the-art methods on the KSC dataset are summarized in Table 18. The OA, Kappa, and AA of the SNC model are 99.46%, 99.41%, and 99.43% respectively. The proposed SNC model outperforms the other methods. The graphs in Figures 9 and 10 demonstrate the correlation between the loss value and classification accuracy of the KSC dataset during the training iteration epochs. The curves reveal that the loss values for all three spatial strategies stabilize after roughly 120 training iteration epochs, and this trend is mirrored in the accuracy variation over the training iteration epochs.

Table 15. Classification accuracy (%) for the KSC dataset using different resampling techniques with the proposed SNC model.

Classes	Random O.S.	Random U.S.	Near-Miss U.S.	SMOTE O.S.	Random U.S. & O.S.	Random U.S. & SMOTE	Proposed
1	100.0	100.0	100.0	100.0	100.0	100.0	100.0
2	96.91	99.69	95.88	98.75	99.38	99.69	99.07
3	94.63	96.88	97.07	96.26	98.44	100.0	100.0
4	100.0	89.72	99.50	96.26	92.21	97.20	97.51
5	98.45	99.44	100.0	99.69	100.0	99.38	99.38
6	92.90	97.81	95.08	97.2	96.26	98.44	99.38
7	97.62	100.0	97.62	99.69	99.69	99.37	99.69
8	100.0	99.42	100.0	99.69	96.25	99.69	100.0
9	99.72	100.0	100.0	99.38	100.0	99.38	100.0
10	100.0	99.69	97.52	99.69	99.07	99.69	99.69
11	98.81	100.0	100.0	99.38	99.38	99.38	99.07
12	100.0	100.0	100.0	99.69	100.0	99.38	100.0
13	99.05	100.0	100.0	100.0	100.0	98.75	99.38
OA	98.33	98.76	98.59	98.91	98.54	98.28	99.46
Kappa	98.31	98.71	98.55	98.85	98.47	98.21	99.41
AA	98.31	98.67	98.56	98.83	98.51	99.23	99.43

The results of the experiment showcased in Figure 11 aim to evaluate the performance of the proposed model in comparison to different state-of-the-art methods. The experiment demonstrates the superiority of our model, as reflected by the accuracy and loss curves, reaffirming its effectiveness in achieving improved results. The findings from this experi-

ment highlight the potential of our approach and its ability to outperform existing methods in terms of accuracy and convergence rate.

Table 16. Effects of different dimensionality reduction methods on the proposed deep learning model for the KSC dataset.

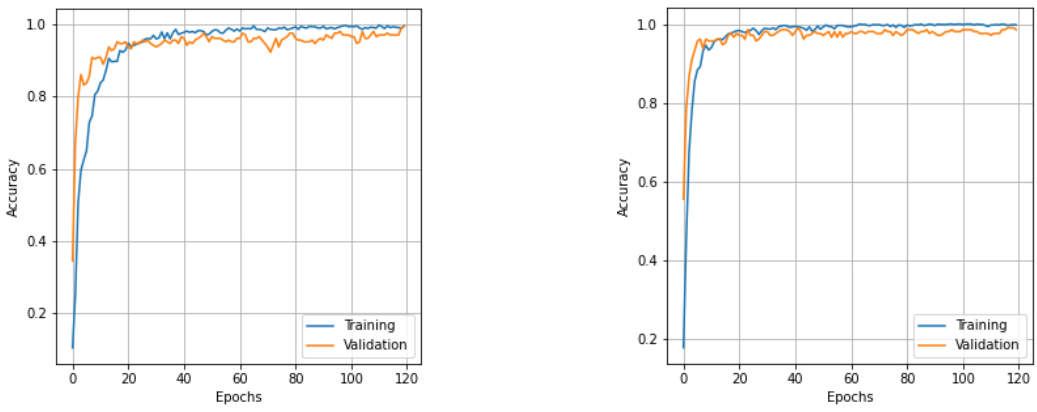
Dimensionality Reduction Methods	Before Re-Sampling						After Re-Sampling					
	10% Training Data			20% Training Data			10% Training Data			20% Training Data		
	OA	Kappa	AA	OA	Kappa	AA	OA	Kappa	AA	OA	Kappa	AA
PCA	96.16	96.1	96.14	98.33	98.26	96.32	99.12	98.94	99.08	99.25	99.18	99.23
Segmented PCA	98.31	98.28	98.27	98.63	98.58	98.56	99.25	99.18	99.23	98.81	98.74	98.73
MNF	98.59	98.51	98.53	98.92	98.88	98.88	98.74	98.66	98.67	98.61	98.52	98.56
Segmented MNF	99.12	99.05	99.08	99.26	99.24	99.23	98.98	98.89	98.93	99.29	99.22	99.23
Bg-MNF	99.17	99.12	99.16	99.15	99.08	99.11	99.21	99.18	99.17	99.37	99.31	99.35
NMF	99.16	99.12	99.11	99.19	99.13	99.15	99.11	98.97	99.1	99.23	99.19	99.21
Proposed (SNC)	99.21	99.16	99.17	99.31	99.22	99.25	99.28	99.23	99.26	99.46	99.41	99.43

