The Effect of the Principal Component Analysis in Convolutional Neural Network for Hyperspectral Image Classification

Taehong Kwak (1), Ahram Song (1), Yongil Kim (1)

Seoul National Univ., 1, Gwanak-ro, Gwanak-gu, Seoul, 08826, Korea
Email: mwsa302@snu.ac.kr; aram200@snu.ac.kr; yik@snu.ac.kr

Keywords: Principal component analysis (PCA), Deep learning, Convolutional neural network (CNN), Hyperspectral image classification (HSIC)

Abstract: Hyperspectral imagery is widely used in image classification due to their continuity and the abundance of spectral information. Recently, manifold deep learning algorithms were proposed for hyperspectral image classification (HSIC) demonstrating a competitive edge in performance over existing methods due to their ability to automatically extract high-level features. However, one significant drawback of hyperspectral images is their high dimensionality, which increases the learning time and processing complexity. To address this problem, several studies have exploited principal component analysis (PCA) as a pre-processing step in the deep learning framework to compress the entire image through a simple statistical processing of the spectral dimension while preserving the spatial information. However, since PCA can result in a loss in the spectral information, other studies have applied the original hyperspectral imagery as input to deep learning networks. At the same time, the impact of dimensionality reduction using PCA in deep learning networks on achieving efficient HSIC is understudied. Hence, the purpose of this study is to analyze the effect of PCA in deep learning for HSIC. In this paper, we verified the efficiency of deep learning networks through various conditions of PCA. We employed a convolutional neural network (CNN), which can extract spatial-spectral features of hyperspectral imagery. To analyze the sensitivity of PCA depending on CNN architectures, a two-dimensional CNN (2D-CNN) and a three-dimensional CNN (3D-CNN) were applied. We quantitatively analyzed the experimental results, which revealed that PCA can effectively reduce an image to its optimal spectral dimension according to CNN models for efficient CNN-based HSIC.

1. Introduction

Advances in sensors and photonics have brought the applicability of hyperspectral imagery with hundreds of narrow spectral bands. Broadband can only discriminate general differences between materials, whereas hyperspectral sensors allow acquiring sufficient spectral information for detailed identification with a high spectral resolution (Lillesand, 2015). Due to the continuity and abundance of this spectral information, hyperspectral image classification (HSIC) has been widely developed. Hyperspectral imagery has been shown to help achieve an accurate and robust classification performance, and various image classification methods have been developed. Conventional classification methods such as k-nearest neighbors (Samaniego, 2008), minimum distance, and logistic regression (Cheng, 2006) also use the abundant spectral information from hyperspectral imagery. Among other algorithms, support vector machine (SVM) has demonstrated a stable performance in HSIC. Fauvel (2008) analyzed the spatial-spectral information of hyperspectral imagery using SVM and morphological profiles, and were able to achieve excellent classification performance. The spatial-spectral properties were also analyzed through the fusion of pixel-wise SVM and a segmentation map obtained using partial clustering (Tarabalka, 2009).

Lately, deep learning has achieved promising performance in various areas including HSIC. For example, Chen (2014) exploited an autoencoder, one of the deep learning algorithms. Convolutional neural network(CNN) is another representative deep learning algorithm widely used for image classification, and also has been applied to HSIC in many studies. High level spatial-spectral features can be obtained using the CNN, allowing it to achieve high and stable classification accuracy. The spectral response function of each pixel was exploited to learn CNN model using the abundant spectral information of hyperspectral imagery (Slavkovikj, 2015; Hu, 2015). A joint spatial-spectral classification framework was constructed using a patch, splitting one pixel and neighbor pixels in a CNN (Makantasis 2015; Chen, 2016; Zhang, 2017). In addition, 3D-CNNs for HSIC have been studied, considering that they can comprehend 3D data cube with achieving a good performance (Li, 2017, He, 2017 and Li, 2017).

