remote sensing

MDPI

Article

Meta-Learner Hybrid Models to Classify Hyperspectral Images

Dalal AL-Alimi [1], Mohammed A. A. Al-qaness [2,3,*], Zhihua Cai [1], Abdelghani Dahou [4], Yuxiang Shao [1] and Sakinatu Issaka [5]

1 School of Computer Science, China University of Geosciences, Wuhan 430074, China; dalal@cug.edu.cn (D.A.-A.); zhcai@cug.edu.cn (Z.C.); shaoyx@cug.edu.cn (Y.S.)
2 State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China
3 Faculty of Engineering, Sana'a University, Sana'a 12544, Yemen
4 Department of Mathematics and Computer Science, Faculty of Science and Technology, University of Ahmed DRAIA, Adrar 01000, Algeria; dahou.abdghani@univ-adrar.edu.dz
5 School of Environmental Studies, China University of Geosciences, Wuhan 430074, China; sakina@cug.edu.cn
* Correspondence: alqaness@whu.edu.cn

Abstract: Hyperspectral (HS) images are adjacent band images that are generally used in remote-sensing applications. They have numerous spatial and spectral information bands that are extremely useful for material detection in various fields. However, their high dimensionality is a big challenge that affects their overall performance. A new data normalization method was developed to enhance the variations and data distribution using the output of principal component analysis (PCA) and quantile transformation, called QPCA. This paper also proposes a novel HS images classification framework using the meta-learner technique to train multi-class and multi-size datasets by concatenating and training the hybrid and multi-size kernel of convolutional neural networks (CNN). The high-level model works to combine the output of the lower-level models and train them with the new input data, called meta-learner hybrid models (MLHM). The proposed MLHM framework with our external normalization (QPCA) improves the accuracy and outperforms other approaches using three well-known benchmark datasets. Moreover, the evaluation outcomes showed that the QPCA enhanced the framework accuracy by 13% for most models and datasets and others by more than 25%, and MLHM provided the best performance.

Keywords: meta-learner; hyperspectral image; classification; remote sensing images; hybrid model; feature fusion

Citation: AL-Alimi, D.; Al-qaness, M.A.A.; Cai, Z.; Dahou, A.; Shao, Y.; Issaka, S. Meta-Learner Hybrid Models to Classify Hyperspectral Images. *Remote Sens.* **2022**, *14*, 1038. https://doi.org/10.3390/rs14041038

Academic Editor: Saeid Homayouni

Received: 28 December 2021
Accepted: 17 February 2022
Published: 21 February 2022

1. Introduction

Imaging spectroscopy, also called hyperspectral (HS) imaging, captures electromagnetic energy produced or reflected from a single place using hundreds of small, continuous spectral bands ranging from visible to infrared wavelengths y [1–3]. The classification of HS images is very important. It plays critical roles in various applications, such as national defense [4], Earth monitoring [5], forest monitoring [6], detection [7,8], geological exploration [9], video summarization, and precision farming [10]. In general, the sample vector of an HS image is high-dimensional, and it has much information on contiguous spectral bands [11].

HS images are known to be images with a very large dimension because of the large numbers of spectral bands. Due to the large size of HS images, reducing the dimension is important. Dimensionality reduction methods, like principal component analysis (PCA), linear discriminant analysis (LDA), independent component analysis (ICA), and uniform manifold approximation and projection (UMAP) [12,13], work to transfer the original data space to a new subspace of dimensionality. The new subspace includes the most informative features and omits the redundancy features. Thus, dimensionality-reduction methods help speed up the classification process, decrease the storage space for data, and

in easily visualization of the data [13–15]. PCA is the most common method, and many studies use it as preprocessing method to decrease the dimension. [16–21]. On the other hand, these methods do not work to normalize or enhance the data distribution.

The classification of HS images received wide attention in the literature. Different approaches have been proposed using various classification methods [14,22]. Earlier, some traditional methods were applied to classify HS images. In [23], four kernel-based methods, namely, standard support vector machine (SVM), regularized radial basis function neural networks, regularized AdaBoost, and kernel Fisher discriminant analysis, were employed for HS classification. The dictionary-based sparse representation was adopted by [24] to enhance HS classification. A traditional SVM classifier was also adopted for this issue in several studies [25–27]. The K-nearest neighbor (KNN) was also employed for HS image classification in several studies [28–30].

Most recently, deep learning (DL) approaches have shown significant performance in HS images classification. One of the most efficient methods is the convolutional neural network (CNN), which achieved considerable accuracy. For example, Cao et al. [31] proposed an active deep learning model for HS images classification. The main idea of this model is to train CNN with a small number of labeled pixels and then select informative pixels actively. More so, Markov random field was applied to enhance the classification accuracy. The experimental results confirmed the significant performance of this model compared to several existing classifiers. Hang et al. [32] proposed an attention-aided CNN to classify HS images. Spectral and spatial classifications were implemented using two subnetworks, called spectral attention and spatial attention, respectively. The results of the spectral and spatial classifications were integrated using an adaptively weighted summation technique. Yu et al. [33] developed an HS classification model using a 2D-3D CNN scheme. The main idea of this scheme is by using a 2D CNN to extract features, where the 3D CNN uses a reduced kernel to exploit band co-relation data. The evaluation outcomes of the 2D-3D CNN model showed its efficiency in selecting features and improving classification accuracy. Due to the dying neurons with the Rectified Linear Unit (ReLU) activation function in the higher layers, the authors in [34] used a hybrid model of 3D-2D CNN to extract the features with non-monotonic activation function and compared with the results of ReLu. Another HS images classification model using an improved CNN model, called Dual-SCNN, was presented in [35]. In [36], the authors proposed a tighter random projection with minimal intra-class variance (TRP-MIV) to improve the class separability. TRP-MIV reduces the dimension between similar values to classify the unsupervised HS images easily. This study [37] started by spotting the edge then removing it to reduce the noise. It works to decrease the dimensionality and increases the training speed by many steps of enhancing the accuracy of CNN and maxpooling. To reduce the total parameter number of HS images in the deep networks, Paul et al. added spatial pyramid pooling (SPP) to the extracted feature maps of 3D-2D CNN [38]. Wei and Zhou introduced a deep learning model called a spatial-aware network (SANet) [39], a particular network to reduce and choose the most critical extracted feature maps. The SANet is a hierarchical network that includes three layers (F, P, R). The F layer uses eight side windows to obtain eight feature maps. The MIN pooling is used in the P layer to reduce the number of F layer feature maps. The final layer uses LDA to reduce the dimensionality and increase the classes variations. Moreover, Zhao et al. [40] used dense connection in the recursive densely connected neural network. These methods try to keep the storage space and solve the lack of samples number in HS images. On the other hand, focusing only on having enough samples does not always increase the final accuracy.

The shallow network of DL can provide more localized information, and the deep network of DL provides more abstract features than the shallow network. Therefore, the authors of [41] combined the output of these sequent layers then fed the combined produced maps into the FC layer to obtain the final output. Shallow CNN can locate small objects, and has less computational cost than deep CNN. On the other hand, deep CNN is a powerful network for extracting big objects and obtaining better features [14].

Therefore, a shallow-deep feature extraction network (SDFE) combined the output of these layers to obtain strong features for larger and small objects and a low loss for the small ones [42]. Due to very deep networks causing the overfitting problem and limited training samples, a cascade dual-scale crossover neural network (CDSCN) can extract more features by applying different spatial and spectral-size convolution kernels [43]. Also, to avoid overfitting, batch normalization (BN) and dropout layers are added between DL layers [44,45].

Additionally, using more layers helps extract more features and increase semantic values, but that causes a reduction in the spatial values and accuracy. So, many studies used ResNet [42], SeNet [46], and BN layers to relieve the vanishing gradient problem and optimize the accuracy. Furthermore, combining different paths of network outputs was used to enhance the spatial values and promote feature extraction like FPN [47], SDFE [42]. In HS images, using only the spectral features is not enough, especially if there is more than one object with identical spectral signatures. Therefore, knowing the shape and texture of an object as spatial information increases the distinction. Thus, merging the spectral and spatial information is necessary to boost the HS image classification accuracy and identify each class easily [14,48]. In many cases, using deep networks or combining many subnetworks does not improve the classification since these end-to-end approaches modify the whole produced weights in each epoch throughout the training time. Therefore, meta-learner and transfer learning methodologies were introduced to avoid altering the whole generated weighs.

Meta-learner is a way to obtain the final results from two levels of methodologies. The first level has many models that send their prediction to the second level, which stacks, combines, and trains previous level models' predictions to get optimal and accurate final results, also called stacking [49–53]. For example, Taormina et al. [54] used four pre-trained models, namely, AlexNet, SqueezeNet, ResNet18, and GoogLeNet, and retrained them with three different freezing levels and retrained from scratch to optimize the accuracy of extracting the features of the two classes of the AIDA dataset. Zhong et al. [55] transferred the pre-trained data to a multiscale spectral-spatial network (MSSN) with multi-scale CNNs to the target domain to solve samples limitation then classified the output of the four different HS classes. Liu et al. [56] used six models to analyze city traffic data, and the stacked generalization framework obtained the best results. This study [57] used random forests in the first and second levels (meta-learner) to recognize accelerometer and audio signals activities and k-fold cross-validation to prevent overfitting. This work [51] combined the predictions of parsimonious models to increase the variety and decrease the generalization error rate to find three essential points of the complete force–displacement curve of bolted components in steel structures.

Although deep learning has provided strong methodologies to improve the classification operation, more research is needed to figure out how to enhance classification and recognize the differences between the classes in the complex and heterogeneous dataset. Shallow layers provide detailed features, and the deep layers provide more semantic information [14,42,58]. On the other hand, shallow layers are unsuitable for complex data, and complex computations of deep layers encounter overfitting problems and need sufficient data. However, we can sum up the difficulties of HS image datasets as follows:

1. The HS image dataset is a high-dimensionality dataset with a massive number of bands, and each band has a different data distribution. Each dataset has a multi-class with different sample numbers.
2. Classification solely based on spectral classification seldom achieves high accuracy because of the objects' complicated spatial distribution and spectral heterogeneity [59]. Each pixel value needs to localize and classify correctly.
3. Because of the backpropagation operation, combining shallow and deep layers in one model does not provide higher accuracy, as compared to training them individually.

Therefore, this study works to solve all the above issues, in order to boost the final results of the deep and shallow networks by utilizing the meta-learner, which stops the

backpropagation to modify the generated weights of the Leve-0 models. The framework of this study works in three stages of processing. First is preprocessing, which works to enhance the data distribution and split it before feeding it into the training model. Then the Level-0 models, which are hybrid models of deep and shallow models, and multi-size kernels of 3D and 2D-CNN, which work to train the input data individually. Finally, the Level-1 model, which chooses the best-extracted weights of the previous stage, reduces the loss and overfitting and optimizes the spatial-spectral extraction simultaneously to obtain the final classification. To sum up, the main contributions of this study are:

1. To propose a novel framework that uses the meta-learner technique to train multi-class and multi-size datasets by concatenating and training the hybrid and multi-size kernel networks of CNN.
2. To provide a new normalization method, called QPCA, based on the output of PCA and quantile transformation, which redistributes the data to be more normal and have less dimensionality.
3. To present an efficient method that can extract more features from simple and complex spatial-spectral data simultaneously by combining the output of the shallow and deep networks without needing to increase the number of samples to increase the accuracy.