Table 17. Effects of different window sizes on overall accuracy and training time for the KSC dataset.

	Window Size (Spatial)						
#	17 ×17	19 × 19	21 × 21	23 × 23	**25 × 25**	27 × 27	29 × 29
Accuracy	98.52	98.96	99.34	99.27	**99.46**	99.12	99.31
Time (s)	24.38	26.88	27.74	30.49	**32.26**	35.34	37.29

Table 18. Classification accuracy (in percentage) of the proposed model compared to state-of-the-art methods on the KSC dataset.

Comparison Method	10% Training Data						20% Training Data					
	OA	Kappa	AA	Precision	Recall	f-1 Score	OA	Kappa	AA	Precision	Recall	f-1 Score
SVM	86.54	85.31	86.33	90.46	89.63	90.04	87.12	86.83	90.35	90.46	91.22	90.84
2D CNN	94.15	94.14	94.26	96.35	98.11	97.22	97.52	96.82	97.51	98.64	98.59	98.61
3D CNN	98.12	98.04	98.12	99.13	0.12	0.24	98.24	98.19	98.16	99.11	99.13	99.12
Fast 3D CNN	98.40	98.21	98.37	99.18	99.12	99.15	98.89	98.73	98.73	99.32	99.3	99.31
Hybrid SN	98.50	98.48	98.59	99.21	99.16	99.18	98.93	98.99	98.52	99.39	99.37	99.38
Spectral-NET	98.82	98.76	98.83	99.23	99.21	99.22	98.98	98.82	98.99	99.49	99.46	99.47
Hybrid 3D-2D CNN	99.14	99.11	99.12	99.58	99.56	99.57	99.16	99.21	99.13	99.59	99.61	99.60
MBDA	99.16	99.08	99.11	99.83	99.61	99.82	99.25	99.22	99.23	99.83	99.89	99.78
Proposed (SNC)	**99.28**	**99.23**	**99.26**	**99.76**	99.75	99.75	**99.46**	**99.41**	**99.43**	**100.0**	**100.0**	99.00



(**a**) (**b**)