At the same time, the high spectral dimensionality of hyperspectral images causes some problems. High-dimensional data not only complicates analysis, but also make the learning process difficult due to a limited number of labelled samples. To solve this problem, many studies have conducted spectral dimensionality reduction methods. Principal component analysis(PCA) is a representative method that analyzes data based on several principal components through statistical analysis, and is often used for dimensionality reduction. The applicability of PCA to hyperspectral imagery has been studied in several papers. For example, Rodarmel (2002) established the applicability of PCA to HSIC, while most existing studies analyzed the impact of PCA in multispectral imagery. They concluded that up to the first 10 PCA bands are significant bands for achieving a similar classification performance to that when using original hyperspectral data. Statistical inference of estimated eigenvectors and eigenvalues was studied for the

dimensionality reduction of hyperspectral imagery through PCA (Bajorski, 2011). Agarwal (2007) proposed hierarchical PCA for an efficient dimensionality reduction of hyperspectral imagery.

Many CNN-based HSIC studies have also applied PCA. Several studies suggested that the size and dimension of hyperspectral imagery requires compression to allow a deep learning network to learn the enormous number of variables effectively (Makantasis, 2015, Zhao, 2016, Chen, 2016, Liang, 2016, Zhang, 2017 and Mei, 2019). Nevertheless, the effects of PCA for deep learning based HSIC still remain understudied. The existing CNN based HSIC studies using PCA have not focused on the impact of PCA and PC bands. Even, another studies have proposed CNN models that show good classification performance depending on suitable design of network, while not using PCA.

The purpose of this study is to analyze the effect of PCA in CNN based HSIC. In this paper, we verified the efficiency of the network through various conditions of PCA. In more detail, the main contributions of this study are as follows: 1) We focus on analyzing the impact of PCA rather than proposing an outperforming classification model. As discussed above, the impact of PCA in CNN is understudied, and this paper may contribute to the applicability of PCA. 2) We use two hyperspectral datasets with different spectral dimensions to explain that PCA can control the problem of dimensionality. 3) Two different models, namely, 2D-CNN and 3D-CNN, are exploited to validate the sensitivity of PCA according to CNN architectures. The remainder of this paper is organized as follows: the methodology of this study is explained in Section 2 including the description of PCA and CNN model architectures. The experimental results and their discussion are introduced in Section 3. Finally, Section 4 provides the conclusion of this study.

2. Methodology

2.1. PCA

PCA is a method that can be used to convert an original imagery into a small and independent set of variables, and effectively reducing the dimensionality of hyperspectral imagery (Jensen, 1987). Applicability of PCA to hyperspectral data results from mathematical properties based on eigenvalue decomposition of data covariance matrix (Σ). The transformation from a raw image to PC bands is based on eigenvalue decomposition as follows:

$$\Sigma = A\Lambda A^T \tag{1}$$

where $A = (a_1, a_2, \cdots, a_N)$ is the eigenvectors matrix, and Λ is the diagonal matrix composed of the eigenvalues.

The first K eigenvectors of A can be used to calculate a transformed pixel vector z_i from an original image pixel vector x_i via the following equation:

$$z_i = [z_1\ z_2\ \vdots\ z_K\ ] = [a_{11}\ \cdots\ a_{1N}\ \vdots \ddots \vdots\ a_{K1}\ \cdots\ a_{KN}\ ][x_1\ \vdots\ x_N\ ] \tag{2}$$

The most notable point in this equation is that the transformed PC bands are independent of each other by the property of eigenvectors. Since hyperspectral imagery has continuous spectral response functions, neighboring bands may have the redundancy problem. PCA can remove this correlation through the dependency of the PC bands. Variance explained by the first few PC bands is calculated as:

$$\frac{\sum_{i=1}^{K} \lambda_i}{\lambda_1 + \lambda_2 + \cdots + \lambda_N} \tag{3}$$

Since the first PC band can explain the highest variance of information, and the explained variance decreases, the appropriate number of PC bands should be determined for an efficient dimensionality reduction of hyperspectral imagery. In this study, we transformed hyperspectral data by increasing the number of PC bands to analyze the impact of PCA for CNN based HSIC.

2.2. 2D-CNN

The CNN is a deep learning algorithm widely-used for processing data with a known grid-like topology (Goodfellow, 2016). In the context of image classification, the CNN can consider spatial and spectral properties by exploiting a convolution filter. CNN image classification models are often constructed based on patches, including target pixels with their neighbor pixels.