2. Methodology and Framework

The framework structure of this study is illustrated in Figure 1. It has three stages of processing. The first stage showcases external normalization to reduce the dimensionality and normalize the input data. The second stage is Level-0 methodologies, and it has two different types of model structure. In these two stages, the input data passes through two different types of scaling, external and internal normalization. The final stage is the Level-1 model to train the output of Level-0 and the input data and obtain the whole framework's final results.

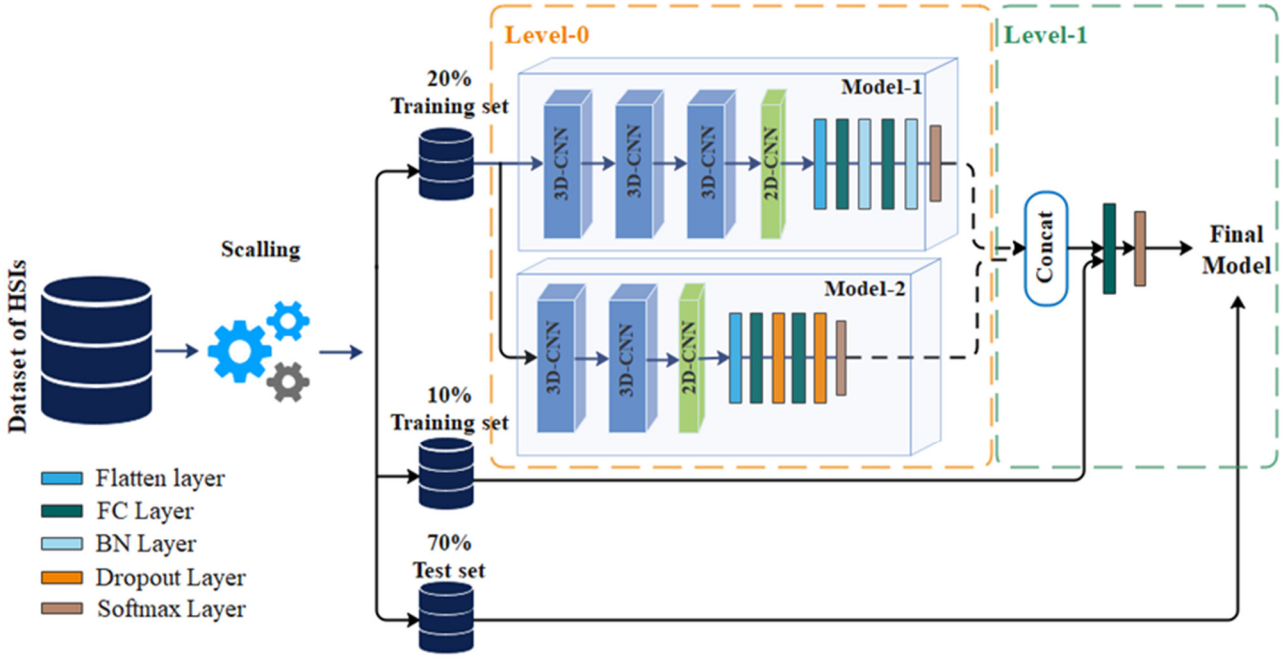


Figure 1. The framework of meta-learner hybrid models (MLHM).

2.1. Optimize the Data Distribution (QPCA, Quantile Transformation Principal Components Analysis)

One of the external normalizations in HS images is principal components analysis (PCA), which reduces the banal numbers of the input dataset and the data dimensionality itself. It is a strategy that employs linear approximation to figure out the most critical

components that help to raise the variance in the input data which helps to reduce the memory space, noise and speed up the performance.

Assume x is the original data which $= [x_1, x_2, \ldots, x_d],\ x \in \mathcal{R}^d$, d is the dimension of x. The standardization of the input data is calculated as:

$$X = (x - \mu)/\sigma(x) \tag{1}$$

where μ is the mean of x and $\sigma(x)$ is the standard deviation.

Then, to promote differences between classes, calculate the covariance matrix for X:

$$C_X = (\sum_{i=1}^{d}(X_i - \overline{X})(X_i - \overline{X})^T)/d - 1 \tag{2}$$

After obtaining the covariance matrix, compute the eigenvectors (ω) of the $C_{d\times d}$ matrix using the biggest eigenvectors k, where k is the dimensionality of the new feature subspace ($k \leq d$). ω is the weight vector in PCA, which will be multiplied by X to obtain the final new subspace of transferred features:

$$\mathcal{P}_{d\times k} = X_{d\times k} \times (\omega_{k\times 1})^T \tag{3}$$

This procedure aids in the categorization of classes and the selection of the most valuable features for classification.

PCA reduces d dimension to a new dimension k, and $\mathcal{P} = [p_1, p_2, \ldots, p_k],\ p \in \mathcal{R}^k$ is the output of PCA for x, where $k \leq d$. The next step is to change the distribution of $\mathcal{P}$ by quantile transformation (Q), which we call the output of this transformation QPCA.

$$y = G_y^{-1}(F_{\mathcal{P}}(\mathcal{P})) \tag{4}$$

where $F_{\mathcal{P}}$ represents cumulative distribution function (CDF) for the output of PCA ($\mathcal{P}$), and G_y^{-1} is the Gaussian cumulative distribution function (GCDF) in y. The GCDF is represented as follows:

$$F(p) = 1/\sigma\sqrt{2\pi}\int_{-\infty}^{+\infty}\exp\left[-(p - \mu/\sigma)^2/2\right]dp \tag{5}$$

where μ is the mean of p, σ is the standard deviation of p. Algorithm 1 shows the main steps of the QPCA processing.

Algorithm 1. The QPCA Processing.

Input: $x \in \mathcal{R}^{n\times d}$
Output: $y \in \mathcal{R}^{n\times k}$
1: Standardize the d-dimensional of $x \to X$.
2: Getting the covariance matrix of X
3: Select the top k eigenvectors ($k \leq d$) to build the weight matrix ω.
4: Transform X by multiplying it with the weight vector of k to get $\mathcal{P}:\ \mathcal{P} = X \times \omega$
5: Normalize the distribution of $\mathcal{P}$ by quantile transformation: $y = G_y^{-1}(F_{\mathcal{P}}(\mathcal{P}))$

2.2. Framework

After keeping the influential spectral bands that contain more information, the input HS image cubs become $\in \mathcal{R}^{T\times H\times k}$, where is the transformation data of QPCA, where T is the width, H is the height, and k is the number of bands. The initial layers of the Level-0 models are 3D-CNN, so we divided the data cubs into small overlapping 3D patches $\mathcal{P} \in \mathcal{R}^{S\times S\times k}$, where $S \times S$ is the window size which equals 25 for all input datasets. 3D patches of $\mathcal{P}$ are given by $(T - S + 1) \times (H - S + 1)$ then reshaped into $(S,\ S,\ k,\ 1)$.

The model-1 of Level-0 has 10 layers. It has three sequential 3D-CNN layers, one 2D-CNN layer, consecutive fully connected (FC) layers, and batch normalization (BN). The formulas for 2D-CNN and 3D-CNN layers are as follows:

$$\mathcal{Y}_{ij}^{py} = \varphi\left(\sum_{m}\sum_{a=1}^{h}\sum_{q=1}^{t} \mathcal{w}_{ijm}^{aq}\mathcal{Y}_{(i-1)m}^{(p+a)(y+q)} + \mathfrak{b}_{ij}\right) \tag{6}$$

$$\mathcal{Y}_{ij}^{pyz} = \varphi\left(\sum_{m}\sum_{r=1}^{d}\sum_{a=1}^{h}\sum_{q=1}^{t} \mathcal{w}_{ijm}^{aqr}\mathcal{Y}_{(i-1)m}^{(p+a)(y+q)(z+r)} + \mathfrak{b}_{ij}\right) \tag{7}$$

$$\varphi(p) = \max(0, p) \tag{8}$$

where (p, y, z) are the variable position in the jth feature map in the ith layer, φ is the ReLu activation function (RAF), $(t,\ h, d)$ are the kernel sizes, (a, q, r) are the kernel indexes, m is feature maps index, and $(\mathcal{w}, b)$ are the weight and bias.

The output dimensional and kernel size of the first 3D-CNN is 8, and (3, 3, 7). The second 3D-CNN is 16, and (3, 3, 5). Finally, the third 3D-CNN has 32 output dimensional and (3, 3, 3) kernel sizes. The output dimensional and kernel size of the 2D-CNN are 64, and (3, 3). Before feeding the output of the 2D-CNN into the first FC layer, the created feature maps pass through the flattening layer, and the number of units of FC layers are 256 and 128, respectively, as indicated in Table 1.

Table 1. The structure of the framework. level-1 layers for model-1 and model-2, and level-2 layers.

Level-0		Level-1
Model-1	**Model-2**	
3D-Convolutional Neural Networks (CNN) 8—(3, 3, 7)—ReLu *	3D-CNN 8—(3, 3, 7)—ReLu	–
3D-CNN 16—(3, 3, 5)—ReLu	3D-CNN 16—(3, 3, 5)—ReLu	–
3D-CNN 32—(3, 3, 3)—ReLu	–	–
2D-CNN 64—(3, 3)—ReLu	2D-CNN 64—(3, 3)—ReLu	
Flatten	Flatten	–
FC 256—ReLu **	FC 256—ReLu	Conct
BN	Dropout	–
FC 128—ReLu	FC 128—ReLu	FC 128—ReLu
BN	BN	–
SoftMax	SoftMax	SoftMax

* Dimensional—(Kernel size)—Activation Function; ** Dimensional—Activation Function.

HS image datasets are distinguished by having multi-classes, and each class has a different number of samples; some have a minimal number of samples. On the other hand, deep learning methodologies often have many weight values for tuning, making it a big challenge to enhance the accuracy of this disparity of the dataset classes. Therefore, many methods were introduced to avoid overfitting or to enhance the final accuracy during the training stage, like BN, dropout, lateral connections [14,19,42,58,60]. In this model, BN layers were used before each FC layer to normalize the results of the internal operation during the training. However, because CNN reduces the dimension of the produced feature map, zero paddings were used for the input data to keep the edge data from losing, as shown in Figure 1 and Table 1.

Model-2 of Level-0 stage is almost like the model-1 of Level-0. It has two 3D-CNN layers. The output dimensional and kernel size of the first 3D-CNN is 8, and (3, 3, 7). The

second 3D-CNN is 16, and (3, 3, 5). The third layer of this model is the 2D-CNN layer, and the output dimensional and kernel sizes of this layer are 64 and (3, 3). Also, two layers of FC with the same number units of model-1, with two dropout layers accounting for 40% dropping rate, were used after each FC layer to avoid overfitting, and finally, the SoftMax layer. This model has fewer layers and less complex computation than model-1, Figure 1, and Table 1.

