Semisupervised Classification Based on Tensor Convolutional Neural Network for Hyperspectral Images

Fuyin Ye\textsuperscript{a,1} and Zhao Chen\textsuperscript{b,1}\*  
\textsuperscript{1}School of Computer Science and Technology, Donghua University, Shanghai, China  
\textsuperscript{a}617823053@qq.com; \textsuperscript{b}\* contact author: chenzhao@dhu.edu.cn

Abstract. Deep neural network has been extensively applied to hyperspectral image (HSI) classification and shown promising performance recently. However, those popular deep learning models scarcely consider low-rank features of high-dimensional HSIs lying in intrinsic data subspaces. Besides, their results largely rely on numerous labeled samples, which are expensive and time-consuming. To address these issues, we propose a Three Dimensional Tensor Convolutional Neural Network (3DTCNN) for spectral-spatial low-rank feature representation and a Semisupervised Classifier based on 3DTCNN (S-TCNN) for classification with limited labeled samples. 3DTCNN integrates tensor decomposition with convolutional neural network to extract discriminative spectral-spatial low-rank features. It also combines multiple supervised discriminators and unsupervised clustering modules to exploit labeled and unlabeled samples. Experimental results show that S-TCNN outperforms several classic supervised classifiers and state-of-the-art semisupervised classifiers on one benchmark hyperspectral data set, Indian Pines. When there are only 3 samples with prior labels per class, the proposed model achieves 98.2\%, in terms of overall accuracy. Therefore, the design of 3DTCNN is proved feasible while the efficacy and efficiency of S-TCNN are validated.

1. Introduction  
With the development of deep learning, fully supervised deep Convolutional Neural Network (CNN) is often used for hyperspectral image classification. For example, Three Dimensional Convolutional Neural Network (3DCNN) can simultaneously process the spatial and spectral information of hyperspectral images, achieving higher classification accuracy than CNN. However, the fully supervised learning model (especially the large deep network) usually needs a large number of prior labeled samples for training, while it takes a lot of money and time to manually label pixels of hyperspectral images in advance. In order to reduce the dependence on manual labeling, semisupervised learning provides an effective solution. Semisupervised learning uses both labeled and unlabeled samples in the training process, so as to obtain higher classification accuracy than the fully supervised algorithm with the same labeled training samples. Its advantages are mainly in two aspects: it not only reduces the demand for a large number of labeled samples, but also can further explore the relationships between data \cite{1, 2}. However, "dimension disaster" is easily caused because of high dimensionality of hyperspectral images and the lack of manual labeling. Therefore, it is necessary to reduce the dimension of data, extract low rank features, compress the amount of data and reduce the computational complexity, as well as improve the classification accuracy. To address these issues, we propose a semisupervised classifier based on Three Dimensional Tensor Convolutional Neural Network (3DTCNN). The model combines tensor decomposition neural network \cite{3} with Convolution Neural Network and uses tensor low rank decomposition, convolution, pooling and other operations to express low rank spectral-spatial features. At the same time, the model combines several supervised judgment and unsupervised clustering modules to fully learn labeled samples and unlabeled samples achieving high-precision semisupervised classification. The main contributions of this paper are as follows: 1) Combining tensor decomposition neural network with three-dimensional convolution neural network, a new three-dimensional tensor convolution network (3DTCNN) is proposed. 2)
3DTCNN is applied to semisupervised classification of hyperspectral images, and S-TCNN is proposed to reduce the reliance of the model on prior samples. 3) By involving labeled data and unlabeled data in training, S-TCNN can effectively express the spectral-spatial low-rank features of hyperspectral images, enhance the distinguishability of features, improve the learning ability and generalization of the model, and thus improve the classification accuracy. Experiments have proved that S-TCNN can achieve high classification accuracy with a small amount of label data, and its performance is better than the state-of-the-art semisupervised classification model and the classic fully-supervised classification model for hyperspectral images.

