



**HAL**  
open science

# Hyperspectral Image Processing based on Tensorial Methods

Qiaoqiao Sun

► **To cite this version:**

Qiaoqiao Sun. Hyperspectral Image Processing based on Tensorial Methods. Signal and Image processing. Ecole Centrale Marseille, 2021. English. NNT : 2021ECDM0008 . tel-03643897

**HAL Id: tel-03643897**

**<https://theses.hal.science/tel-03643897v1>**

Submitted on 17 Apr 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

École Doctorale : Ecole Doctorale Physique et Sciences de la Matière (ED352)

Laboratoire de l'Institut Fresnel

## THÈSE DE DOCTORAT

pour obtenir le grade de

DOCTEUR de l'ÉCOLE CENTRALE de MARSEILLE

Discipline : Optique, Photonique et Traitement d'Image

### Hyperspectral Image Processing based on Tensorial Methods

par

**Qiaoqiao SUN**

**Directeur de thèse : Prof. Salah BOURENNANE**

*Soutenue le 15 octobre 2021*

*devant le jury composé de :*

Prof. Yide WANG	Ecole Polytechnique de l'Université de Nantes	Rapporteur
Prof. Ahmed BOURIDANE	Northumbria University, Newcastle, Royaume-Uni	Rapporteur
Prof. Eric MOREAU	Université de Toulon	Examinateur
Dr. Xuefeng LIU	Qingdao University of Science and Technology	Examinateur
Prof. Salah BOURENNANE	Ecole Centrale de Marseille	Directeur de thèse



# Acknowledgments

I would like to express my deepest appreciation to my supervisor, Prof. Salah Bourennane, for receiving me to start my PhD journey in France, and for his vast knowledge and very valuable advice throughout my PhD study. His tolerance, patient guidance, continuous encouragement, enthusiastic engagement and his never ending stream of ideas have benefited me a lot.

A big thanks to the reviewers, Prof. Yide WANG and Prof. Ahmed BOURIDANE, for accepting to be the reviewers, Prof. Eric MOREAU and Dr. Xuefeng LIU to be the examiners of my thesis. In addition, I would like to express my deepest appreciation to Prof. Yide WANG for helping to apply for the project and Dr. Xuefeng LIU for her continued help during my master's and Ph.D., which has a profound impact on my life.

I also would like to thank my super lovely and kind friends that I have met in Marseille. Thank them for accompanying me over the past three years, so I didn't feel lonely and helpless. I am honored to know them during my PhD study, especially Chunyang Li, Jiaqi Yang, Fangfang Yang, Na Shao and Jingke Hou.

I would also like to thank the members of the GSM (Groupe Signaux Multidimensionnels) team of Institut Fresnel. Although a short time spent with them, I am still very happy to know them and cherish every opportunity to meet. Special thanks to Xiaoxi Pan for patiently answering the various questions I encountered, Julien Wojak for his technical support, and Caroline Fossati for her kind help in the office.

Special thanks to the China Scholarship Council and École Centrale Marseille, who offered me the opportunity to study and live in France.

Finally, I would like to thank my grandparents, my parents, my brother, and all my family and friends, especially Bin Liu, for their love, support and encouragement all the time.



# Publications

- Journal papers

1. **Q. Sun**, X. Liu, and S. Bourennane. Unsupervised Multi-level Feature Extraction for Improvement of Hyperspectral Classification. *Remote Sensing*, vol. 13, no, 8, p. 1602, 2021.
2. **Q. Sun**, X. Liu, S. Bourennane, and B., Liu. Multiscale denoising autoencoder for improvement of target detection. *International Journal of Remote Sensing*, vol. 42, no. 8, pp. 3002 - 3016, 2021.
3. **Q. Sun**, and S. Bourennane. Hyperspectral image classification with unsupervised feature extraction. *Remote Sensing Letters*, vol. 11, no. 5, pp. 475 - 484, 2020.
4. X. Liu, **Q. Sun**, Y. Meng, M. Fu, and S. Bourennane. Hyperspectral image classification based on parameter-optimized 3D-CNNs combined with transfer learning and virtual samples. *Remote Sensing*, vol. 10, no, 9, p. 1425, 2018.