The level-1 network has three layers: one is the concatenation layer to combine the output of Level-0 models, the second one is the FC layer which has 128 units, and the last layer is the SoftMax layer to get the final results. Whiles RAF was used in all layers of the two levels. In meta-leaner and during the training, the weights are given as:

$$\hat{w}^{ml} = \underset{w}{argmin} \sum_{i=1}^{\mathcal{N}} \left[y_i \sum_{m=1}^{M} w_m \hat{y}_m(x_i) \right]^2 \tag{9}$$

where $\hat{y}_m(x_i)$ is the prediction of x, using model m and M in this study = 2, and $\mathcal{N}$ is the sample number. Rather than choosing one model (M = 1), the meta-learner combines the Level-0 models with estimated optimal weights. This method leads to better predictions often. This stage usually works to minimize the loss of the output of the two levels. The loss function works as follows:

$$Loss = -\sum_{i=1}^{\mathcal{N}} y_i . log \hat{y}_i \tag{10}$$

where y_i is the real value (ground truth), $\hat{y}_i$ is the score of the level-1 network for the input data and the output of level-0 models. Algorithm 2 explains the operation of the meta-leaner and the work of the MLHM framework.

Algorithm 2. The Steps of the Proposed MLHM.

Input: Training data $\mathcal{D}_1 = \{X_i, y_i\}_{i=1}^{m} \left(X_i \in \mathbb{R}^n, y_i \in \Upsilon\right)$ and
$\mathcal{D}_2 = \left\{X_j, y_j\right\}_{j=m}^{e} \left(X_j \in \mathbb{R}^n, y_j \in \Upsilon\right)$

Output: An ensemble classifier $\hat{y}_i$

1: Step 1: Learn Level-0 classifiers
2: **for** $t \leftarrow 1$ to m **do**
3: Learn a base classifier y_t based on $\mathcal{D}_1$
4: **end for**
5: Step 2: Train the dataset of $\mathcal{D}_2$
6: Keep the weights of training stage $\{X_i', y_i\}, where\ x_i' = \{y_1(X_i), y_1(X_i), \ldots, y_m(X_i)\}$
7: Step 3: Learn the Level-1 classifier
8: **for** $j \leftarrow m\ to\ e$ **do**
9: Redistribute weights based on the new training data and the previous weights. Learn a new classifier $\hat{y}$ based on the newly constructed data set
10: **end for**
11: **Return** $\hat{y}(X) = \hat{y}(y_1(X_j), y_1(X_j), \ldots, y_e(X_j))$

3. Experiment

The experiments were executed on three widely used datasets of hyperspectral images, the Indian Pines dataset, the Pavia-University dataset, and the Kennedy Space Center.

3.1. Datasets

The first data set is the Indian Pines (IPs) dataset gathered by the AVIRIS sensor from northwest India, containing 16 classes and having 145 × 145 spatial dimensions. The spatial resolution is 20 m per pixel. It contains 220 spectral bands with a wavelength ranging from

0.4 µm to 2.5 µm. However, after removing bands covering the water absorption region, the bands have been reduced to 200, as shown in Figure 2.

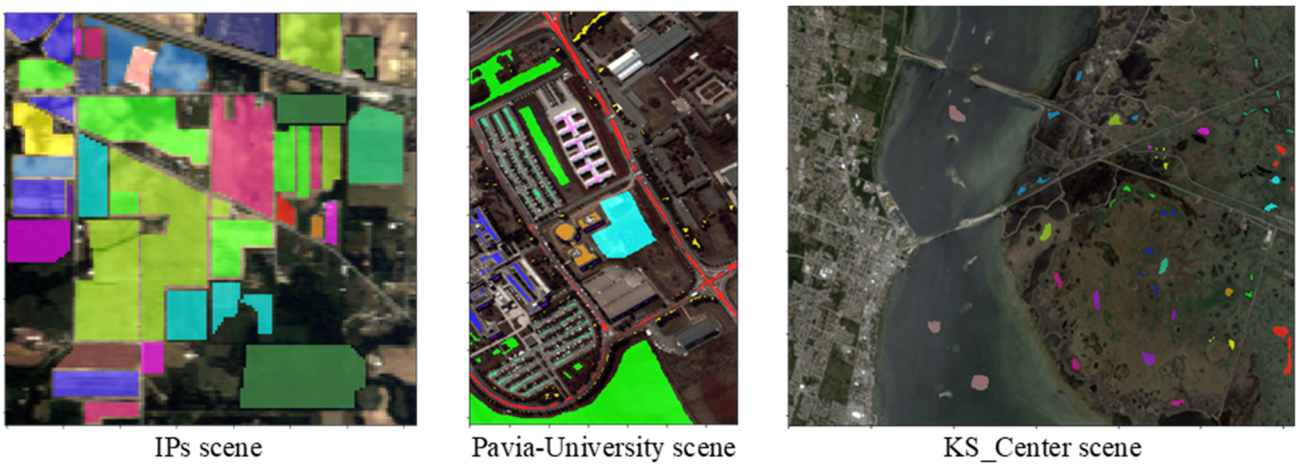


Figure 2. Dataset images with the ground truth.

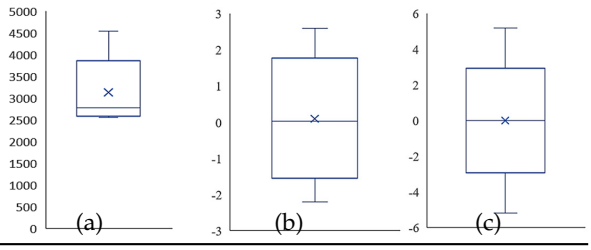
The second data set is the Pavia-University dataset which was collected by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor from northern Italy, including nine urban land-cover types and having 610 × 340 spatial dimensions, Figure 2. The spatial resolution is 1.3 m per pixel. The original data set contains 103 spectral bands with wavelength ranging from 0.43 to 0.86 µm.

The third dataset is the Kennedy Space Center (KS_Center) dataset which was acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over KS_Center, Florida. It contains 13 classes. The image dimension is 512 × 614 pixels, and the high spatial resolution is 18 m pixels. The original data set contains 176 spectral bands, Figure 2.

The class names of each dataset and the sample number in level-0, level-1 models, and testing sets are shown in Tables 2–4. The three datasets were split into three parts, representing 20% of the training set for the level-0 models, 10% of the training set for the level-1 model, and the rest of the data, which is 70%, is the testing set to test the final results of this study framework.

Table 2. One example for the data distribution of the Indian Pines (IPs) dataset (band-1). (a) is the distribution of the original data, (b) is for the data after using PCA, and (c) is the shape of the data after using QPCA.

	Normal	PCA	QPCA
Count	21,025	21,025	21,025
Mean	2957.36	6.5×10^{-16}	−0.0011
Std	354.919	1	0.95359
Min	2560	−2.2061	−5.1993
25%	2602	−0.887	−0.6739
50%	2780	0.0208	0.0031
75%	3179	0.97266	0.67668
Max	4536	2.58808	5.19934



To assess the accuracy of final results and to compare them with the results of other methods, the Kappa coefficient (KA), overall accuracy (OA), and average accuracy (AA) were used to measure the classification accuracy. Adam was used as an optimizer function with a 0.001 learning rate in the training time. The number of epochs and batch size were 100 and 256 in the level-0 and level-1 networks.

3.2. Experimental Results and Comparisons

This section clearly shows the output of QPCA and its effect on the data distribution. The output of each level in MLHM is presented for the three different datasets and their performance. Furthermore, MLHM is compared with other methods; with and without QPCA.

Table 3. The sample number of level-0 and level-1 datasets. It also indicates the results of the two level-0 models and the final results of the level-1 model for the IPs dataset.

#	Classes	Level-0 Samples	Level-1 Samples	Testing Samples	Level-0		Level-1
					QPCA-Model-1	QPCA-Model-2	QPCA-MLHM
1	Alfalfa	9	4	33	1	1	1
2	Corn-notill (CN)	285	114	1029	0.964	0.977	0.983
3	Corn-mintill (CM)	166	66	598	0.998	1	1
4	Corn	47	19	171	0.988	1	1
5	Grass-pasture (GP)	97	39	347	0.994	1	0.997
6	Grass-trees (GT)	146	58	526	0.998	0.990	0.994
7	Grass-pasture-mowed (GPM)	6	2	20	1	1	1
8	Hay-windrowed (HW)	96	38	344	1	1	1
9	Oats	4	2	14	1	1	1
10	Soybean-notill (SN)	194	78	700	0.994	1	1
11	Soybean-mintill (SM)	491	197	1767	0.998	0.998	0.997
12	Soybean-clean (SC)	118	48	427	0.991	0.979	0.998
13	Wheat	41	16	148	0.993	1	1
14	Woods	253	101	911	1	1	1
15	Buildings-Grass-Trees-Drives (BGTD)	77	31	278	1	0.996	1
16	Stone-Steel-Towers (SST)	19	7	67	0.970	0.970	0.970
	Kappa accuracy (%)				99.119	99.320	99.536
	Overall accuracy (%)				99.228	99.404	99.594
	Average accuracy (%)				99.310	99.443	99.618
	Training Time (s)				138.50	70.02	28.28
	Testing Time (s)				1.61	1.22	2.66

Table 4. The sample number of level-0 and level-1 datasets. It also indicates the results of the two level-0 models and the final results of the level-1 model for the Pavia-University dataset.

#	Classes	Level-0 Samples	Level-1 Samples	Testing Samples	Level-0		Level-1
					QPCA-Model-1	QPCA-Model-2	QPCA-MLHM
1	Asphalt	1326	530	4775	1	1	1
2	Meadows	3730	1492	13427	1	1	1
3	Gravel	420	168	1511	0.999	1	1
4	Trees	613	245	2206	0.995	0.995	0.999
5	Painted metal sheets (BMS)	269	108	968	1	1	1
6	Bare Soil (BS)	1006	402	3621	1	1	1
7	Bitumen	266	106	958	1	0.994	1
8	Self-Blocking Bricks (SBB)	736	295	2651	0.999	0.995	0.999
9	Shadows	189	76	682	1	1	1
	Kappa accuracy (%)				99.931	99.880	99.978
	Overall accuracy (%)				99.948	99.909	99.984
	Average accuracy (%)				99.920	99.83	99.977
	Training Time (s)				117.32	79.65	27.00
	Testing Time (s)				2.40	2.33	3.92

3.2.1. QPCA Output

Generally, collected data have many things that affect the training, such as insufficient data, missing values, highly skewed distribution, and outliers. In order to train the data, first, showing data by a specific graph is necessary to decide which kind of processing is needed to improve the input data before and through training time, as shown in Figure 3a the original data distribution of band-1 for Indian Pines dataset. When the dataset has values with variance dimensions, some larger than others, these large values can dominate the training in DL and affect the accuracy [61]. Therefore, to enhance the data before feeding it to the training network, preprocessing methodologies are necessary.

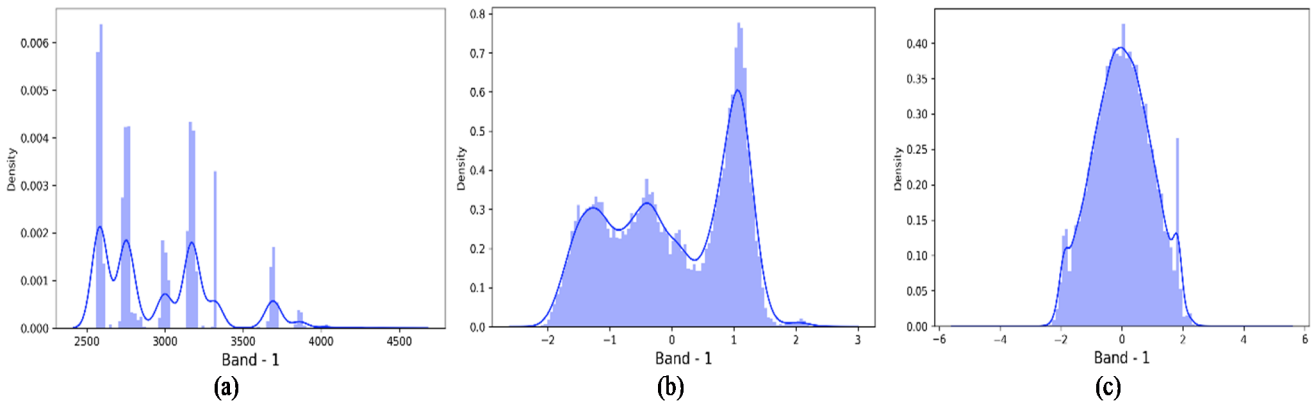


Figure 3. The three shapes of the data distribution, (**a**) is the original data, (**b**) is the change of data distribution after using principal components analysis (PCA), and (**c**) is the data distribution after using quantile transformation principal components analysis (QPCA).