Table 1 shows the 2D-CNN models used in this study. To compare the variability of classification performance according to the number of PC bands, the same CNN model is applied to all datasets pre-processed using PCA. In addition, the classification results of the PCA-CNN are compared with the results of the original hyperspectral imagery. A deeper and more complex model is exploited in the Original-CNN, since the original hyperspectral imagery has very high spectral dimensionality. The model consists of three convolutional layers and two fully connected layers. The size of the convolution filter is fixed to 3×3, and the number of filters is set to an appropriate value through repeated experiments. Each convolutional layer is followed by batch normalization (BN) for regularization. Dropout is used in the fully-connected layers to control overfitting, and the last layer has the same number of nodes as the number of its class.

Table 1 2D-CNN model architecture

PCA-CNN		Original-CNN	
Conv2D	(3,3) filter#:20 Zero Padding BN	**Conv2D**	(3,3) filter#:75 Zero Padding BN
Conv2D	(3,3) filter#:60 BN	**Conv2D**	(3,3) filter#:150 BN
Conv2D	(3,3) filter#:180 BN	**Conv2D**	(3,3) filter#:225 BN
Fully Connected	Nodes #: 180 Dropout (0.5)	**Fully Connected**	Nodes #: 180 Dropout (0.5)
Fully Connected	Nodes #: the number of classes	**Fully Connected**	Nodes #: the number of classes

2.2. 3D-CNN

The 3D-CNN can learn a deeper feature cube through a 3D kernel and is mainly used for analyzing video or image change. In particular, that hyperspectral data is an image cube with high spectral dimensions can lead to the applicability of 3D-CNN for HSIC. Its 3D convolution filter can help interpret the enormous spectral information of hyperspectral imagery. Table 2 represents the model architectures of the 3D-CNN built in this study. Similar to the 2D-CNN, the PCA-CNN uses the same model, while a different model is used in the original hyperspectral data. The size of the 3D convolution filter in this paper is $3 \times 3 \times K$, where K denotes the filter size of the spectral dimension, which should be set appropriately according to the model structure. For the Original-CNN, we construct a deeper architecture to optimize more spectral parameters.

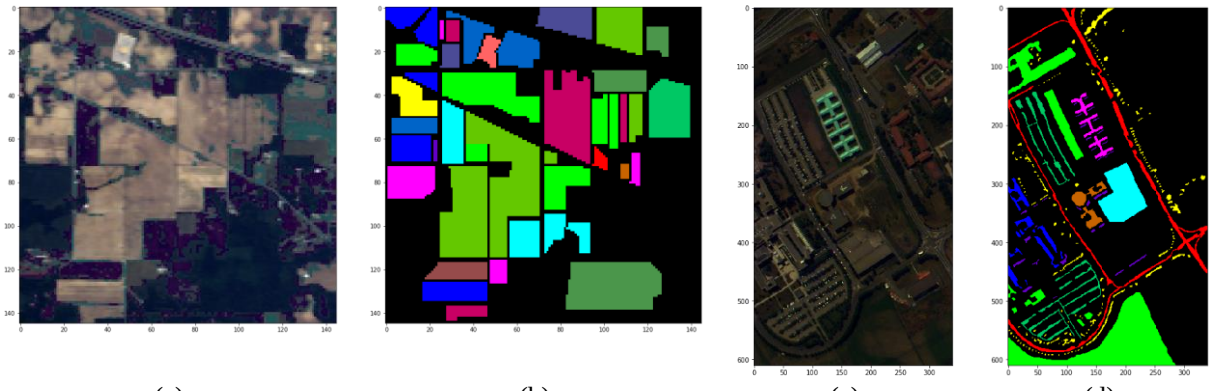
Table 2 3D-CNN model architecture

PCA-3D-CNN		Original-3D-CNN	
Conv3D	(3,3,7) filter#:5 Zero Padding BN	**Conv3D**	(3,3,9) filter#:3 Zero Padding BN
Conv3D	(3,3,5) filter#:15 BN	**MaxPooling**	Kernel: (1,1,2)
Conv3D	(3,3,3) filter#:45 BN	**Conv3D**	(3,3,7) filter#:6 BN
Fully Connected	Nodes #: 180 Dropout (0.5)	**MaxPooling**	Kernel: (1,1,2)
Fully Connected	Nodes #: the number of classes	**Conv3D**	(3,3,5) filter#:9 Zero Padding BN
		Conv3D	(3,3,3) filter#:12 BN
		Fully Connected	Nodes #: 270 Dropout(0.25)
		Fully Connected	Nodes #: 180 Dropout (0.25)
		Fully Connected	Nodes #: the number of classes