- Conference papers

1. **Q. Sun**, X. Liu, and S. Bourennane. Optimal Parameter Selection in Hyperspectral Classification Based on Convolutional Neural Network. *5th International Conference on Frontiers of Signal Processing (ICFSP)*, Marseille, France, 18 - 20 September, 2019.
2. **Q. Sun**, and S. Bourennane. Unsupervised feature extraction based on improved Wasserstein generative adversarial network for hyperspectral classification. *Multimodal Sensing: Technologies and Applications*, Munich, German, 23 - 28 Juin, 2019.



# Contents

<b>Acknowledgments</b>	<b>i</b>
<b>Résumé étendu</b>	<b>1</b>
<b>Abbreviations</b>	<b>13</b>
<b>Introduction</b>	<b>15</b>
<b>1 Supervised feature extraction based on 2D-CNN for hyperspectral classification</b>	<b>21</b>
1.1 Introduction . . . . .	21
1.2 Overview of 2D-CNN . . . . .	21
1.3 Hyperspectral classification based on 2D-CNN . . . . .	23
1.3.1 Optimal parameter selection based on 2D-CNN . . . . .	23
1.3.2 Data set description and assessment criteria . . . . .	24
1.3.2.1 Data set description . . . . .	24
1.3.2.2 Assessment criteria . . . . .	26
1.3.3 Experimental results . . . . .	27
1.3.3.1 Input size . . . . .	27
1.3.3.2 Network structure . . . . .	29
1.3.3.3 Number of units in the fully connected layer . . . . .	29
1.3.3.4 Activation function and pooling method . . . . .	30
1.3.3.5 Optimization method . . . . .	31
1.3.3.6 Batch size . . . . .	32
1.3.3.7 Number of convolutional kernels . . . . .	32
1.3.3.8 Number of epochs . . . . .	33
1.3.3.9 Comparison of classification results . . . . .	34
1.4 Conclusion . . . . .	36



<b>2</b>	<b>Supervised feature extraction based on 3D-CNN for hyperspectral classification</b>	<b>37</b>
2.1	Introduction . . . . .	37
2.2	Hyperspectral classification based on 3D-CNN . . . . .	37
2.2.1	Band selection . . . . .	39
2.2.1.1	Introduction of SIFT and DP algorithms . . . . .	39
2.2.1.2	Proposed FMDP method for band selection . . . . .	40
2.2.1.3	Experimental results of band selection . . . . .	43
2.2.2	Solutions with limited labeled samples . . . . .	47
2.2.2.1	Transfer learning . . . . .	48
2.2.2.2	Virtual samples . . . . .	50
2.2.2.3	3D-CNN with transfer learning and virtual samples . . . . .	50
2.3	Experimental results . . . . .	52
2.3.1	Details of 3D-CNN . . . . .	52
2.3.2	Details of 3D-CNN-TL . . . . .	52
2.3.3	Details of 3D-CNN-VS . . . . .	55
2.3.4	Details of 3D-CNN-TV . . . . .	56
2.3.5	Comparison of classification results . . . . .	57
2.4	Conclusion . . . . .	58
<b>3</b>	<b>Unsupervised feature extraction based on GAN for hyperspectral classification</b>	<b>59</b>
3.1	Introduction . . . . .	59
3.2	Overview of GAN . . . . .	60
3.3	Proposed unsupervised feature extraction method based on 3D-WGAN-GP . . . . .	62
3.3.1	Proposed dimensionality reduction method . . . . .	62
3.3.2	Details of proposed unsupervised feature extraction method . . . . .	64
3.3.3	Experimental results . . . . .	65
3.4	Conclusion . . . . .	69
<b>4</b>	<b>Unsupervised feature extraction based on CAE for hyperspectral classification</b>	<b>73</b>
4.1	Introduction . . . . .	73
4.2	Overview of AE . . . . .	74
4.3	Proposed multi-level feature extraction method . . . . .	76