The output of PCA depends on the results of the mean and variance—standardization; both are very sensitive to the outliers. This way of standardization is difficult with outlier values because this equation results would be skewed. Standardization works to shift the data, but the shape of the distribution does not change. Hence, the PCA does not work to enhance the data distribution and classification. Figure 3b shows that features with a large variance will dominate the training and affect the final results. On the other hand, in some cases, the outliers help to enhance the classification, especially in a huge and multi-class dataset, so they cannot be neglected. Quantile transformation (Q) is used to solve this problem.

The problem of outliers and skewed distribution is solved by Equation (4). As shown in Figure 3c, all the data are transferred under a normal curve and thus improves the classification process. With little frequency data and more correlation and repetition, data form the same normal distribution curve. Unlike the PCA, GCDF does not neglect any value. Hence, it enhances the data. Equation (4) normalizes the data to be normal-like. Figure 3 and Table 2 show the big difference between the original data distribution, PCA and QPCA. The data of QPCA is more stable and has normal distribution than PCA, which helps to obtain more stable training and higher accurate classification.

3.2.2. Meta-Learner Hybrid Model (MLHM) Results

The MLHM is a framework that follows the meta-learner technique or meta-learning. The network of level-1 works to train the trained weight of level-0 models. In the same word, the level-1 network learns from level-0 learning models. Level-1 network freezes the level-0 models to prevent them from modifying their generated weights, and it works to train them with its new extracted weights to reach the best accuracy. It can be observed that MLHM does not let the input data pass through a very deep network, and at the same time, it takes the benefits of deep training models to obtain more accurate features.

First of all, the two models of Level-0 were trained separately and later combined the output of these two models and fed them into the level-1 model to enhance the operation

of the previous level and train it with different data. This operation allowed the network to decide, choose, and obtain the best weight for each class. While model-1 works to deeply extract the complex classes' features, model-2 enhances feature extraction and the localization of less complex and small data. Both models of Level-0 work to extract spatial-spectral features. Therefore, the level-1 model is fed by the features of the level-0 models and it, in turn, enhances the extracted weight and provides the best results for the different classes.

Tables 3–5 show the results of each level of the three different datasets. In level-0 models, model-1 results are better than model-2. However, in some classes of IPs and Pavia-University datasets, model-2 had better results. The data distribution of KS_Center is very complex, and the sample numbers of its classes are almost similar. For these reasons, the accuracy of model-2 is very high compared to model-1 for all classes. Although the Level-1 model has fewer layers, it appreciably optimized the final results of IPs and Pavia-University datasets. In the KS_Center dataset, because the accuracy of model-1 of Level-0 is less than model-2 of Level-0, that affected the final results of QPCA-MLHM and obtained relatively lower results, barely 0.03. Figures 4–6 show the final results of each model. It clearly showed the ability of the framework to extract the features and detect the classes of each target correctly.

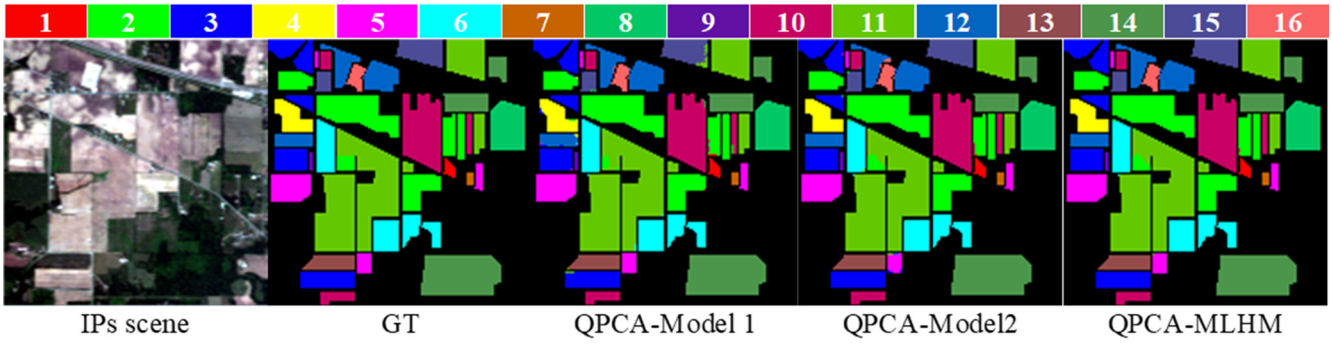


Figure 4. The results of both levels models for the IPs dataset.

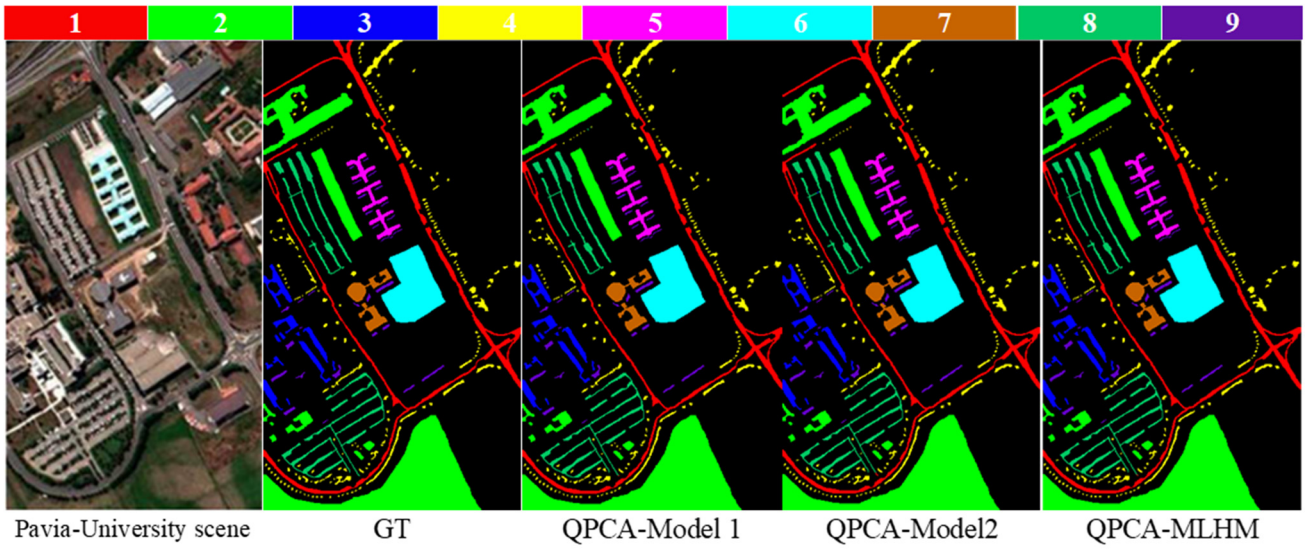


Figure 5. The results of both levels models for the Pavia-University dataset.

Table 5. The sample number of level-0 and level-1 datasets. It also indicates the two level-0 models' results and the final results of the level-1 model for the KS_center dataset.

#	Classes	Level-0 Samples	Level-1 Samples	Testing Samples	Level-0		Level-1
					QPCA-Model-1	QPCA-Model-2	QPCA-MLHM
1	Scrub	152	61	548	1	1	1
2	Willow swamp (WS)	49	19	175	0.817	0.977	0.971
3	Cabbage palm hammock (CPH)	51	21	184	0.152	0.984	0.989
4	Cabbage palm/oak hammock (CPOH)	50	20	182	0.967	0.995	0.995
5	Slash pine (SP)	32	13	116	0.793	0.991	0.991
6	Oak/broadleaf hammock (OBH)	46	18	165	1	1	1
7	Hardwood swamp (HS)	21	8	76	1	1	1
8	Graminoid marsh (GM)	86	34	311	0.936	1	0.997
9	Spartina marsh (SM)	104	42	374	1	1	1
10	Cattail marsh (CM)	81	32	291	0.990	1	1
11	Salt marsh (SM)	84	34	301	1	1	1
12	Mud flats (MF)	101	40	362	0.994	1	1
13	Wate	185	74	668	1	1	1
	Kappa accuracy (%)				92.774	99.733	99.703
	Overall accuracy (%)				93.525	99.760	99.734
	Average accuracy (%)				89.610	99.590	99.563
	Training Time (s)				86.13	15.28	6.04
	Testing Time (s)				0.35	0.31	0.51

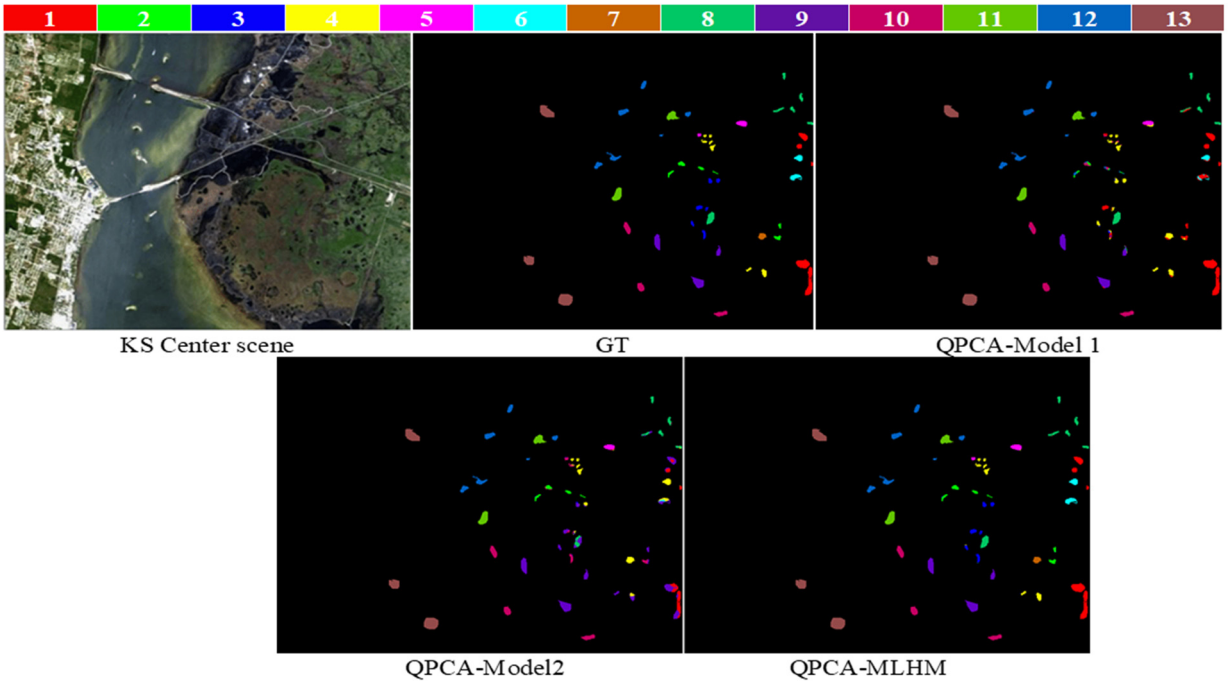


Figure 6. The results of both levels models for the KS_Center dataset.