3. Experimental results and discussion

3.1. Data

In this study, we use two popular hyperspectral datasets with different land cover conditions. The first dataset is the Indian pines (IP) hyperspectral imagery gathered by Airborne Visible Infrared Imaging Spectrometer (AVIRIS). The imagery consists of 145×145 pixels and 220 spectral bands ranging from $0.4\ \mu m\ to\ 2.5\ \mu m$. Figure 1(a) and 1(b) illustrate a true color image and ground-truth image from the IP dataset, respectively. The dataset includes 16 classes mostly representing vegetation covers. The second dataset is the Pavia University (PU) hyperspectral imagery acquired by Reflective Optics System Imaging Spectrometer (ROSIS). The imagery consists of 610×340 pixels and 104 spectral bands ranging $from\ 0.43\ \mu m\ to\ 0.86\ \mu m$. Fig. 1(c) and 1(d) also show the true color image and ground-truth image from the PU dataset, respectively. The dataset includes 9 classes.



(a) (b) (c) (d)

Figure 1 Hyperspectral datasets used in this study: (a) true color image of IP, (b) ground-truth of IP, (c) true color image of PU, (d) ground-truth of PU

We split the IP and PU datasets into training and test samples, and the number of training samples are fixed at 4000 to enable the quantitative comparison between the two sites. The training samples are further split into training and validation samples at a ratio of 4:1. The validation data are classified by the trained model for each epoch to check if the model is overfitted.

3.2. Experimental design

PCA is first applied to the two datasets to achieve dimensionality reduction. The compressed datasets are used to train the 2D-CNN and 3D-CNN models. The trained models are then used to classify the test samples. We increase the number of PC bands and compare the classification results. Considering the randomness of the learning process and sampling, the experiments are repeated ten times under the same conditions, and the average values of the results, except for the upper and lower 10% are used to compare the models. In addition, the results of classification using the original imagery are compared with the results of the PCA-CNN.

3.3. Results

Tables 3~6 list the overall results representing the test accuracy and average time of each experiment. All values are the trimmed averages of ten runs under the same conditions. To remove the variability of the epochs when considering the training time, we analyze the training time per epoch (s/epoch).

Table 3 Classification results of the 2D-CNN in IP dataset

PC bands	Test Accuracy	Training time	PC bands	Test Accuracy	Training time	PC bands	Test Accuracy	Training time
1	59.272%	3.95541	**8**	88.622%	4.164303	**40**	92.974%	4.764624
2	77.814%	3.859872	**9**	88.634%	4.325336	**50**	94.782%	5.086493
3	83.96%	3.84	**10**	90.506%	4.25603	**60**	93.496%	5.126323
4	85.402%	3.932787	**15**	92.74%	4.222758	**70**	93.97%	5.308274
5	91.748%	3.891231	**20**	92.748%	4.461368	**80**	93.118%	5.128304
6	88.782%	3.840269	**25**	94.226%	4.657486	**90**	93.34578%	5.324848
7	88.404%	4.352944	**30**	92.864%	4.692681	**Original**	91.782%	15.83879

Table 4 Classification results of the 2D-CNN in PU dataset

PC bands	Test Accuracy	Training time	PC bands	Test Accuracy	Training time	PC bands	Test Accuracy	Training time
1	72.942%	4.223332	**8**	94.953%	4.604753	**40**	97.093%	4.905299
2	89.016%	4.110097	**9**	95.128%	4.036593	**50**	97.276%	5.335518
3	89.945%	4.191614	**10**	96.914%	4.310066	**60**	97.062%	5.186912
4	93.046%	4.331318	**15**	96.90%	4.434026	**70**	97.03%	5.536213
5	94.073%	4.202385	**20**	97.018%	4.622949			
6	95.036%	4.396774	**25**	97.320%	4.913668			
7	94.269%	4.377177	**30**	97.050%	4.954998	**Original**	96.624%	12.74241