2. Background knowledge

Let \( X \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M} \) be an \( M \)-order tensor, where \( I_m (m=1,2,\cdots,M) \) is the dimension of the \( m \)-th order. Expand \( X \) in mode \( m \) and obtain the \( m \)-th order-flattened matrix

\[
X_{(m)} = mat_{(m)}(X) \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{m-1} \times I_{m+1} \times \cdots \times I_M},
\]

(1)

where \((X_{(m)})_{i_1i_2\cdots i_{m-1}i_{m+1}\cdots i_M} = (X)_{i_1i_2\cdots i_{m-1}i_{m+1}\cdots i_M},\) and \(i_1\cdots i_M \in \bigcup_{i=1}^{m-1} I_i + \bigcup_{i=m+1}^{M} I_{i} + \cdots + \bigcup_{i=1}^{M} I_{i} \).

Vectorize the tensor \( X \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M} \) and arrive at

\[
g = vec(X) = vec(X_{(1)}) \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}.
\]

(2)

the mode \( m \) multiplication of tensor \( X \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M} \) and matrix \( U \in \mathbb{R}^{I_m \times J} \) is defined as follows,

\[
Q = X \times_m U = X_{(m)} \times \left( U_{(m)} \right)^\tau,
\]

(3)

where \( Q = \sum_{i_1}^{I_1} \sum_{i_2}^{I_2} \cdots \sum_{i_M}^{I_M} (X)_{i_1i_2\cdots i_{m-1}i_{m+1}\cdots i_M} \times (U)_{i_{m+1}\cdots i_M} \times (U)_{i_{m+2}\cdots i_M} \times \cdots \times (U)_{i_M} \).

the generalization of principal component analysis (PCA) on higher-order tensors is Tucker decomposition [4]. Given the tensor \( X \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M} \) and the projection matrix \( U^{(m)} \in \mathbb{R}^{I_m \times I_m} \) (\( J_m \leq I_m \)), then the Tucker decomposition expression of \( X \) is

\[
X = G_{(1)} \times \cdots \times \left( U^{(M)} \right)_{(M)} + E
\]

(4)

3. Methodology

Hyperspectral images have high dimensionality and insufficient manual labeling, which causes difficulties in feature extraction and fully supervised classification. However, semisupervised learning can obtain good classification results by virtue of only a small number of labeled samples, which is suitable for learning with insufficient prior labels. In order to effectively get the spectral-spatial low-rank features of hyperspectral images and improve the accuracy of pixel-level classification, this paper designs a Three Dimensional Tensor Convolutional Neural Network (3DTCNN) and proposes a semisupervised classifier (S-TCNN) based on 3DTCNN. The framework is shown in Figure 1.

![Figure 1. Framework of S-TCNN.](image-url)
research [5, 6], it can be seen that convolution, pooling and tensor-matrix multiplication can all be expressed as tensor contraction. Therefore, Tucker decomposition can be combined with convolution, pooling and fully connected to construct a tensor convolutional neural network (TCNN). Through three-dimensional convolution, pooling, fully connected and nonlinear mapping, 3DTCNN can learn deep abstract feature models. And combined with Tucker decomposition, 3DTCNN can extract spectral-spatial low-rank features. As a result, it exhibits strong feature representation capabilities and can also compress data and reduce the computational complexity of the model.

Hyperspectral image $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, where $I_1$, $I_2$ and $I_3$ are the number of rows, columns and bands, respectively. Take the neighborhood for each pixel number (if necessary, add zeros to $\mathbf{X}$) to construct the sub-tensor $\mathbf{Y}_{I_2}^{(0)} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ as the input sample of 3DTCNN, where $i_{1,2,\ldots, I_1, I_2, I_3}$ is the preset value. When $\mathbf{Y}_{I_2}^{(0)}$ is a training sample, the one-hot encoding label $\mathbf{y} \in \mathbb{R}^C$ of the sample can be obtained from the ground truth, where $C$ is the number of categories.