In addition, using BN in the deeper network such as Model 1 of Level-0 in the experiment was more effective than using the dropout layer. That notwithstanding, using the dropout layer in the shallow model provides higher accuracy than using BN, which was observed in model 2 of Level-0.

Figure 7 shows the loss values during the training time, and it can be observed that the Level-1 model gives a more stable and smoother training than models of Level-0.

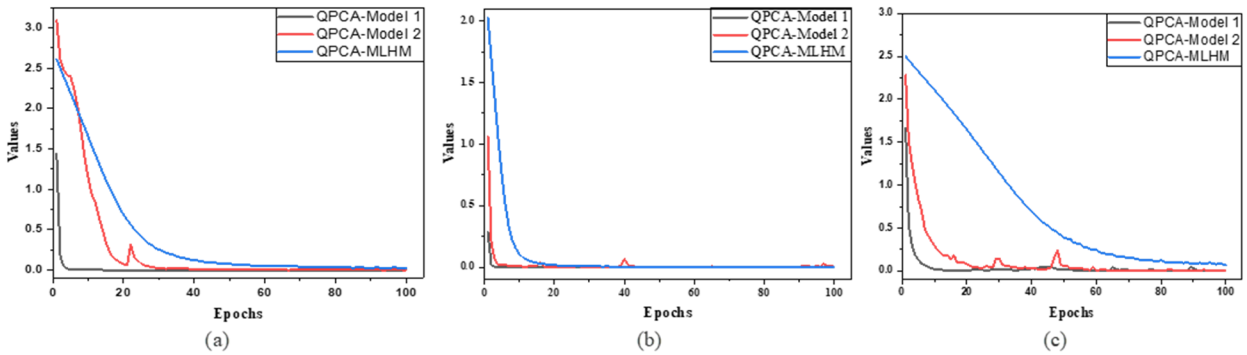


Figure 7. Loss values of the three datasets for the two models of Level-0 (QPCA-Model 1 and QPCA-Model 1) and Level-1 model (QPCA-MLHM). (**a**) is the loss values of the IPs dataset, (**b**) is the loss values of the Pavia-University dataset, and (**c**) is the loss values of the Kennedy Space Center dataset.

3.2.3. Comparisons with Other Methods

Also, to see the effectiveness of the QPCA optimizer and MLHM framework, four different state-of-the-art models with various structures were trained, one time with PCA and the other with QPCA preprocessing. The four models are SVM, CNN-1d [62], VGG-16 [63], and HybridCNN [19]. The experiments of MLHM and other methods have been repeated 10 times, and the mean and standard deviation of KA, OA, and AA coefficients are recorded to compare the results of the various methods.

VGG-16 is a deep network of 14 2D-CNN layers having the same kernel size (KS) (3×3), two FC layers, both using RAF, and four layers of max pooling (2, 2). The bounds number was reduced to 30 for the IP dataset and 15 for Pavia-University and KS_Center datasets. The training and testing set rates were 20% and 80%.

CNN1d is a layer of 1D-CNN to train spatial information with RAF, and 24 KS, two layers of FC, max pooling (2, 2), and BN. The bounds number of all datasets was reduced to 30 and split into 20% as training set and 80% as testing set, respectively.

HybridCNN works to train spectral-spatial features of hybrid window sizes (15, 13, 9) reduced by PCA in the preprocessing stage to 15 for all input data. Also, it divides the input data into three sets of representation: 20% training, 10% validation, and 70% for testing. Three layers of 3D-CNN trained each window size with different KSs and RAFs. Then, extracted features of the three networks were then concatenated and fed into a network of two 2D-CNN layers with different KSs and RAF to obtain the final classification. The epoch and batch sizes were 50 and 200. Adam optimizer with 0.001 learning rate (LR) was used in VGG-16, CNN1d, and HybridCNN.

Tables 6 and 7 shows that SVM and CNN1d with QPCA achieved less accuracy in the IPs and Pavia-University datasets. On the other hand, because the KS_Center dataset is the most complex, QPCA enhanced the data distribution to be more normal. So, QPCA significantly increased the accuracy of SVM, which is around 25%, and 19% for CNN1d in the KS_Center dataset, as indicated in Table 8. Furthermore, the QPCA optimizer played a great role with all the other models in all the datasets, as shown in Tables 6–8. For instance, with the IPs dataset, QPCA improved the output of the VGG-16 by 10% compared to using only PCA, Table 6. Similarly, in the KS_Center dataset, the QPCA increased the results of the HybridCNN model by more than 13%, Table 8. QPCA functioned to obtain the most influential bands of the input dataset. Thus, it improved the models' results.

Table 6. The comparison of the various models with PCA and QPCA preprocessing for the IPs dataset.

#	Calsses	PCA-SVM	QPCA-SVM	PCA-CNN1d	QPCA-CNN1d	PCA-VGG-16	QPCA-VGG-16	PCA-Hybrid CNN	QPCA-Hybrid CNN	PCA-MLHM	QPCA-MLHM
1	Alfalfa	0.70	0.16	0.78	0.44	0.00	0.00	0.97	0.97	1	1
2	CN	0.72	0.72	0.79	0.72	0.92	0.98	0.98	1	0.98	0.98
3	CM	0.75	0.67	0.71	0.70	0.94	0.99	0.99	1	1	1
4	Corn	0.64	0.45	0.62	0.60	1.00	0.93	0.98	1	1	1
5	GP	0.92	0.87	0.91	0.94	0.66	0.86	0.98	0.98	0.99	1
6	GT	0.95	0.96	0.93	0.95	0.57	0.99	1.00	1	1	0.99
7	GPM	0.96	0.82	1.00	1.00	0.00	0.00	1.00	1	1	1
8	HW	0.97	0.96	1.00	1.00	0.14	1.00	1.00	1	1	1
9	Oats	0.56	0.38	0.25	0.50	0.00	0.38	0.86	1	0.86	1
10	SN	0.72	0.65	0.73	0.70	0.93	0.95	0.98	0.98	1	1
11	SM	0.86	0.78	0.86	0.82	0.99	0.99	1.00	1	1	1
12	SC	0.74	0.64	0.69	0.69	0.95	0.98	0.97	0.99	0.98	1
13	Wheat	0.97	0.93	0.95	1.00	0.99	1.00	1.00	1	1	1
14	Woods	0.96	0.91	0.97	0.99	0.99	0.99	1.00	1	1	1
15	BGTD	0.61	0.57	0.69	0.60	0.96	0.84	0.99	1	1	1
16	SST	0.87	0.81	0.84	0.79	0.00	0.04	0.97	0.99	1	0.97
	KA (%)	80 (0.09)	73.91 (0.35)	80.39 (1.27)	78.33 (0.93)	83.77 (13.13)	94.68 (2.86)	98.96 (0.82)	99.25 (0.28)	99.39 (0.07)	99.41 (0.06)
	OA (%)	82.50 (0.08)	77.16 (0.31)	82.84 (1.11)	81.03 (0.80)	85.87 (11.38)	95.34 (2.50)	99.09 (0.72)	99.34 (0.25)	99.47 (0.07)	99.47 (0.06)
	AA (%)	80.28 (0.18)	70.63 (0.54)	79.48 (3.15)	78.24 (2.16)	66.58 (19.31)	81.89 (9.03)	98.17 (1.04)	98.71 (0.51)	99.01 (0.40)	99.36 (0.27)
	Tr.T. [1] (s)	0.41	0.67	21.87	36.04	53.44	50.95	65.60	60.40	41.67	25.47
	Ta.T. [2] (s)	2.40	2.45	0.25	0.27	2.20	2.24	1.93	1.83	4.53	3.64

[1] Trining Time (Tr.T); [2] Testing Time (Ta.T).

Table 7. The comparison of the various models with PCA and QPCA preprocessing for the Pavia-University dataset.

#	Calsses	PCA-SVM	QPCA-SVM	PCA-CNN1d	QPCA-CNN1d	PCA-VGG-16	QPCA-VGG-16	PCA-Hybrid CNN	QPCA-Hybrid CNN	PCA-MLHM	QPCA-MLHM
1	Asphalt	0.95	0.94	0.95	0.90	1	1	1	1	1	1
2	Meadows	0.98	0.96	0.98	0.96	1	1	1	1	1	1
3	Gravel	0.80	0.74	0.80	0.72	1	1	1	1	1	1
4	Trees	0.94	0.90	0.94	0.86	0.99	0.99	1	1	1	1
5	BMS	1.00	1.00	0.99	0.99	1	1	1	1	1	1
6	BS	0.88	0.88	0.91	0.83	1	1	1	1	1	1
7	Bitumen	0.86	0.86	0.91	0.79	1	1	0.99	1	1	1
8	SBB	0.91	0.82	0.89	0.82	1	1	0.98	1	1	1
9	Shadows	1	1	1	0.99	0.96	0.98	1	1	0.99	1
	KA (%)	92.09 (0.01)	89.66 (0.03)	93.69 (0.30)	91.88 (0.46)	99.68 (0.24)	99.81 (0.05)	99.86 (0.06)	99.93 (0.04)	99.93 (0.01)	99.94 (0.02)
	OA (%)	93.86 (0.62)	92.22 (0.02)	95.25 (0.22)	93.89 (0.35)	99.76 (0.18)	99.86 (0.04)	99.90 (0.05)	99.95 (0.03)	99.94 (0.01)	99.96 (0.02)
	AA (%)	92.37 (0.59)	90.07 (0.02)	93.70 (0.40)	92.01 (0.49)	99.48 (0.29)	99.68 (0.07)	99.82 (0.07)	99.91 (0.05)	99.88 (0.04)	99.90 (0.04)
	Tr.T. (s)	1.58	2.03	77.81	78.64	234.05	208.66	239.24	211.71	38.25	38.05
	Ta.T.(s)	6.34	7.01	0.66	0.71	7.45	7.39	7.88	7.51	6.08	6.03

Table 8. The comparison of the various models with PCA and QPCA preprocessing for the KS_center dataset.