Table 5 Classification results of the 3D-CNN in IP dataset

PC bands	Test Accuracy	Training time	PC bands	Test Accuracy	Training time	PC bands	Test Accuracy	Training time
7	89.5%	3.742201	**20**	95.82%	9.429098	**60**	98.648%	23.23326
8	91.334%	5.07809	**25**	96.434%	11.84459	**70**	98.764%	26.06963
9	91.742%	5.943251	**30**	96.846%	12.10583	**80**	98.956%	29.37223
10	92.228%	6.184954	**40**	97.81%	16.16366	**90**	98.918%	32.14787
15	95.106%	7.799459	**50**	98.442%	19.51984	**Original**	95.704%	54.29377

Table 6 Classification results of the 3D-CNN in PU dataset

PC bands	Test Accuracy	Training time	PC bands	Test Accuracy	Training time	PC bands	Test Accuracy	Training time
7	96.484%	3.670954	**20**	98.883%	9.375724	**60**	98.900%	23.35722
8	96.888%	5.097154	**25**	98.955%	10.82272	**70**	98.658%	26.39879
9	97.000%	5.926387	**30**	98.956%	12.36203			
10	98.329%	6.034626	**40**	99.014%	15.56481			
15	98.854%	7.32512	**50**	98.921%	18.68645	**Original**	98.595%	27.6109

2D-CNN and 3D-CNN

Figure 2 visually shows the averages of the test accuracy and training time for the 2D-CNN of IP dataset (a) and PU dataset (b). For the two datasets, the test accuracy increases as the number of the first few PC bands increases, then converges to a specific value at a certain PC band. For the IP dataset, the results show a stable accuracy of more than 93% from 50 PC bands, which can explain the upper 99.7% variance of the original spectral information (Table 7). Pre-processed dataset with the first ten PC bands in the PU dataset shows almost 97% test accuracy and no longer meaningfully increasing classification results. The first ten PC bands in the PU dataset can explain the upper 99.8% variance, which is similar to the IP dataset.

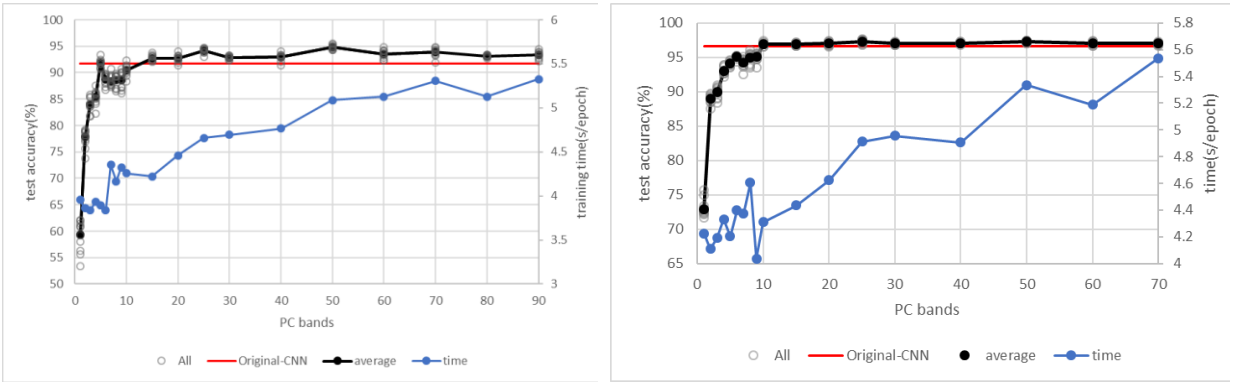
Table 7 Explained variance ratio for each PC bands

IP	**PC bands**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
	Variance ratio	68.49	92.02	93.51	94.33	95.03	95.55	95.95	96.31	96.62	96.91
	PC bands	**15**	**20**	**25**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
	Variance ratio	98.01	98.65	99.00	99.24	99.54	99.72	99.83	99.90	99.94	99.97
PU	**PC bands**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
	Variance ratio	58.31	94.41	98.85	99.15	99.36	99.54	99.66	99.73	99.78	99.81
	PC bands	**15**	**20**	**25**	**30**	**40**	**50**	**60**	**70**		
	Variance ratio	99.89	99.93	99.95	99.96	99.98	99.99	99.99	99.99		

Figure 3 represents the test accuracy and training time of the 3D-CNN. Similar to the 2D-CNN, the test accuracy converges on a specific PC band in the 3D-CNN (IP: 50 PC bands, PU: 15 PC bands) corresponding to certain explained variance ratio (IP: 99.727%, PU: 99.895%). In conclusion, the stable classification performance of the PCA-CNN can be attained when using specific PC bands explaining certain variance regardless of two datasets and model architectures.