4.3.1	Details of the proposed framework . . . . .	76
4.3.2	Experimental results . . . . .	78
4.3.2.1	Network Construction . . . . .	78
4.3.2.2	Comparison and analysis of experimental results . . .	78
4.4	Proposed 3D-M <sup>2</sup> CAE framework for small target feature extraction and classification . . . . .	89
4.4.1	Details of proposed framework . . . . .	90
4.4.2	Experimental results . . . . .	91
4.4.2.1	Data set description . . . . .	91
4.4.2.2	Network construction . . . . .	92
4.4.2.3	Result analysis of $HSI_a$ data set . . . . .	93
4.4.2.4	Result analysis of Pavia University data set . . . . .	98
4.4.2.5	Visual observation and comparison . . . . .	99
4.5	Conclusion . . . . .	101
<b>5</b>	<b>Deep learning models for improvement of target detection</b>	<b>105</b>
5.1	Introduction . . . . .	105
5.2	Spectral reconstruction for denoising and target detection . . . . .	106
5.2.1	Methodology . . . . .	106
5.2.1.1	n mode unfolding . . . . .	107
5.2.1.2	Denoising autoencoder . . . . .	107
5.2.2	Target detection combined a multiscale denoising autoencoder	108
5.2.3	Spectral reconstruction by denoising autoencoder . . . . .	108
5.2.4	Proposed model for target detection . . . . .	109
5.2.5	Experimental results . . . . .	110
5.2.5.1	Experiments on simulated data . . . . .	111
5.2.5.2	Experiments on real-world data . . . . .	114
5.2.5.3	Small target detection . . . . .	117
5.3	Unsupervised segmentation for small target detection . . . . .	118
5.3.1	Unsupervised segmentation . . . . .	118
5.3.2	Experimental results . . . . .	119
5.4	Conclusion . . . . .	122
	<b>Conclusion and future works</b>	<b>123</b>
	<b>List of Figures</b>	<b>127</b>

List of Tables	133
Bibliography	135

# Résumé étendu

Une image hyperspectrale provient de l'observation d'une même scène dans de nombreuses longueurs d'onde espacées de quelques nanomètres et contiguës. L'intérêt est d'obtenir un spectre que l'on peut assimiler comme continu. En imagerie hyperspectrale on considère que chaque matériau, présent dans la scène, reflète des ondes électromagnétiques de manière spécifique. Ainsi, Le spectre de chaque pixel diffère suivant le ou les matériaux présents et on peut par exemple discriminer deux objets de même couleur composés de matériaux différents. La représentation d'une image hyperspectrale sous forme de tenseur d'ordre trois ou d'un tableau tridimensionnel est la plus utilisée, où les dimensions impliquées sont de même nature : deux dimensions spatiales et une dimension spectrale, comme le montre la figure 1. Ceci nous permet de prendre en compte simultanément l'information spatiale et l'information spectrale.

En raison des caractéristiques des HSI, l'imagerie hyperspectrale est largement utilisée dans de nombreux domaines [1,2], tels que l'exploration minérale [3], l'agriculture [4,5], l'environnement [6]. Dans ces applications on s'intéresse plus particulièrement à la classification et la détection de cibles dans les images hyperspectrales.

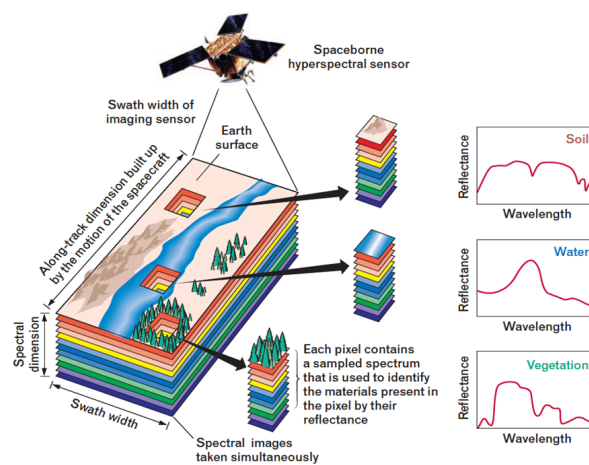


Figure 1: Schéma de principe de l'imagerie hyperspectrale.