Suppose the total number of layers in the network is $L$, and the number of layers is $l$ ($l=1,2,\ldots,L$). If the $l$-th layer is a three-dimensional convolutional layer or a pooling layer, the expression is as follows,

$$\mathbf{Y}^{(l)} = \sigma \left( \mathbf{X}^{(l)} \times_{1,2,3} \mathbf{W}^{(l)} \right).$$

where $\mathbf{Y}^{(l)} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is the output feature of the current layer. $\mathbf{X}^{(l)} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is the input sample, composed of the sub-tensor $\mathbf{B}^{(i-l)} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ of the output feature $\mathbf{Y}^{(i-i)} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ of the previous layer (when $l=1$ , $\mathbf{Y}^{(i-i)} = \mathbf{Y}_{I_2}^{(0)}$, $I_{1}^{(i-i)} = I_{3}$. $I_{4}^{(i-i)} = 1$ ).

$\mathbf{W}^{(l)} \in \mathbb{R}^{K_1 \times K_2 \times K_3 \times I_4}$ is the weight of the convolutional template or pooling template, $\sigma(*)$ is a nonlinear mapping function. In 3DTCNN, all nonlinear mapping functions are set as Swish functions.

Suppose the stride is $S_{n}^{(l)} \in \mathbb{N}^{+}$ and the padding dimension is $P_{n}^{(l)} \in \mathbb{Z}$ , then

$I_{n}^{(l)} = I_{n}^{(l-1)} - K_{n}^{(l)} + P_{n}^{(l)} \times S_{n}^{(l)+1}, n = 1,2,\ldots,n^{(l-1)}$, where $n=1,2,3$.

If the $l$-th layer is the Tucker decomposition layer, the expression is as follows,

$$\mathbf{Y}^{(l)} = \sigma \left( \mathbf{X}^{(l)} \times_{1} \mathbf{W}^{(l,1)} \times_{2} \mathbf{W}^{(l,2)} \times_{3} \mathbf{W}^{(l,3)} \times_{4} \mathbf{W}^{(l,4)} \right).$$

where $\mathbf{Y}^{(l)} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is the output low-rank feature, $\mathbf{X}^{(l)} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is the input sample (the previous layer output $\mathbf{Y}^{(l-1)}$), $\mathbf{W}^{(l,m)} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$ is the subspace projection matrix, and $\sigma(*)$ is the nonlinear mapping function. Taking the reference data set Indian Pines as an example, the structure of 3DTCNN is shown in Table 1. Let the input sample dimension be 23×23×200, and the output feature dimension is 512×1. It can be seen that 3DTCNN can represent low-rank spectral-spatial features and compress the amount of data.

3.2. Semisupervised classification

In order to fully learn unlabeled samples and labeled samples, S-TCNN combines multiple supervised decision devices and unsupervised clustering models [2] to achieve semisupervised learning. As shown in Figure 2, the main modules are as follows: K-means (generating unsupervised clustering labels), two-layer fully connected network U (learning of all sample features), two-layer fully connected network S (feature learning of labeled samples), one-layer fully connected network F (feature fusion of labeled samples). Suppose that K-means obtains $C$ clusters, and the cluster label of each hyperspectral pixel $\mathbf{x}_i$ is $y_{mi} \in \mathbb{R}^C$ ($i=1,2,\ldots,I_2$), then the loss function of the fully connected network U is.
\[ \ell_u = -\frac{1}{l} \sum_{i=1}^{l} y_{ui}^T \log(y_{ui}). \]  

(7)

where \( y_{ui} \) is the estimated label of the sub-tensor sample \( Y_i^{(0)} \) of the network U. Suppose there are \( l_i \) samples \( Y_i^{(0)} \) with manual labels, and the definition in section 2.1 shows that the label of this sample is \( y \in \mathbb{R}^C \), then the loss function of the fully connected network S is

\[ \ell_i = -\frac{1}{l_i} \sum_{i=1}^{l_i} y_i^T \log(y_i). \]