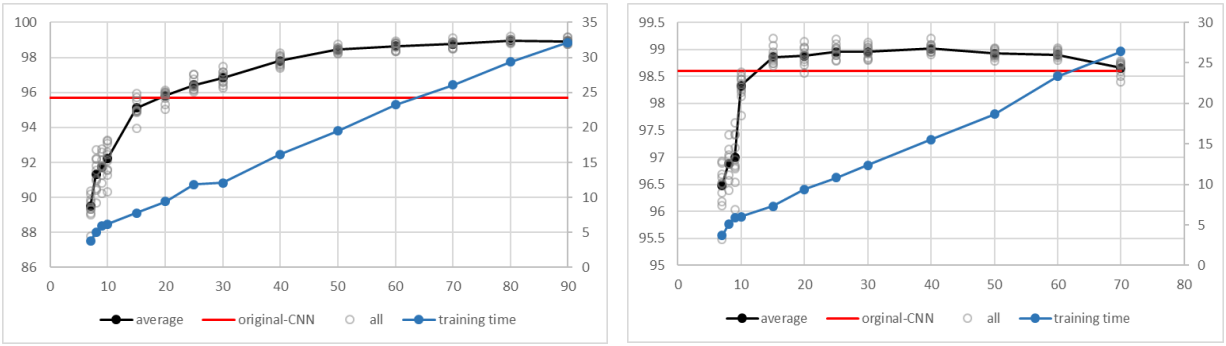
Comparison between the two models

The difference in the test accuracy of 3D-CNN with the Original-CNN is larger than that of the 2D-CNN in both datasets. The 3D kernel has more parameters than the 2D kernel, resulting in the 3D-CNN being more sensitive to the curse of dimensionality than the 2D-CNN. In the same context, the difference in the test accuracy at PU dataset is lower than in the IP dataset, where a higher spectral dimensionality further worsens the Hughes phenomenon. Thus, it can be concluded that the higher spectral dimensionality is and the more parameters in a model are, the more effective PCA is for reducing dimensionality.



(a) IP dataset (b) PU dataset

Figure 2 Result graphs for the 2D-CNN: (a) IP dataset, (b) PU dataset



(a) IP dataset (b) PU dataset

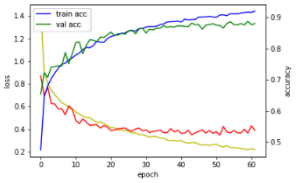
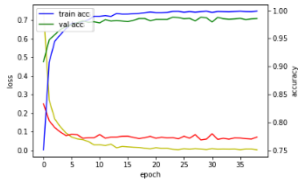
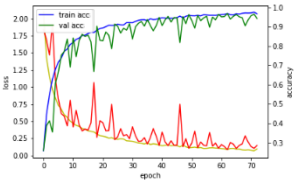
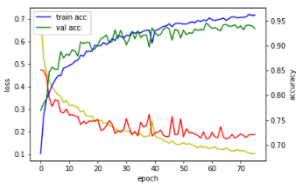
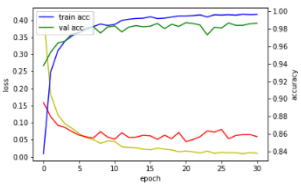
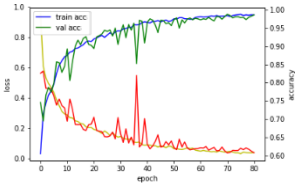
Figure 3 Result graphs for the 3D-CNN: (a) IP dataset, (b) PU dataset

Training time of the 2D-CNN, 3D-CNN and Original-CNN

After the first few PC bands for the PU dataset, the PCA-CNN test accuracy is almost the same as that of the Original-CNN. Even for the IP dataset, the test accuracy of the PCA-CNN is comparable or superior to that of the Original-CNN from the first ten PC bands. This means that the first few PC bands contain enough spectral information to train CNN models with competitive performance to the original imagery. In addition, training time further highlights the significance of PCA for deep learning based HSIC. The four cases (2D-CNN and 3D-CNN in the IP and PU datasets) demonstrate that the training time in the meaningful minimum PC bands is much smaller than that in the Original-CNN. In addition to the training time per epoch, the total number of epochs is higher than that in the Original-CNN, which meant that the total training time could range from as little as 10 minutes up to 76 minutes. Thus, the application of PCA for CNN-based HSIC enables a more efficient model training without significant differences in the classification performance.