Cependant, comme les HSI contiennent généralement des centaines de bandes spectrales, la charge de calculs augmente rapidement et les techniques conventionnelles ne sont plus efficaces pour le traitement de données de grande dimension, principalement en raison de la malédiction de la dimensionnalité [7]. La réduction de dimension spectrale ou l'extraction de caractéristiques pertinentes de l'image hyperspectrale sont des moyens efficaces de résoudre le problème de la malédiction de la dimensionnalité. L'extraction des caractéristiques est l'une des étapes les plus importantes pour la classification et la détection. Les données hyperspectrales sont des données typiques non linéaires, qui contiennent de nombreuses informations spectrales et spatiales. Les méthodes traditionnelles d'extraction de caractéristiques sont basées sur des transformations linéaires, telles que l'analyse factorielle (FA), l'analyse en composantes principales (PCA), qui sont limitées pour l'extraction de caractéristiques non linéaires et des caractéristiques à un niveau profond [8].

Au cours de ces dernières années, considéré comme une branche importante de l'apprentissage automatique [9–12], l'apprentissage en profondeur a attiré une large attention en raison de ses bonnes performances pour l'analyse de données et l'extraction de caractéristiques [13, 14]. En extrayant les caractéristiques des données d'entrée du bas vers le haut du réseau, les modèles d'apprentissage en profondeur peuvent former les caractéristiques de haut niveau et non linéaires [15]. Certains modèles d'apprentissage profond sont utilisés à la classification en imagerie hyperspectrale et quelques résultats sont obtenus, tels que l'autoencodeur (AE) [16], les réseaux de croyances profondes (DBN) [17], les réseaux de neurones récurrents [18, 19], les réseaux de neurones résiduels (ResNet) [20] et les réseaux de neurones convolutifs (CNN) [21].

## **Objectif et contributions de ce travail de thèse**

Compte tenu de l'énorme potentiel de l'apprentissage en profondeur en traitement d'images et de nombreux modèles pouvant gérer des données multidimensionnelles de manière flexible, les modèles d'apprentissage en profondeur sont principalement utilisés pour le traitement des HSI dans cette thèse. Les cibles sont débruitées, classées et détectées en extrayant entièrement les caractéristiques spectrales et spatiales et en analysant les données hyperspectrales.

Étant donné que les opérations basées sur la convolution peuvent gérer des données multidimensionnelles de manière flexible, 2D-CNN et 3D-CNN sont respectivement utilisés pour l'extraction et la classification des caractéristiques. 2D-

CNN montre un grand potentiel dans la préservation de la structure spatiale des cibles. 3D-CNN peut prendre directement des données 3D en entrée, ce qui permet d’exploiter les informations spectrales et spatiales en même temps. De plus, une méthode de réglage des paramètres est proposée pour sélectionner les paramètres à leur tour selon le principe de la variable unique. Le réseau avec des paramètres optimaux nous aide à obtenir de meilleurs résultats. Cependant, qu’il soit 2D-CNN ou 3D-CNN, le réseau est optimisé en minimisant l’erreur entre la sortie et l’étiquette, ce qui signifie que CNN nécessite un grand nombre d’échantillons étiquetés pour garantir ses performances. Malheureusement, les échantillons étiquetés dans les HSI sont limités et la collecte d’échantillons étiquetés prend du temps et demande beaucoup de travail. Pour résoudre ce problème, les méthodes du transfert de l’apprentissage et de la génération d’échantillons virtuels sont introduites. Le transfert de l’apprentissage permet à un système de reconnaître et d’appliquer les connaissances et les compétences acquises dans de précédents domaines/tâches à de nouveaux domaines/tâches [22]. Si on dispose d’une HSI (données sources) avec suffisamment d’échantillons étiquetés et avec les mêmes caractéristiques spatiales que la HSI à classer (données cibles), alors l’apprentissage par transfert peut être utilisé pour réduire le besoin d’échantillons étiquetés de données cibles. Si les données sources ne sont disponibles, on peut générer des échantillons virtuels à partir des échantillons originaux des données cibles pour résoudre le problème d’insuffisance d’échantillons pour l’apprentissage.