Figure 9. *Cont.*

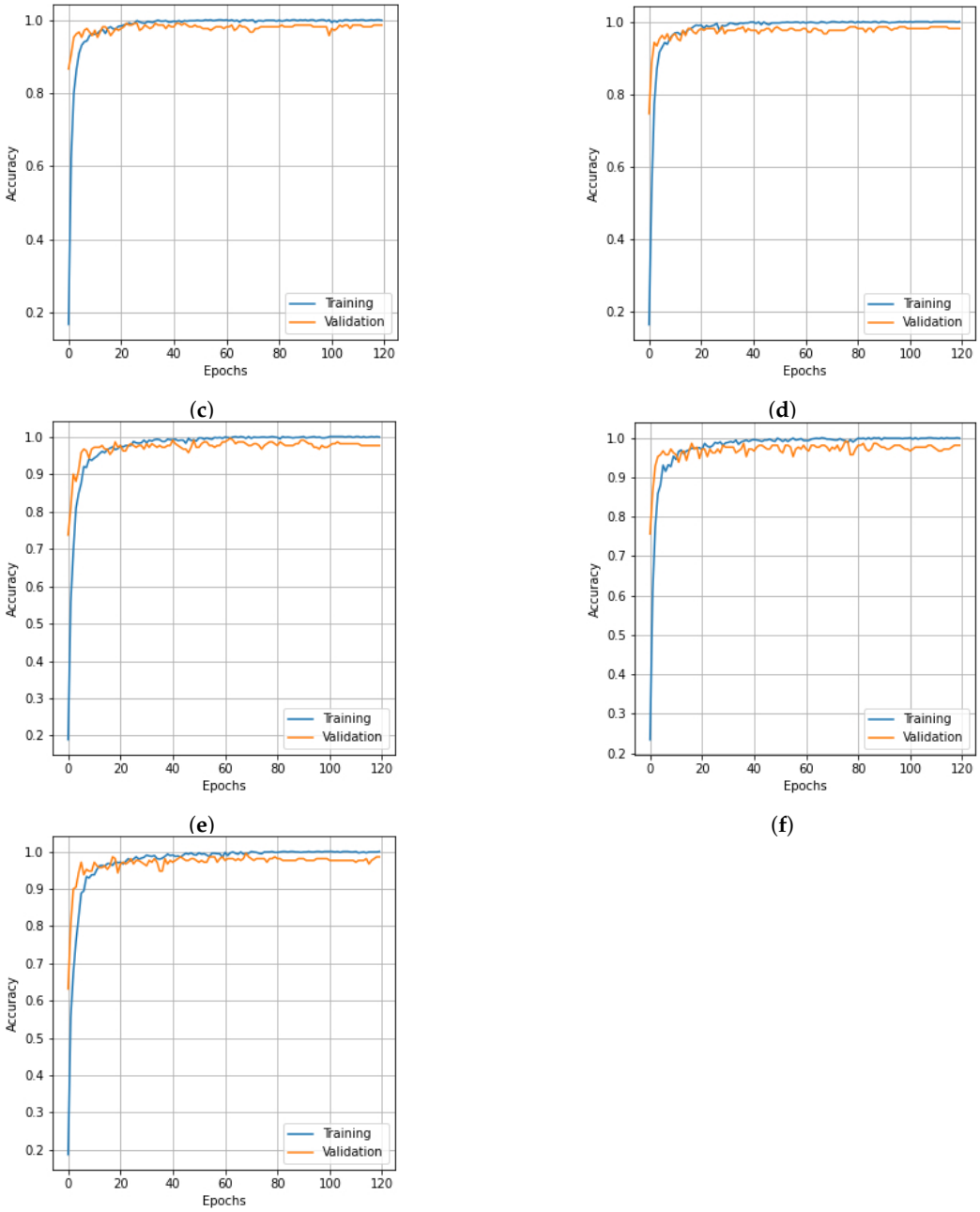


Figure 9. Accuracy curves for training and validation data of KSC dataset obtained with different dimensionality reduction methods: (**a**) PCA, (**b**) Segmented PCA, (**c**) MNF, (**d**) Segmented MNF, (**e**) Bg-MNF, (**f**) NMF, (**g**) proposed method (SNC).