(8)

where \( y_i \) is the estimated label of the sub-tensor sample \( Y_i^{(0)} \) of the network S. Similarly, the loss function of the fully connected network F is

\[ \ell_f = -\frac{1}{l_f} \sum_{i=1}^{l_f} y_{fi}^T \log(y_{fi}). \]

(9)

where \( y_{fi} \) is the estimated label of the sub-tensor \( Y_i^{(0)} \) of the network F. The loss function of 3DTCNN is the overall objective function of S-TCNN, which is calculated as follows

\[ \ell = \ell_f + \ell_i + \ell_u. \]

(10)

| Number | Layer         | Input                | Output               | Weight                        | Stride |
|--------|---------------|----------------------|----------------------|-------------------------------|--------|
| 1      | 3D-Conv       | 23x23x1              | 1x23x23x64           | 23x1x8x63x64                  | [1,1,3]|
| 2      | 3D-Conv       | 1x23x63x64           | 1x23x3x64x64         | 1x1x3x64x64                   | [1,1,1]|
| 3      | 3D-Conv       | 1x23x63x64           | 1x1x3x64x64         | 1x1x3x64x64                   | [1,1,1]|
| 4      | 1D-Tucker     | 1x23x63x64           | 1x23x62x64         | 63x62(f_i^{(0)} + f_i^{(-1)}) | /      |
| 5      | 3D-Conv       | 1x23x62x64           | 1x1x30x128          | 1x23x3x64x128                 | [1,1,2]|
| 6      | 3D-Conv       | 1x1x30x128           | 1x1x28x128          | 1x1x3x128x128                 | [1,1,1]|
| 7      | 3D-Conv       | 1x1x28x128           | 1x1x28x128          | 1x1x3x128x128                 | [1,1,1]|
| 8      | 1D-Tucker     | 1x1x28x128           | 1x1x27x128          | 28x27(f_i^{(0)} + f_i^{(-1)}) | /      |
| 9      | 3D-Conv       | 1x1x27x128           | 1x1x13x256          | 1x1x3x128x256                 | [1,1,2]|
| 10     | 3D-Conv       | 1x1x13x256           | 1x1x11x256          | 1x1x3x256x256                 | [1,1,1]|
| 11     | 3D-Conv       | 1x1x11x256           | 1x1x11x256          | 1x1x3x256x256                 | [1,1,1]|
| 12     | 1D-Tucker     | 1x1x11x256           | 1x1x10x256          | 1x10x(f_i^{(0)} + f_i^{(-1)}) | /      |
| 13     | 3D-Conv       | 1x1x10x256           | 1x1x1x512           | 1x1x10x256x512                 | [1,1,1]|
| \     | Flatten       | 1x1x1x512            | 512x1(K^{(0)}x1)     | /                             | /      |

4. Methodology

4.1. Data set

We conduct experiments on one benchmark hyperspectral data set, Indian Pines, which contains 21025 pixels and 200 spectral bands. In this dataset, 10249 pixels from 16 classes are labeled in total. Figure 2 shows the ground truth of the Indian Pines.

4.2. Setup

In order to verify the superiority of S-TCNN in classification performance, it is compared with the classic SVM and six state-of-the-art methods SGL, LCMR SC-MK, EPF, LBP and IFRF [1]. The evaluation indicators are overall accuracy (OA), class accuracy (CA), average class accuracy (AA) and Kappa coefficient. The conditions of the experiment strictly follow the work of Sellars et al. [1]: 3, 5, 7, 10, and 15 samples are randomly selected from each category of the data set as "manually labeled
samples" (or "labeled samples"), and the rest are unlabeled samples. The labeled and the unlabeled are both involved in training. Similar to SGL, S-TCNN estimates labels of the initially unlabeled samples while being trained by them. The Adam optimizer is used for stochastic gradient descent. The initial value of learning rate is 0.001 and exponentially decayed with a factor 0.9. For the data set, number of clusters ($C$) of K-means is set to 100. Each experiment was repeated 10 times.

**Figure 2.** Ground truth of the Indian Pines data set, whereas each identical color represents one class and class names are shown on the right side of the map.