Afin de mieux comparer les résultats de classification obtenus par différents modèles, les valeurs de précision globale (OA) sont principalement utilisées pour évaluer la précision de la classification. Pour une meilleure comparaison visuelle, les cartes de classification de l’ensemble des données de l’Université de Pavie obtenues par 2D-CNN, 3D-CNN et 3D-CNN avec le transfert d’apprentissage et en utilisant des échantillons virtuels (3D-CNN-TV) sont illustrées dans la figure 2 .

On peut voir à partir de la figure 2 que la carte de classification de 2D-CNN a plus de pixels mal classés par rapport aux cartes de classification obtenues avec 3D-CNN, ce qui montre que 3D-CNN a de plus grandes capacités à exploiter pleinement les informations . En outre, la méthode proposée 3D-CNN-TV permet d’obtenir la carte la plus claire et la valeur la plus élevée de OA, ce qui montre que cette méthode 3D-CNN-TV a un grand potentiel en imagerie hyperspectrale pour la classification.

L’extraction non supervisée de caractéristiques qui n’implique pas d’échantillons étiquetés est un autre moyen pour résoudre le problème de manque de données étiquetées. Le réseau antagoniste génératif (GAN) est entraîné de manière antagoniste

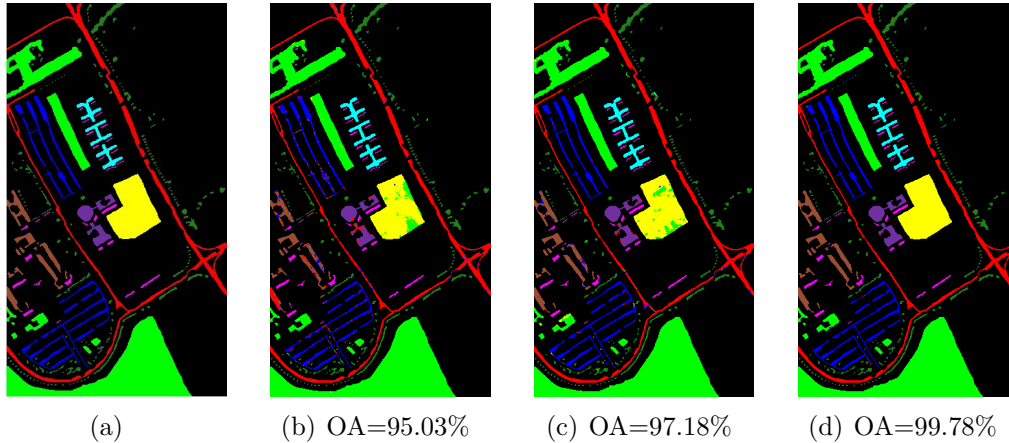


Figure 2: Cartes de classification de l’Université de Pavie obtenues par différentes méthodes: (a) Vérité terrain, (b) 2D-CNN, (c) 3D-CNN, (d) 3D-CNN-TV.

ne nécessitant aucun échantillon étiqueté et il est l’un des réseaux de l’apprentissage non supervisé les plus prometteurs [23]. L’AE apprend une représentation des données d’entrée via un encodeur, puis décode la représentation pour reconstruire les données d’entrée. La représentation à faible dimension des données par l’AE peut être utilisée comme les caractéristiques des données d’entrée. Du fait que les paramètres de l’AE sont optimisée en minimisant l’erreur entre les données reconstruites et les données d’entrée, aucune étiquette n’est impliquée, on parle alors d’un modèle non supervisé. Par conséquent, des méthodes d’extraction de caractéristiques non supervisées basées sur GAN et AE sont proposées pour résoudre le problème lié à la limitation du nombre des échantillons étiquetés. Étant donné que l’opération de convolution 3D est effectuée dans le domaine spatial et spectral, et qu’elle a un grand potentiel pour extraire entièrement les caractéristiques spectrales et spatiales, le générateur et le discriminateur de 3D-GAN sont construits sur des sous-réseaux de convolution entièrement 3D et de déconvolution 3D. L’autoencodeur convolutif 3D conçu (3D-CAE) est composé de couches convolutives 3D et de couches de déconvolution 3D.