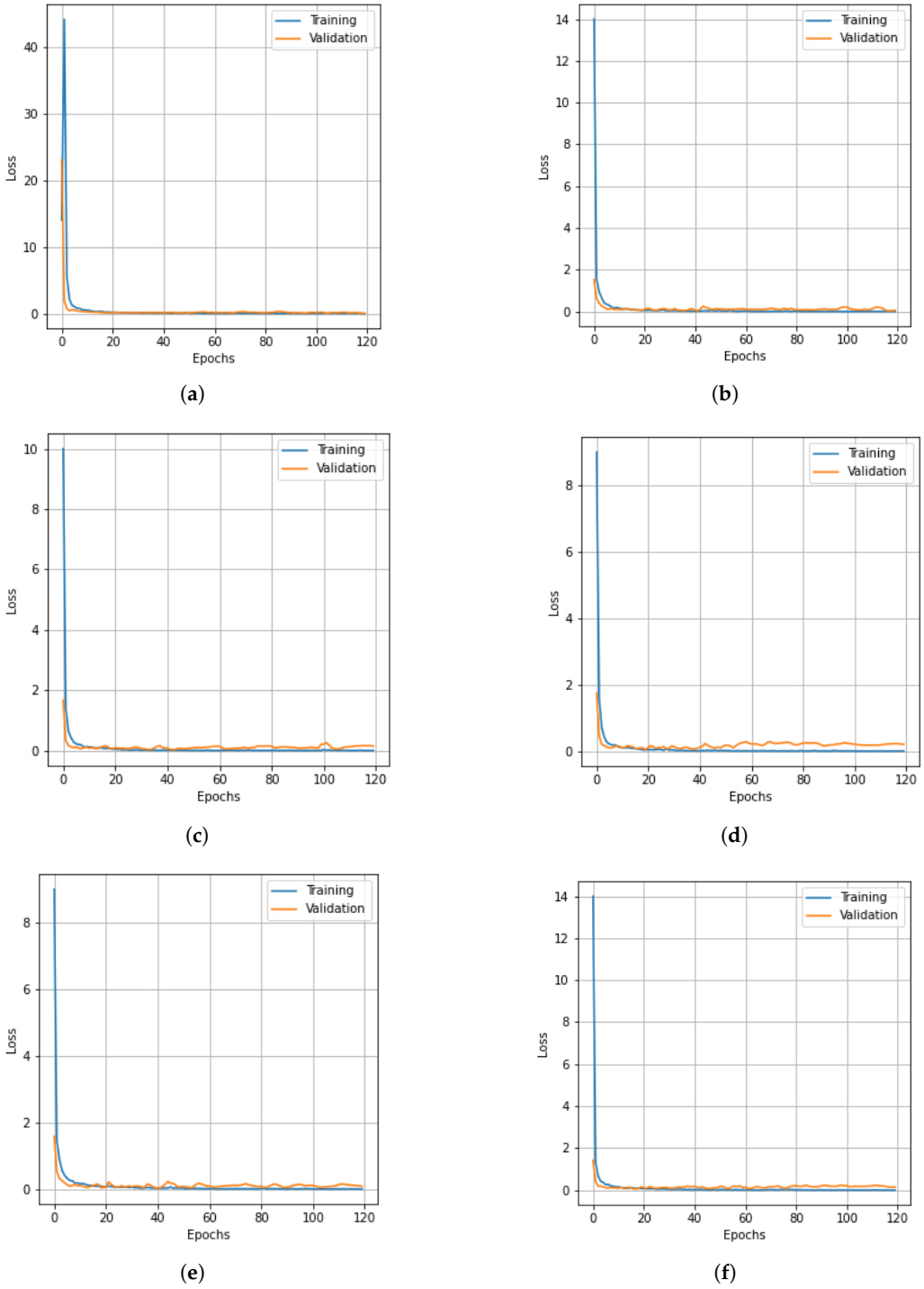
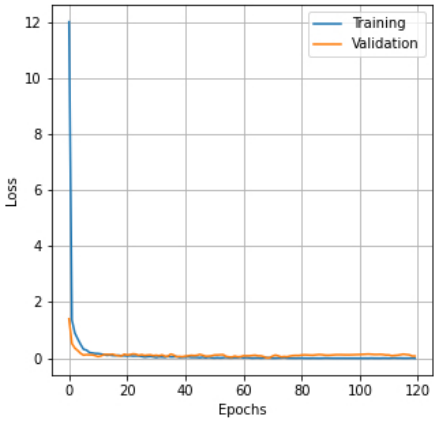
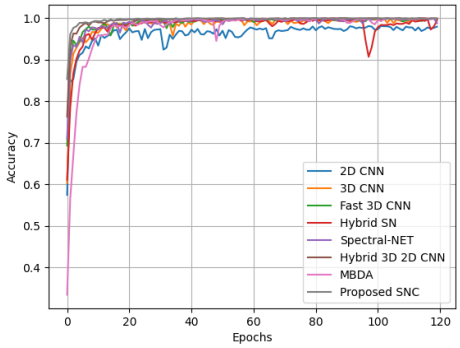