Model training history

The instability of the Original-CNN can be visually confirmed through the model training history as well as the quantitative values of the test accuracy. Figure 4 illustrates the training histories of the 2D-CNN and 3D-CNN for the IP and PU datasets. The figure includes the histories in the fewest PC bands with poor classification results (2D-CNN: two PC bands, 3D-CNN: seven PC bands), the minimum PC bands showing stable classification accuracies (IP: 15 PC bands, PU: 50 PC bands), and the Original-CNN. All graphs are depicted for the case of the highest number of epochs in each experiment under identical conditions. In the training history graph of the PCA-CNN, the validation loss and accuracy are less fluctuant for the fewest bands and minimum bands with good classification performance, while the validation accuracy in the fewest PC bands are fixed at too low values. However, the history graph of the Original-CNN is very unstable. This means that the relative lack of samples due to high spectral dimensionality makes the model overfit to the training data; hence, the validation is unstable.

	Fewest	**Minimum**	**Original**
2D-CNN in IP	 2 PC bands	 50 PC bands	 Original
2D-CNN in PU	 2 PC bands	 15 PC bands	 Original

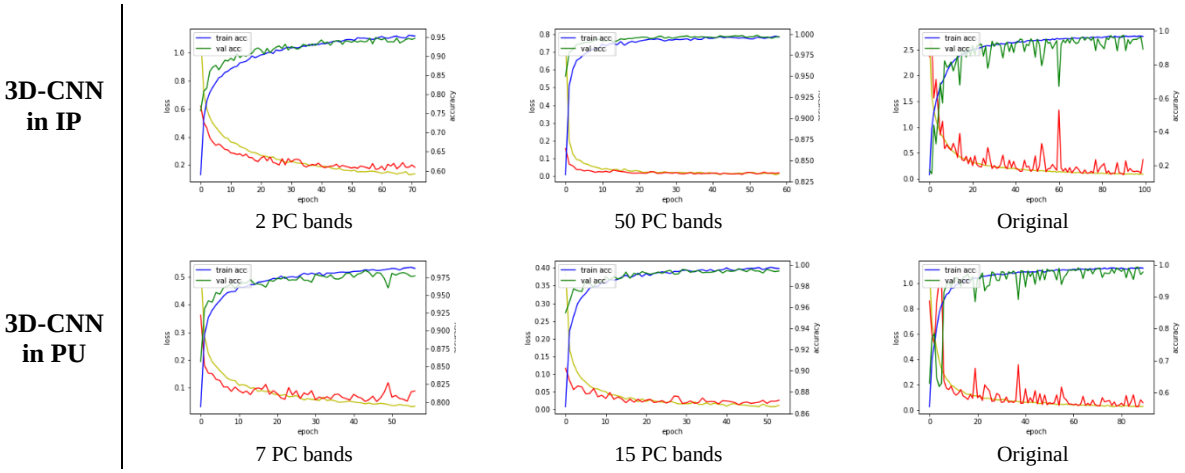


Figure 4 Training history of the considered CNN models

4. Conclusion

In this study, we analyzed the impact of PCA for CNN based HSIC. The classification accuracy increased in proportion to the size of the spectral dimension reduced using PCA. However, the accuracy did not increase further after a certain level. The minimum PC bands allowing to achieve a stable accuracy could explain similar variance ratios for the two tested datasets and models, which means that PC bands explaining nearly 99.8% of the variance can perform the most efficient classification. With regard to the efficiency, the impact of PCA was better than when using the original hyperspectral imagery reducing the computational complexity and time. In addition, PCA can be applied more efficiently to models with more parameters and data with larger spectral dimension due to the lack of relative samples.