#	Calsses	PCA-SVM	QPCA-SVM	PCA-CNN1d	QPCA-CNN1d	PCA-VGG-16	QPCA-VGG-16	PCA-Hybrid CNN	QPCA-Hybrid CNN	PCA-MLHM	QPCA-MLHM
1	Scrub	0.97	0.90	0.97	0.95	0.98	1	0.96	1	1	1
2	WS	0	0.71	0.78	0.78	0.59	0.62	0.92	0.98	0.96	0.99
3	CPH	0	0.68	0	0.71	0	0.92	0.80	0.97	0.98	0.99
4	CPOH	0	0.35	0.16	0.48	0.60	0.94	0.64	0.99	0.88	0.99
5	SP	0	0.46	0	0.28	0.95	0	0.84	0.95	0.94	0.98
6	OBH	0	0.32	0	0.30	0.10	1	0.43	0.99	0.99	1
7	HS	0	0.71	0	0.71	0	0	0.70	1	0.93	1
8	GM	0.18	0.55	0.36	0.58	0.02	0.96	0.79	0.98	0.96	1
9	SM	0.71	0.83	0.76	0.87	0.04	1	0.80	1	1	1
10	CM	0.02	0.55	0.04	0.49	0.84	1	0.94	0.99	0.95	1
11	SM	0.88	0.92	0.83	0.96	0.60	1	1	1	1	1
12	MF	0.52	0.69	0.65	0.76	0.97	0.95	0.97	1	1	1
13	Wate	1	0.98	0.98	0.95	0.97	1	1	1	1	1
	KA (%)	45.36 (0.10)	71.80 (0.39)	54.07 (1.69)	73.36 (0.79)	54.32 (13.94)	91.54 (5.01)	85.97 (4.59)	99.11 (0.15)	97.44 (0.61)	99.57 (0.13)
	OA (%)	52.68 (0.08)	74.71 (0.35)	59.54 (1.55)	76.08 (0.72)	60.15 (11.51)	92.41 (4.49)	87.38 (4.19)	99.20 (0.14)	97.70 (0.55)	99.62 (0.11)
	AA (%)	32.87 (0.06)	66.85 (0.54)	42.73 (0.96)	69.39 (1.60)	53.93 (12.17)	85.54 (8.59)	84.63 (3.10)	98.76 (0.27)	96.42 (0.83)	99.34 (0.20)
	Tr.T. (s)	0.07	0.04	20.14	21.15	72.54	72.72	32.42	36.96	8.80	8.38
	Ta.T.(s)	0.31	0.20	0.25	0.17	1.07	0.99	0.96	1.06	0.91	0.89

The KS_Center dataset has the most complex data distribution, so getting very high accuracy classification without enhancing the data distribution beforehand is complicated in the classification [39,64]. Not always increasing the number of samples gives higher accuracy. Thus, using QPCA introduced a very significant enhancement with the KS_Center dataset and gave the best accuracy in all models used, as can be seen in Table 8.

The structure of each compared model is different; SVM is simple and one of the old classification methods, SVM and CNN1d, have a less complex operation than others. The VGG-16 has 16 layers; 14 are 2D-CNN, and two are FC layers with RAF. HybridCNN is a hybrid model of three parallel lines of 3D-CNN layers, combined by 2D-CNN layers and hybrid window sizes with RAF to extract spatial-spectral features of the input data. However, the MLHM framework combines and trains the output of different computation and structure models. These models use hybrid CNN (2D and 3D) with different kernel sizes and BN and dropout layers to avoid overfitting. From Tables 6–8, it can be seen that this study's framework with QPCR normalization had the best results in all datasets.

Furthermore, the MLHM framework with PCA also obtained the highest accuracy compared to the other models with PCA only. Because HybridCNN was designed to obtain the spatial-spectral features simultaneously and use 3D-CNN, it had better results than VGG-16, even though it had more layers than others. In general, the QPCA-MLHM was the best flexible framework that provided an effective way to classify HS images and had the highest accuracy. Moreover, it introduced the best results for the most accurate different classes in all datasets, as shown in Figures 8–10.

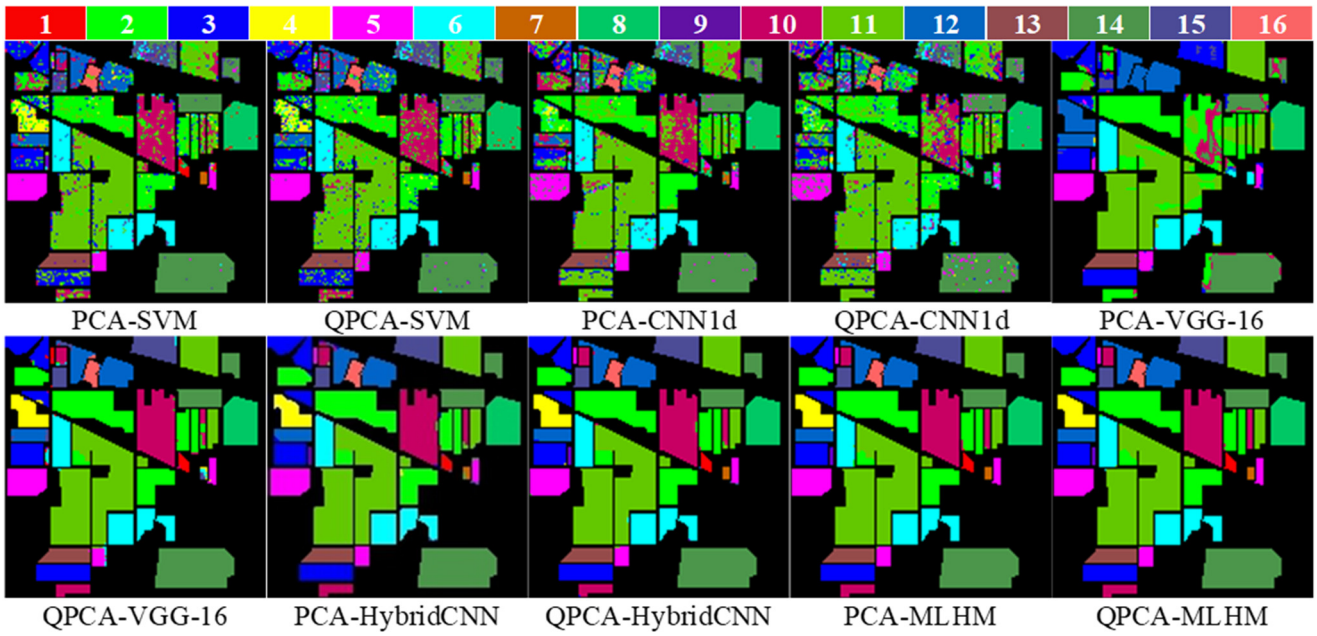


Figure 8. The output of the various models with PCA and QPCA preprocessing for the IPs dataset.

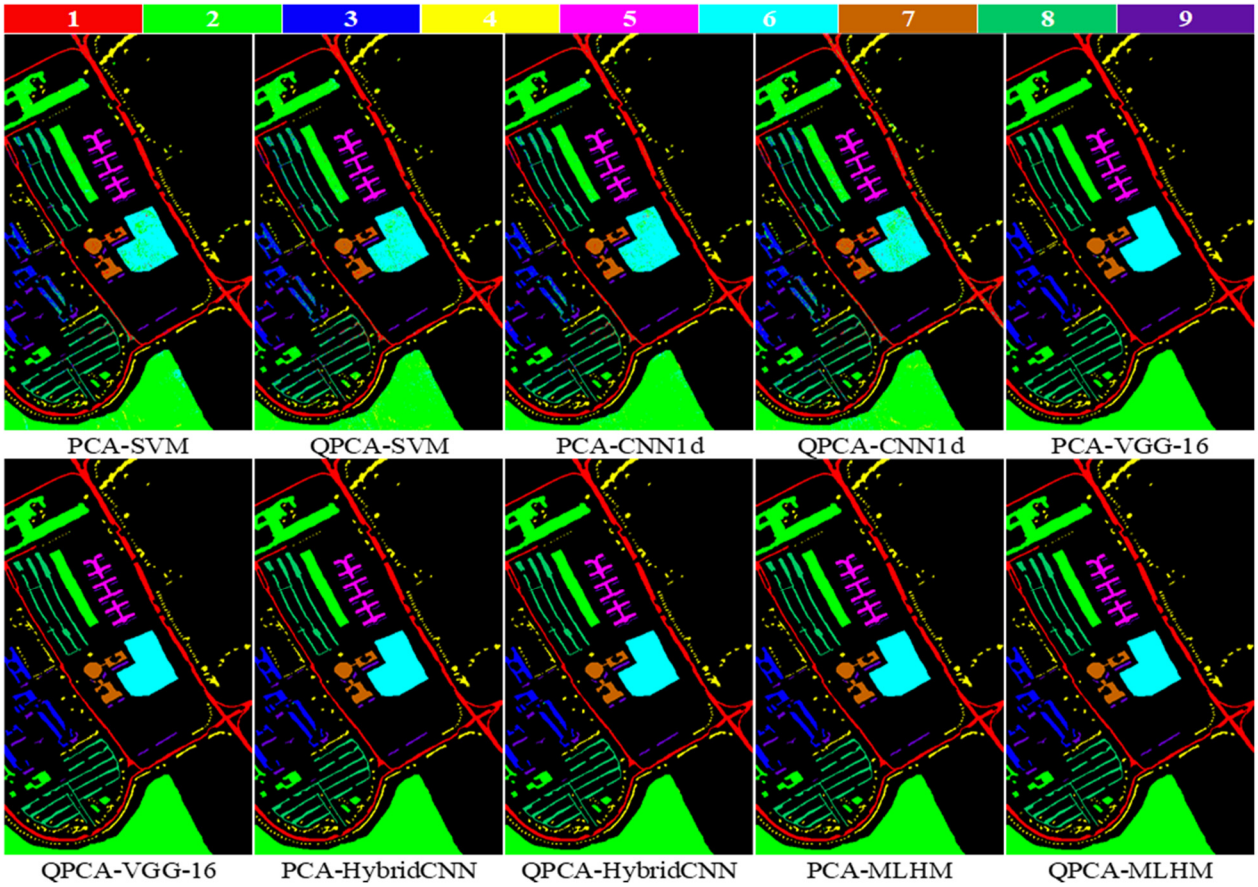


Figure 9. The output of the various models with PCA and QPCA preprocessing for the Pavia University dataset.