Figure 10. *Cont.*

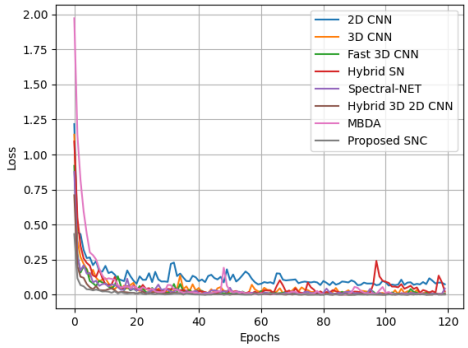


(**g**)

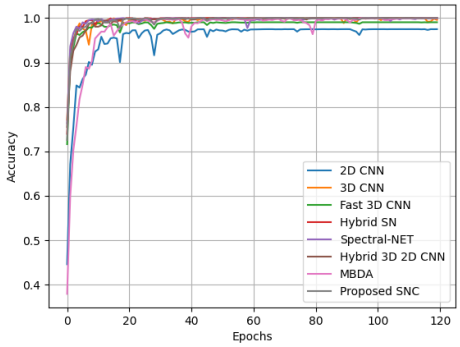
Figure 10. Loss curves for training and validation data of the KSC dataset obtain with different dimensionality reduction methods: (**a**) PCA, (**b**) Segmented PCA, (**c**) MNF, (**d**) Segmented MNF, (**e**) Bg-MNF, (**f**) NMF, (**g**) proposed method (SNC).



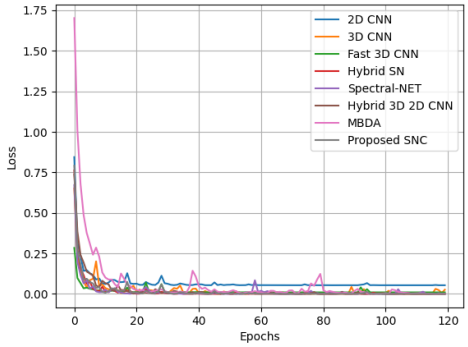
(**a**)



(**b**)

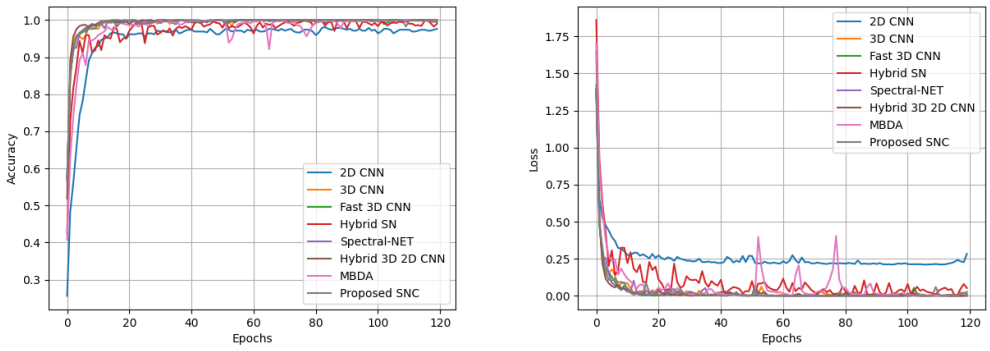


(**c**)



(**d**)

Figure 11. *Cont.*



(**e**) (**f**)

Figure 11. Accuracy and loss curves for training epochs obtained from state-of-the-art methods. Salinas dataset: (**a**) accuracy and (**b**) loss curves; Pavia University dataset: (**c**) accuracy and (**d**) loss curves; KSC dataset: (**e**) accuracy and (**f**) loss curves.