Une explosion ou une disparition de gradient peuvent être provoquées lors de la formation du GAN. Pour résoudre ce problème, le Wasserstein GAN 3D (3D-WGAN) est amélioré en ajoutant une pénalité de gradient pour intégrer la contrainte de Lipschitz, dénommé 3D-WGAN-GP, est utilisé pour créer une technique d’extraction de caractéristiques non supervisée. Lorsque le 3D-WGAN-GP est bien entraîné, nous pensons que le générateur a appris la distribution des données réelles et que le discriminateur a une forte capacité d’extraction de caractéristiques. Le

discriminateur peut convertir des données de grande dimension en données de faible dimension, ce qui est cohérent avec notre objectif d'extraction de caractéristiques. Par conséquent, le discriminateur optimisé peut être transféré en tant qu'extracteur de caractéristiques.

Les encodeurs d'un 3D-CAE ont une structure hiérarchique de bas en haut, et ils sont comme des pyramides de caractéristiques. Les couches inférieures correspondent principalement aux informations, telles que les bords, la texture et les contours, et les couches supérieures correspondent principalement aux informations sémantiques. Afin d'exploiter pleinement les représentations apprises, une méthode d'extraction de caractéristiques multi-niveaux non supervisée basée sur un 3D-CAE est proposée. De plus, les caractéristiques multi-niveaux sont directement obtenues à partir de différentes couches codées de l'encodeur optimisé, ce qui est plus efficace que l'apprentissage de plusieurs réseaux. L'utilisation complète des informations détaillées dans les couches inférieures et des informations sémantiques dans les couches supérieures peut apporter des avantages complémentaires et améliorer les résultats de la classification.

Afin de vérifier l'efficacité des méthodes proposées, les résultats de classification basés sur différentes caractéristiques (caractéristiques de 3D-WGAN-GP, caractéristiques à un seul niveau de 3D-CAE et caractéristiques à plusieurs niveaux de 3D-CAE) sont comparés. Plus les résultats de la classification sont bons, meilleures sont les caractéristiques correspondantes. La méthode machines à vecteurs de support (SVM) est utilisée comme classifieur dans cette expérience.

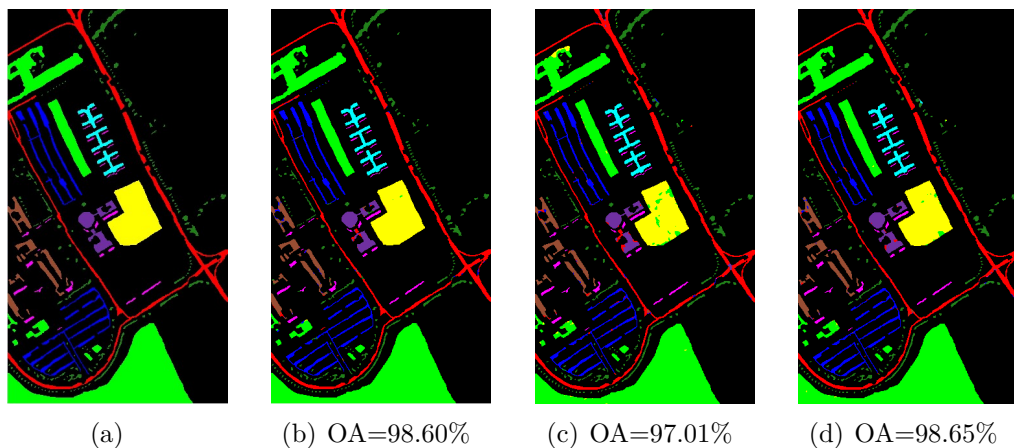


Figure 3: Cartes de classification de l'Université de Pavie obtenues par différentes méthodes: (a) Vérité terrain, (b) 3D-WGAN-GP, (c) 3D-CAE avec fonctionnalités à un seul niveau, (d) 3D-CAE avec fonctionnalités multi-niveaux.

On peut voir sur la figure 3 que les valeurs de OA obtenues par 3D-WGAN-GP



et 3D-CAE sont élevées. En général, les cartes de classification de 3D-WGAN-GP et 3D-CAE avec des caractéristiques multi-niveaux ont moins de pixels mal classés, ce qui prouve que les méthodes d'extraction de caractéristiques non supervisées proposées ont des perspectives prometteuses pour les HSI.