### 4.3. Results

The visual classification result of S-TCNN is shown in Figure 3. It can be seen that when 3, 5, and 15 pixels are selected as the labeled samples for each category, S-TCNN can obtain a classification map with good visual effects. When the amount of labeled samples changes, the overall accuracy of each algorithm is shown in Figure 4. When 10 pixels of each category are selected as the labeled samples, the overall accuracy, average accuracy and Kappa coefficient of each algorithm are shown in Figure 5. It can be seen that the classification performance of S-TCNN is better than the current most advanced semi-supervised classifier SGL and other comparison algorithms. When there are only 3 labeled samples of each category, the overall accuracy of S-TCNN is as high as 98.2%, while the overall accuracy of SGL is only 78.7%. The proposed S-TCNN model can achieve high classification accuracy and stable performance. It shows that with good learning ability, S-TCNN can express spectral-spatial low-rank features of hyperspectral images and distinguish easily confusing pixels.

**Figure 3.** (a) Ground truth of the Indian Pines images and (b) classification results of S-TCNN using 3 manually annotated samples per class (OA: 98.24%, AA: 98.99%, Kappa: 0.98), (c) 5 manually annotated samples per class (OA: 99.59%, AA: 99.50%, Kappa: 0.99), and (d) 15 manually annotated samples per class (OA: 99.98%, AA: 99.99%, Kappa: 0.99). In this figure, each identical color represents one class, which is the same as Figure 2.

**Figure 4.** OA (%) of 8 algorithms for Indian Pines under different training conditions.
Figure 5. CA, AA and Kappa coefficients (%) of 8 algorithms for Indian Pines with 10 labeled samples per class.

5. Conclusion
A semisupervised classifier based on tensor convolutional neural network (S-TCNN) is proposed, which can effectively classify the pixels of hyperspectral images. By combining three-dimensional convolution and tensor decomposition, S-TCNN effectively expresses the spectral-spatial low-rank features of hyperspectral images, removes redundant data. And multiple fully-supervised classifiers and unsupervised clustering models are combined to fully learn labeled samples and unlabeled samples, which means that S-TCNN can achieve high classification accuracy with a small amount of labeled samples. Experiments show that the classification results of S-TCNN are better than classic classifiers and a variety of advanced semisupervised classification algorithms. Our current research is limited to the spectral dimension. In future research, we will make a deeper and more diversified combination of convolution, pooling, full connection and tensor decomposition in different dimensions.

Acknowledgments
This research has been financially supported by grants from the National Natural Science Foundation of China (No. 61702094), the Dawn Project of Shanghai Municipal Education Commission (No. 18CG38), the Young Scientists’ Sailing Project of Science and Technology Commission of Shanghai Municipal (No. 17YF1427400).

References
[1] Sellars P, Aviles-Rivero A, Schönlieb C. Superpixel Contracted Graph-Based Learning for Hyperspectral Image Classification [J]. IEEE Trans. Geosci. Remote Sens., vol. 58, no. 6, pp. 4180−4193, June 2020, doi: 10.1109/TGRS.2019.2961599.
[2] Zhang J, Wei W, Zhang L, et al. Improving Hyperspectral Image Classification with Unsupervised Knowledge Learning [C]. 2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 2019, pp. 2722−2725, doi: 10.1109/IGARSS.2019.8898323.
[3] Chien J, Bao Y. Tensor-Factorized Neural Networks [J]. IEEE Transactions on Neural Networks & Learning Systems, 2017:1998−2011.
[4] Cichochi A, Zdunek R, Phan A, et al. Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multiway Data Analysis and Blind Source Separation, Wiley, 2009.
[5] Chen Z, Yu X. A novel tensor network for tropical cyclone intensity estimation [J]. IEEE Trans. Geosci. Remote Sens., accepted, Aug. 2020.
[6] Yu X, Chen Z, Chen G, et al. A tensor network for tropical cyclone wind speed estimation [C]. 2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 2019.