4. Conclusions and Future Work

In this article, we propose a hybrid deep learning-based model for the extraction of features and recognition of objects from unbalanced Hyperspectral Imaging (HSI) data. The suggested model uses a resampling method to address the class imbalance problem and maintain the equilibrium of the model's performance metrics. The proposed approach considers the overall properties of the data for feature extraction, using correlation as the preferred metric for measuring mutual information between HSI bands. The presented method facilitates a new form of subgrouping-based dimensionality reduction of the original data, thereby lowering the processing time required for complete subgrouping. Additionally, the study introduces a 3D-CNN and 2D-CNN hybrid model for object classification in HSI which combines spatial and spectral characteristics to enhance classification accuracy. The incorporation of both 3D and 2D convolutions makes the hybrid model more computationally efficient, requiring fewer learning parameters than 3D-CNN alone. Together, the presented approaches (SNC model) perform better than the baselines in terms of error rate, which is a common metric for evaluating per-pixel object classification performance.

Future research in this field could focus on incorporating past domain information into the proposed deep learning model. Moreover, transfer learning approaches could be investigated to address the limitations of the current deep learning model, which requires significantly more data for training than alternatives built using more conventional machine learning techniques. Overall, the suggested hybrid deep learning-based model holds great potential for improving the recognition of objects from unbalanced HSI data, and further research in this area could lead to significant advancements in this field.

Author Contributions: Conceptualization, M.T.I. and M.R.I.; methodology, M.T.I. and M.R.I.; software, M.T.I. and M.R.I.; validation, M.T.I., M.R.I. and M.P.U.; formal analysis, M.R.I., M.P.U. and A.U.; investigation, M.P.U. and A.U.; resources, M.T.I., M.R.I. and M.P.U.; data curation, M.T.I. and M.R.I.; writing—original draft preparation, M.T.I. and M.R.I.; writing—review and editing, M.P.U. and A.U.; visualization, M.T.I. and M.R.I.; supervision, M.R.I., M.P.U. and A.U.; funding acquisition, M.P.U. and A.U. All authors have read and agreed to the published version of this manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data experimented on in this study are available in [45].