In our future work, we will analyze the sensitivity of PCA to more complex and diverse models as well as other hyperspectral images with different cover conditions. Furthermore, the effect of the increasing number of samples on the applicability of PCA can be analyzed.

ACKNOWLEDGEMENTS

This research was supported by the research and development program funded by the Spatial Information Research Institution of Korea Land and Geospatial Informatix Corporation.

5. Reference

Agarwal, A., El-Ghazawi, T., El-Askary, H., & Le-Moigne, J., 2007, December. Efficient hierarchical-PCA dimension reduction for hyperspectral imagery. In 2007 IEEE International Symposium on Signal Processing and Information Technology (pp. 353-356). IEEE.

Bajorski, P., 2011. Statistical inference in PCA for hyperspectral images. IEEE Journal of selected topics in Signal Processing, 5(3), 438-445.

Chen, Y., Lin, Z., Zhao, X., Wang, G., & Gu, Y., 2014. Deep learning-based classification of hyperspectral data. IEEE Journal of Selected topics in applied earth observations and remote sensing, 7(6), 2094-2107.

Chen, Y., Jiang, H., Li, C., Jia, X., & Ghamisi, P., 2016. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. IEEE Transactions on Geoscience and Remote Sensing, 54(10), 6232-6251.

Cheng, Q., Varshney, P. K., & Arora, M. K., 2006. Logistic regression for feature selection and soft classification of remote sensing data. IEEE Geoscience and Remote Sensing Letters, 3(4), 491-494.

Fauvel, M., Benediktsson, J. A., Chanussot, J., & Sveinsson, J. R., 2008. Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles. IEEE Transactions on Geoscience and Remote Sensing, 46(11), 3804-3814

Goodfellow, I., Bengio, Y., & Courville, A., 2016. Deep learning. MIT press.

He, M., Li, B., & Chen, H., 2017, September. Multi-scale 3d deep convolutional neural network for hyperspectral image classification. In 2017 IEEE International Conference on Image Processing (ICIP) (pp. 3904-3908). IEEE.

Hu, W., Huang, Y., Wei, L., Zhang, F., & Li, H., 2015. Deep convolutional neural networks for hyperspectral image classification. Journal of Sensors, 2015.

Jensen, J. R., & Lulla, K., 1987. Introductory digital image processing: a remote sensing perspective.

Li, Y., Zhang, H., & Shen, Q., 2017. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. Remote Sensing, 9(1), 67.

Liang, H., & Li, Q., 2016. Hyperspectral imagery classification using sparse representations of convolutional neural network features. Remote Sensing, 8(2), 99.

Lillesand, T., Kiefer, R. W., & Chipman, J., 2015. Remote sensing and image interpretation. John Wiley & Sons.

Makantasis, K., Karantzalos, K., Doulamis, A., & Doulamis, N., 2015, July. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (pp. 4959-4962). IEEE.

Mei, X., Pan, E., Ma, Y., Dai, X., Huang, J., Fan, F., ... & Ma, J., 2019. Spectral-Spatial Attention Networks for Hyperspectral Image Classification. Remote Sensing, 11(8), 963.

Rodarmel, C., & Shan, J., 2002. Principal component analysis for hyperspectral image classification. Surveying and Land Information Science, 62(2), 115-122.

Samaniego, L., Bárdossy, A., & Schulz, K., 2008. Supervised classification of remotely sensed imagery using a modified $ k $-NN technique. IEEE Transactions on Geoscience and Remote Sensing, 46(7), 2112-2125.

Slavkovikj, V., Verstockt, S., De Neve, W., Van Hoecke, S., & Van de Walle, R., 2015, October. Hyperspectral image classification with convolutional neural networks. In Proceedings of the 23rd ACM international conference on Multimedia (pp. 1159-1162). ACM.

Tarabalka, Y., Benediktsson, J. A., & Chanussot, J., 2009. Spectral–spatial classification of hyperspectral imagery based on partitional clustering techniques. IEEE Transactions on Geoscience and Remote Sensing, 47(8), 2973-2987.

Zhang, H., Li, Y., Zhang, Y., & Shen, Q., 2017. Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network. Remote Sensing Letters, 8(5), 438-447.

Zhao, W., & Du, S., 2016. Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. IEEE Transactions on Geoscience and Remote Sensing, 54(8), 4544-4554.