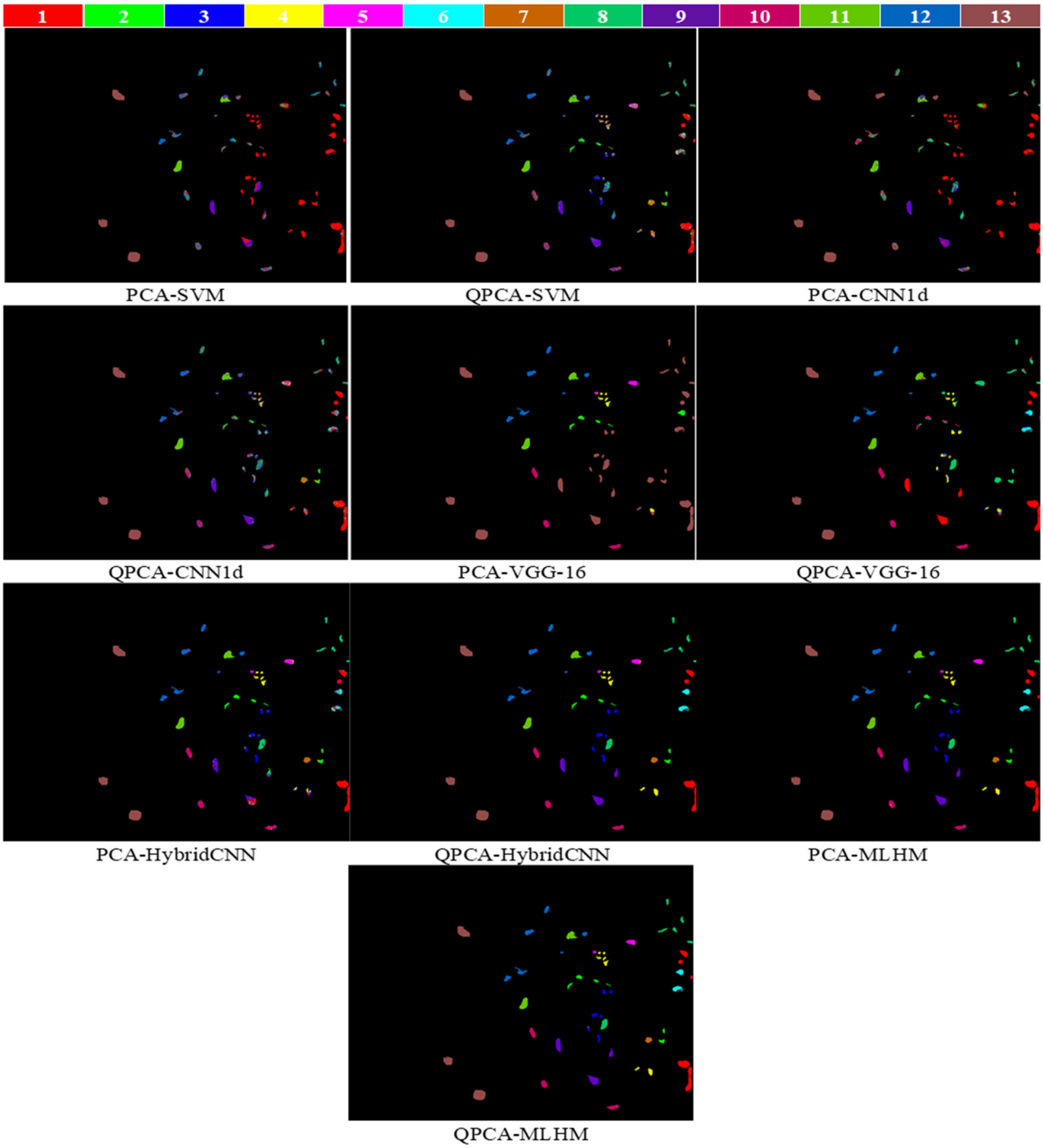


Figure 10. The output of the various models with PCA and QPCA preprocessing for the KS_Center dataset.

In the very deep networks (VGG-16, HybridCNN, MLHM), the QPCA preprocessing speeded up the running time more than PCA. QPCA works to normalize the distribution; thus, it improves spatial information. Moreover, the VGG-16 model focuses on extracting the spatial features, while HybridCNN and MLHM focus on extracting spatial-spectral features. These models achieved higher accuracy and less performance time with QPCA. The SVD and CNN1d process the spectral features, resulting in less accuracy with QPCA. MLHM as a deep model extracting spatial-spectral features is the fastest model, Tables 6–8.

Figure 11 shows the accuracy during the training time for the eight experiments of the four models with PCA and QPCA. It can be observed that QPCA makes the training

more stable and accelerates the process of obtaining higher accuracy more than using PCA, especially with the KS_Center dataset. Moreover, MLHM introduces the best processing, and it is the smoothest model during the training time.

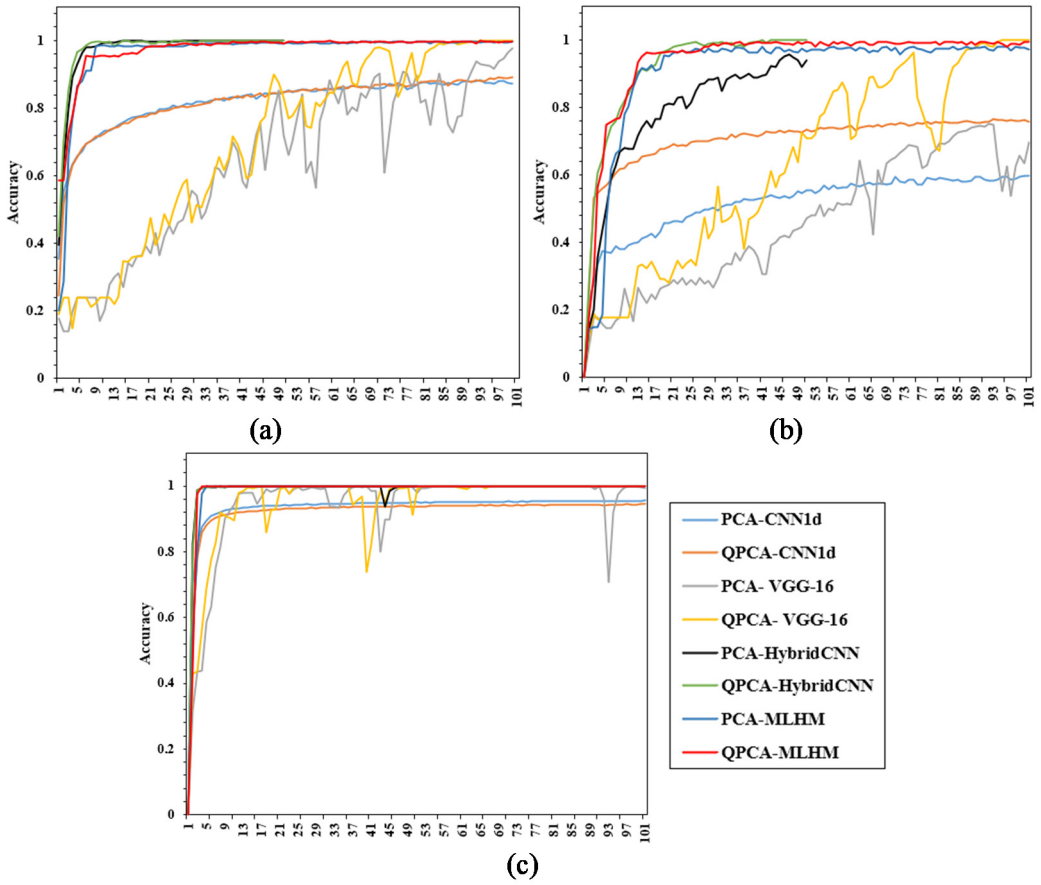


Figure 11. The training accuracy in each model for the three used datasets: (**a**) IPs dataset, (**b**) the Pavia University dataset, and (**c**) the KS_Center dataset.

4. Conclusions

This study proposed quantile transformation principal component analysis (QPCA) as preprocessing to enhance the variations between HS image pixels and the discriminability of different pixels, and to redistribute data so that it appeared normal. However, to improve the feature extraction and deal with the challenges of HS images, this study combined the shallow and deep network of hybrid structures of 2D and 3D-CNN in a novel framework to provide high classification accuracy for the HS image data on two levels. This framework structure with two learning levels was proposed, MLHM. It follows the meta-learner technique to learn from learned models. Two base models (shallow and deep) used hybrid CNN with different kernel sizes; three HS image datasets were used to evaluate the proposed framework. From experiments of MLHM and the other compared methods, the following conclusions were drawn: (1) combining the output of shallow and deep models using meta-learner techniques provides better results than training a deep or hybrid model. (2) QPCA enhances the data distribution and reduces the dimension of each input dataset before training. (3) QPCA-MLHM showed the best classification for each class in each dataset. (4) BN works better in deep networks, and dropout is better in shallow networks. (5) QPCA enhanced the results by 13% for several models and more than 25% for the SVM model. (6) MLHM, compared with the deep models, provides the best performance and speed. Future work needs to be focused on improving the data distribution and skewness and promoting feature extraction performance.

Author Contributions: Conceptualization, M.A.A.A.-q. and Z.C.; Formal analysis, D.A.-A. and A.D.; Funding acquisition, M.A.A.A.-q.; Methodology, D.A.-A.; Project administration, M.A.A.A.-q.; Software, D.A.-A.; Supervision, Z.C., and M.A.A.A.-q.; Validation, Z.C.; Visualization, D.A.-A.; Writing—original draft, D.A.-A.; Writing—review & editing, M.A.A.A.-q., A.D., Y.S. and S.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by LIESMARS Special Research Funding.

Data Availability Statement: All the datasets are available at this link: http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes, accessed on 10 October 2021.

Conflicts of Interest: The authors reported no potential conflict of interest.

References

1. Mei, X.; Pan, E.; Ma, Y.; Dai, X.; Huang, J.; Fan, F.; Du, Q.; Zheng, H.; Ma, J. Spectral-Spatial Attention Networks for Hyperspectral Image Classification. *Remote Sens.* **2019**, *11*, 963. [CrossRef]
2. Tao, C.; Wang, Y.; Cui, W.; Zou, B.; Zou, Z.; Tu, Y. A transferable spectroscopic diagnosis model for predicting arsenic contamination in soil. *Sci. Total Environ.* **2019**, *669*, 964–972. [CrossRef]
3. Ma, L.; Crawford, M.M.; Zhu, L.; Liu, Y. Centroid and Covariance Alignment-Based Domain Adaptation for Unsupervised Classification of Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 2305–2323. [CrossRef]
4. Luft, L.; Neumann, C.; Freude, M.; Blaum, N.; Jeltsch, F. Hyperspectral modeling of ecological indicators—A new approach for monitoring former military training areas. *Ecol. Indic.* **2014**, *46*, 264–285. [CrossRef]
5. Camps-Valls, G.; Tuia, D.; Bruzzone, L.; Benediktsson, J.A. Advances in Hyperspectral Image Classification: Earth Monitoring with Statistical Learning Methods. *IEEE Signal Process. Mag.* **2013**, *31*, 45–54. [CrossRef]
6. Tusa, E.; Laybros, A.; Monnet, J.-M.; Dalla Mura, M.; Barré, J.-B.; Vincent, G.; Dalponte, M.; Féret, J.-B.; Chanussot, J. Fusion of hyperspectral imaging and LiDAR for forest monitoring. In *Hyperspectral Imaging*; Amigo, J.M., Ed.; Elsevier: Amsterdam, The Netherlands, 2020; Volume 32, pp. 281–303. ISBN 0922-3487. [CrossRef]
7. Guo, A.T.; Huang, W.J.; Dong, Y.Y.; Ye, H.C.; Ma, H.Q.; Liu, B.; Wu, W.B.; Ren, Y.; Ruan, C.; Geng, Y. Wheat Yellow Rust Detection Using UAV-Based Hyperspectral Technology. *Remote Sens.* **2021**, *13*, 123. [CrossRef]
8. Zheng, Q.; Huang, W.; Ye, H.; Dong, Y.; Shi, Y.; Chen, S. Using continous wavelet analysis for monitoring wheat yellow rust in different infestation stages based on unmanned aerial vehicle hyperspectral images. *Appl. Opt.* **2020**, *59*, 8003. [CrossRef]
9. Cui, J.; Yan, B.; Dong, X.; Zhang, S.; Zhang, J.; Tian, F.; Wang, R. Temperature and emissivity separation and mineral mapping based on airborne TASI hyperspectral thermal infrared data. *Int. J. Appl. Earth Obs. Geoinf.* **2015**, *40*, 19–28. [CrossRef]
10. Li, Y.; Qian, M.; Liu, P.; Cai, Q.; Li, X.; Guo, J.; Yan, H.; Yu, F.; Yuan, K.; Yu, J.; et al. The recognition of rice images by UAV based on capsule network. *Clust. Comput.* **2018**, *22*, 9515–9524. [CrossRef]
11. Zhang, Y.; Ma, Y.; Dai, X.; Li, H.; Mei, X.; Ma, J. Locality-constrained sparse representation for hyperspectral image classification. *Inf. Sci.* **2020**, *546*, 858–870. [CrossRef]
12. McInnes, L.; Healy, J.; Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv* **2018**, arXiv:1802.03426.
13. Anowar, F.; Sadaoui, S.; Selim, B. Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). *Comput. Sci. Rev.* **2021**, *40*, 100378. [CrossRef]
14. Imani, M.; Ghassemian, H. An overview on spectral and spatial information fusion for hyperspectral image classification: Current trends and challenges. *Inf. Fusion* **2020**, *59*, 59–83. [CrossRef]
15. Ghojogh, B.; Samad, M.N.; Mashhadi, S.A.; Kapoor, T.; Ali, W.; Karray, F.; Crowley, M. Feature Selection and Feature Extraction in Pattern Analysis: A Literature Review. *arXiv* **2019**, arXiv:1905.02845.
16. Zhang, X.; Wei, Y.; Yao, H.; Ye, Z.; Zhou, Y.; Zhao, Y. Locally Homogeneous Covariance Matrix Representation for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 9396–9407. [CrossRef]
17. Yao, H.; Yang, M.; Chen, T.; Wei, Y.; Zhang, Y. Depth-based human activity recognition via multi-level fused features and fast broad learning system. *Int. J. Distrib. Sens. Netw.* **2020**, *16*, 1550147720907830. [CrossRef]
18. Yuan, Y.; Jin, M. Multi-type spectral spatial feature for hyperspectral image classification. *Neurocomputing* **2021**, *in press*. [CrossRef]
19. Mohan, A.; Venkatesan, M. HybridCNN based hyperspectral image classification using multiscale spatiospectral features. *Infrared Phys. Technol.* **2020**, *108*, 103326. [CrossRef]
20. Roy, S.K.; Dubey, S.R.; Chatterjee, S.; Chaudhuri, B.B. FuSENet: Fused squeeze-and-excitation network for spectral-spatial hyperspectral image classification. *IET Image Process.* **2020**, *14*, 1653–1661. [CrossRef]
21. 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]
22. Li, S.; Song, W.; Fang, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Deep Learning for Hyperspectral Image Classification: An Overview. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6690–6709. [CrossRef]
23. Camps-Valls, G.; Bruzzone, L. Kernel-based methods for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 1351–1362. [CrossRef]