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

References

1. Zhao, J.L.; Lerner, A.; Shu, Z.; Gao, X.J.; Zee, C.S. Imaging spectrum of neurocysticercosis. *Radiol. Infect. Dis.* **2015**, *1*, 94–102. [CrossRef]
2. Mallapragada, S.; Wong, M.; Hung, C.C. Dimensionality reduction of hyperspectral images for classification. In Proceedings of the Ninth International Conference on Information, Tokyo, Japan, 9–12 October 2018; Volume 1.
3. Yuen, P.W.; Richardson, M. An introduction to hyperspectral imaging and its application for security, surveillance and target acquisition. *Imaging Sci. J.* **2010**, *58*, 241–253. [CrossRef]
4. Lv, W.; Wang, X. Overview of hyperspectral image classification. *J. Sens.* **2020**, *2020*. [CrossRef]
5. Uddin, M.P.; Mamun, M.A.; Hossain, M.A. PCA-based feature reduction for hyperspectral remote sensing image classification. *IETE Tech. Rev.* **2021**, *38*, 377–396. [CrossRef]
6. Jayaprakash, C.; Damodaran, B.B.; Sowmya, V.; Soman, K. Dimensionality reduction of hyperspectral images for classification using randomized independent component analysis. In Proceedings of the 2018 5th IEEE International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 22–23 February 2018; pp. 492–496.
7. Boori, M.S.; Paringer, R.A.; Choudhary, K.; Kupriyanov, A.V. Comparison of hyperspectral and multi-spectral imagery to building a spectral library and land cover classification performanc. *Comput. Opt.* **2018**, *42*, 1035–1045. [CrossRef]
8. Aparna, G.; Rachana, K.; Rikhita, K.; Phaneendra Kumar, B.L. Comparison of Feature Reduction Techniques for Change Detection in Remote Sensing. In *Evolution in Signal Processing and Telecommunication Networks, Proceedings of Sixth International Conference on Microelectronics, Electromagnetics and Telecommunications (ICMEET 2021), Bhubaneswar, India, 27–28 August 2021*; Springer: Berlin, Germnay, 2022; Volume 2, pp. 325–333.
9. Rodarmel, C.; Shan, J. Principal component analysis for hyperspectral image classification. *Surv. Land Inf. Sci.* **2002**, *62*, 115–122.
10. Luo, G.; Chen, G.; Tian, L.; Qin, K.; Qian, S.E. Minimum noise fraction versus principal component analysis as a preprocessing step for hyperspectral imagery denoising. *Can. J. Remote Sens.* **2016**, *42*, 106–116. [CrossRef]
11. Du, Q.; Zhu, W.; Yang, H.; Fowler, J.E. Segmented principal component analysis for parallel compression of hyperspectral imagery. *IEEE Geosci. Remote Sens. Lett.* **2009**, *6*, 713–717.
12. Lixin, G.; Weixin, X.; Jihong, P. Segmented minimum noise fraction transformation for efficient feature extraction of hyperspectral images. *Pattern Recognit.* **2015**, *48*, 3216–3226. [CrossRef]
13. Tsuge, S.; Shishibori, M.; Kuroiwa, S.; Kita, K. Dimensionality reduction using non-negative matrix factorization for information retrieval. In Proceedings of the 2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236), Tucson, AZ, USA, 7–10 October 2001; Volume 2, pp. 960–965.
14. Wu, L.; Kong, C.; Hao, X.; Chen, W. A short-term load forecasting method based on GRU-CNN hybrid neural network model. *Math. Probl. Eng.* **2020**, *2020*, 1–10. [CrossRef]
15. Liu, G.; Xiao, L.; Xiong, C. Image classification with deep belief networks and improved gradient descent. In Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 21–24 July 2017; Volume 1, pp. 375–380.
16. Farrell, M.D.; Mersereau, R.M. On the impact of PCA dimension reduction for hyperspectral detection of difficult targets. *IEEE Geosci. Remote Sens. Lett.* **2005**, *2*, 192–195. [CrossRef]
17. Van Der Maaten, L.; Postma, E.; Van den Herik, J. Dimensionality reduction: A comparative. *J. Mach. Learn. Res.* **2009**, *10*, 13.
18. Du, Q.; Chang, C.I. Segmented PCA-based compression for hyperspectral image analysis. In *Proceedings of the Chemical and Biological Standoff Detection*; SPIE: Bellingham, WA, USA, 2004; Volume 5268, pp. 274–281.
19. Chen, G.; Qian, S.E. Evaluation and comparison of dimensionality reduction methods and band selection. *Can. J. Remote Sens.* **2008**, *34*, 26–36. [CrossRef]
20. Zhang, Z.Y. Nonnegative matrix factorization: Models, algorithms and applications. *Data Mining Found. Intell. Paradig.* **2012**, *2*, 99–134.
21. Kumar, N.; Uppala, P.; Duddu, K.; Sreedhar, H.; Varma, V.; Guzman, G.; Walsh, M.; Sethi, A. Hyperspectral tissue image segmentation using semi-supervised NMF and hierarchical clustering. *IEEE Trans. Med Imaging* **2018**, *38*, 1304–1313. [CrossRef]
22. Harikiran, J.J.H. Hyperspectral image classification using support vector machines. *IAES Int. J. Artif. Intell.* **2020**, *9*, 684. [CrossRef]
23. Petersson, H.; Gustafsson, D.; Bergstrom, D. Hyperspectral image analysis using deep learning—A review. In Proceedings of the 2016 IEEE Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA), Oulu, Finland, 12–15 December 2016; pp. 1–6.
24. Rissati, J.; Molina, P.; Anjos, C. Hyperspectral image classification using random forest and deep learning algorithms. In Proceedings of the 2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS), Santiago, Chile, 22–26 March 2020; p. 132.
25. Vaddi, R.; Manoharan, P. Hyperspectral image classification using CNN with spectral and spatial features integration. *Infrared Phys. Technol.* **2020**, *107*, 103296. [CrossRef]
26. Medus, L.D.; Saban, M.; Frances-Villora, J.V.; Bataller-Mompean, M.; Rosado-Muñoz, A. Hyperspectral image classification using CNN: Application to industrial food packaging. *Food Control* **2021**, *125*, 107962. [CrossRef]
27. Li, Y.; Zhang, H.; Shen, Q. Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]