24. Chen, Y.; Nasrabadi, N.M.; Tran, T.D. Hyperspectral Image Classification Using Dictionary-Based Sparse Representation. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3973–3985. [CrossRef]
25. Li, S.; Wu, H.; Wan, D.; Zhu, J. An effective feature selection method for hyperspectral image classification based on genetic algorithm and support vector machine. *Knowl.-Based Syst.* **2011**, *24*, 40–48. [CrossRef]
26. Yang, L.; Yang, S.; Jin, P.; Zhang, R. Semi-Supervised Hyperspectral Image Classification Using Spatio-Spectral Laplacian Support Vector Machine. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 651–655. [CrossRef]
27. Chen, Y.-N.; Thaipisutikul, T.; Han, C.-C.; Liu, T.-J.; Fan, K.-C. Feature Line Embedding Based on Support Vector Machine for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 130. [CrossRef]
28. Guo, Y.; Han, S.; Li, Y.; Zhang, C.; Bai, Y. K-Nearest Neighbor combined with guided filter for hyperspectral image classification. *Procedia Comput. Sci.* **2018**, *129*, 159–165. [CrossRef]
29. Tu, B.; Huang, S.; Fang, L.; Zhang, G.; Wang, J.; Zheng, B. Hyperspectral Image Classification via Weighted Joint Nearest Neighbor and Sparse Representation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 4063–4075. [CrossRef]
30. Huang, K.; Li, S.; Kang, X.; Fang, L. Spectral–Spatial Hyperspectral Image Classification Based on KNN. *Sens. Imaging* **2016**, *17*, 1–13. [CrossRef]
31. Cao, X.; Yao, J.; Xu, Z.; Meng, D. Hyperspectral Image Classification with Convolutional Neural Network and Active Learning. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4604–4616. [CrossRef]
32. Hang, R.; Li, Z.; Liu, Q.; Ghamisi, P.; Bhattacharyya, S.S. Hyperspectral Image Classification with Attention-Aided CNNs. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 2281–2293. [CrossRef]
33. Yu, C.; Han, R.; Song, M.; Liu, C.; Chang, C.-I. A Simplified 2D-3D CNN Architecture for Hyperspectral Image Classification Based on Spatial–Spectral Fusion. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 2485–2501. [CrossRef]
34. Bandyopadhyay, M. Multi-stack hybrid CNN with non-monotonic activation functions for hyperspectral satellite image classification. *Neural Comput. Appl.* **2021**, *33*, 14809–14822. [CrossRef]
35. Huang, L.; Chen, Y. Dual-Path Siamese CNN for Hyperspectral Image Classification with Limited Training Samples. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 518–522. [CrossRef]
36. Zhao, Q.; Jia, S.; Li, Y. Hyperspectral remote sensing image classification based on tighter random projection with minimal intra-class variance algorithm. *Pattern Recognit.* **2021**, *111*, 107635. [CrossRef]
37. Ramamurthy, M.; Robinson, Y.H.; Vimal, S.; Suresh, A. Auto encoder based dimensionality reduction and classification using convolutional neural networks for hyperspectral images. *Microprocess. Microsyst.* **2020**, *79*, 103280. [CrossRef]
38. Paul, A.; Bhoumik, S.; Chaki, N. SSNET: An improved deep hybrid network for hyperspectral image classification. *Neural Comput. Appl.* **2020**, *33*, 1575–1585. [CrossRef]
39. Wei, Y.; Zhou, Y. Spatial-Aware Network for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 3232. [CrossRef]
40. Zhao, J.; Huang, T.; Zhou, Z. Hyperspectral image super-resolution using recursive densely convolutional neural network with spatial constraint strategy. *Neural Comput. Appl.* **2020**, *32*, 14471–14481. [CrossRef]
41. Zhao, G.; Liu, G.; Fang, L.; Tu, B.; Ghamisi, P. Multiple convolutional layers fusion framework for hyperspectral image classification. *Neurocomputing* **2019**, *339*, 149–160. [CrossRef]
42. Al-Alimi, D.; Shao, Y.; Feng, R.; Al-Qaness, M.A.A.; Elaziz, M.A.; Kim, S. Multi-Scale Geospatial Object Detection Based on Shallow-Deep Feature Extraction. *Remote Sens.* **2019**, *11*, 2525. [CrossRef]
43. Cao, F.; Guo, W. Cascaded dual-scale crossover network for hyperspectral image classification. *Knowl.-Based Syst.* **2019**, *189*, 105122. [CrossRef]
44. Yang, J.; Xiong, W.; Li, S.; Xu, C. Learning structured and non-redundant representations with deep neural networks. *Pattern Recognit.* **2018**, *86*, 224–235. [CrossRef]
45. Guo, Y.; Cao, H.; Bai, J.; Bai, Y. High Efficient Deep Feature Extraction and Classification of Spectral-Spatial Hyperspectral Image Using Cross Domain Convolutional Neural Networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 1–12. [CrossRef]
46. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
47. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
48. Gao, H.; Yang, Y.; Li, C.; Zhang, X.; Zhao, J.; Yao, D. Convolutional neural network for spectral-spatial classification of hyperspectral images. *Neural Comput. Appl.* **2019**, *31*, 8997–9012. [CrossRef]
49. Ribeiro, M.H.D.M.; Coelho, L.D.S. Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series. *Appl. Soft Comput.* **2019**, *86*, 105837. [CrossRef]
50. Shamaei, E.; Kaedi, M. Suspended sediment concentration estimation by stacking the genetic programming and neuro-fuzzy predictions. *Appl. Soft Comput.* **2016**, *45*, 187–196. [CrossRef]
51. Pernía-Espinoza, A.; Fernandez-Ceniceros, J.; Antonanzas, J.; Urraca, R.; Martinez-De-Pison, F. Stacking ensemble with parsimonious base models to improve generalization capability in the characterization of steel bolted components. *Appl. Soft Comput.* **2018**, *70*, 737–750. [CrossRef]
52. Wozniak, M.; Graña, M.; Corchado, E. A survey of multiple classifier systems as hybrid systems. *Inf. Fusion* **2014**, *16*, 3–17. [CrossRef]

53. Kang, S.; Cho, S.; Kang, P. Multi-class classification via heterogeneous ensemble of one-class classifiers. *Eng. Appl. Artif. Intell.* **2015**, *43*, 35–43. [CrossRef]
54. Taormina, V.; Cascio, D.; Abbene, L.; Raso, G. Performance of Fine-Tuning Convolutional Neural Networks for HEp-2 Image Classification. *Appl. Sci.* **2020**, *10*, 6940. [CrossRef]
55. Zhong, C.; Zhang, J.; Wu, S.; Zhang, Y. Cross-Scene Deep Transfer Learning with Spectral Feature Adaptation for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 2861–2873. [CrossRef]
56. Liu, X.; Yu, L.; Peng, P.; Lu, F. A Stacked Generalization Framework for City Traffic Related Geospatial Data Analysis. In *Web Technologies and Applications*; Morishima, A., Zhang, R., Zhang, W., Chang, L., Fu, T.Z.J., Liu, K., Yang, X., Zhu, J., Zhang, Z., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 265–276.
57. Garcia-Ceja, E.; Galván-Tejada, C.E.; Brena, R. Multi-view stacking for activity recognition with sound and accelerometer data. *Inf. Fusion* **2018**, *40*, 45–56. [CrossRef]
58. Xu, Y.; Du, B.; Zhang, F.; Zhang, L. Hyperspectral image classification via a random patches network. *ISPRS J. Photogramm. Remote Sens.* **2018**, *142*, 344–357. [CrossRef]
59. Cheng, C.; Li, H.; Peng, J.; Cui, W.; Zhang, L. Hyperspectral Image Classification Via Spectral-Spatial Random Patches Network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 4753–4764. [CrossRef]
60. Garbin, C.; Zhu, X.; Marques, O. Dropout vs. batch normalization: An empirical study of their impact to deep learning. *Multimed. Tools Appl.* **2020**, *79*, 12777–12815. [CrossRef]
61. Fernández, A.; García, S.; Galar, M.; Prati, R.C.; Krawczyk, B.; Herrera, F. *Learning from Imbalanced Data Sets*, 1st ed.; Springer International Publishing: Cham, Switzerland, 2018; ISBN 978-3-319-98073-7. [CrossRef]
62. Paoletti, M.E.; Haut, J.M.; Plaza, J.; Plaza, A. Deep learning classifiers for hyperspectral imaging: A review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *158*, 279–317. [CrossRef]
63. Jiao, L.; Liang, M.; Chen, H.; Yang, S.; Liu, H.; Cao, X. Deep Fully Convolutional Network-Based Spatial Distribution Prediction for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5585–5599. [CrossRef]
64. Chu, Y.; Lin, H.; Yang, L.; Zhang, D.; Diao, Y.; Fan, X.; Shen, C. Hyperspectral image classification based on discriminative locality preserving broad learning system. *Knowl.-Based Syst.* **2020**, *206*, 106319. [CrossRef]