28. Ahmad, M.; Khan, A.M.; Mazzara, M.; Distefano, S.; Ali, M.; Sarfraz, M.S. A fast and compact 3-D CNN for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *19*, 1–5. [CrossRef]
29. Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 277–281. [CrossRef]
30. Kotsiantis, S.; Kanellopoulos, D.; Pintelas, P. Handling imbalanced datasets: A review. *GESTS Int. Trans. Comput. Sci. Eng.* **2006**, *30*, 25–36.
31. Sun, Y.; Wong, A.K.; Kamel, M.S. Classification of imbalanced data: A review. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23*, 687–719. [CrossRef]
32. Chaubey, Y.P. Resampling methods: A practical guide to data analysis. *Technometrics* **2000**, *42*, 311. [CrossRef]
33. Somasundaram, A.; Reddy, U.S. Data imbalance: Effects and solutions for classification of large and highly imbalanced data. In Proceedings of the 1st International Conference on Research in Engineering, Computers and Technology (ICRECT 2016), Tiruchirappalli, India, 8–10 September 2016.
34. Liu, B.; Tsoumakas, G. Dealing with class imbalance in classifier chains via random undersampling. *Knowl.-Based Syst.* **2020**, *192*, 105292. [CrossRef]
35. Zheng, Z.; Cai, Y.; Li, Y. Oversampling method for imbalanced classification. *Comput. Inform.* **2015**, *34*, 1017–1037.
36. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
37. Alsowail, R.A. An Insider Threat Detection Model Using One-Hot Encoding and Near-Miss Under-Sampling Techniques. In Proceedings of the International Joint Conference on Advances in Computational Intelligence: IJCACI 2021, online, 23–24 October 2021; Springer: Berlin, Germany, 2022; pp. 183–196.
38. Borgognone, M.G.; Bussi, J.; Hough, G. Principal component analysis in sensory analysis: Covariance or correlation matrix? *Food Qual. Prefer.* **2001**, *12*, 323–326. [CrossRef]
39. Billah, M.; Waheed, S. Minimum redundancy maximum relevance (mRMR) based feature selection from endoscopic images for automatic gastrointestinal polyp detection. *Multimed. Tools Appl.* **2020**, *79*, 23633–23643. [CrossRef]
40. Tsai, F.; Lin, E.K.; Yoshino, K. Spectrally segmented principal component analysis of hyperspectral imagery for mapping invasive plant species. *Int. J. Remote Sens.* **2007**, *28*, 1023–1039. [CrossRef]
41. Chakraborty, T.; Trehan, U. Spectralnet: Exploring spatial-spectral waveletcnn for hyperspectral image classification. *arXiv* **2021**, arXiv:2104.00341.
42. Yin, J.; Qi, C.; Huang, W.; Chen, Q.; Qu, J. Multibranch 3d-dense attention network for hyperspectral image classification. *IEEE Access* **2022**, *10*, 71886–71898. [CrossRef]
43. Diakite, A.; Jiangsheng, G.; Xiaping, F. Hyperspectral image classification using 3D 2D CNN. *IET Image Process.* **2021**, *15*, 1083–1092. [CrossRef]
44. Mounika, K.; Aravind, K.; Yamini, M.; Navyasri, P.; Dash, S.; Suryanarayana, V. Hyperspectral image classification using SVM with PCA. In Proceedings of the 2021 6th IEEE International Conference on Signal Processing, Computing and Control (ISPCC), Solan, India, 7–9 October 2021; pp. 470–475.
45. Graña, M.; Veganzons, M.; Ayerdi, B. Hyperspectral Remote Sensing Scenes. 2021. Available online: https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes (accessed on 11 July 2023).