Étant donné que les petites cibles sont plus sensibles à la taille de l'entrée et ont moins d'échantillons, l'analyse de petites cibles est plus difficile dans les HSI. Pour trouver un équilibre entre les différentes cibles et améliorer les résultats de la classification des petites cibles, un nouveau réseau multi-taille et multi-modèle basé sur 3D-CAE, appelé 3D-M<sup>2</sup>CAE, est proposé. Trois 3D-CAE avec différentes tailles d'entrée centrées sur le pixel observé sont utilisées pour construire le réseau et extraire les caractéristiques. De plus, afin de réduire le temps d'apprentissage, le réseau est construit et entraîné de manière progressive en utilisant le transfert d'apprentissage [22, 24, 25]. Les poids des couches intermédiaires du dernier 3D-CAE sont transférés du précédent 3D-CAE optimisé, ce qui accélère et facilite l'apprentissage du réseau. Bénéficiant de cette méthode d'entraînement, les caractéristiques d'une même cible sont obtenues de manière efficace aux différentes tailles du réseau.

Pour mieux évaluer les performances de la méthode proposée, des méthodes d'extraction de caractéristiques supervisées basées sur DBN et une méthode d'extraction de caractéristiques non supervisée basée sur FA sont considérées pour la comparaison. Les données de l'image  $HSI_a$  qui contiennent des cibles de petites dimensions sont utilisées. Les cartes de classification obtenues par les différentes méthodes sont présentées dans la figure 4.

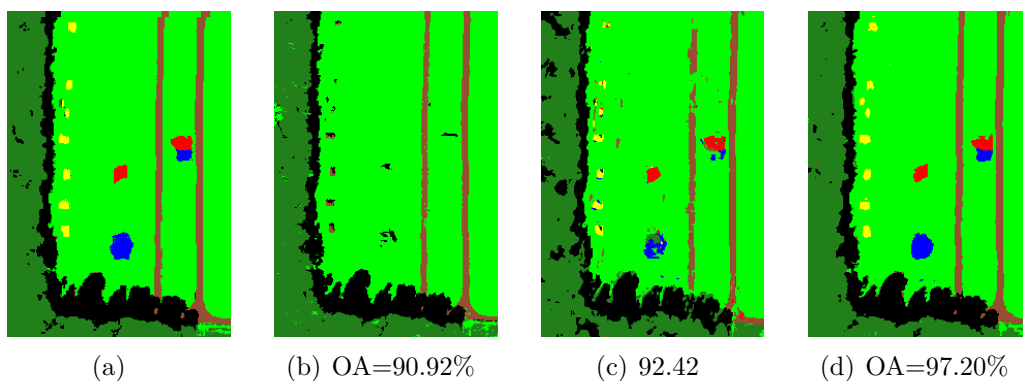


Figure 4: Cartes de classification du  $HSI_a$  obtenues par différentes méthodes: (a) Vérité terrain, (b) FA, (c) DBN, (d) Proposé 3D-M<sup>2</sup>CAE.

En comparant la vérité terrain et les cartes de classification de la figure 4, nous pouvons constater que presque toutes les petites cibles sont mal classées dans la

figure 4 (b). Les résultats de classification dans la Figure 4 (c) sont améliorés par rapport à la Figure 4 (a), mais il y a encore beaucoup de pixels jaunes et bleus qui ne sont pas correctement classés. Lorsque le 3D-M<sup>2</sup>CAE proposé est utilisé pour obtenir des caractéristiques, la carte de classification correspondante (figure 4 (d)) est la plus claire et il y a peu de pixels mal classés dans les régions rouge, bleue et jaune, ce qui montre que la méthode proposée a un grand potentiel dans l'extraction de caractéristiques et la classification de petites cibles.

Dans tout ce qui précède, nous nous sommes principalement intéressés à l'extraction des caractéristiques et la classification des HSI. La détection des cibles est également l'une des importantes applications en traitement HSI. Avec une signature spectrale connue appelée aussi modèle spectral, les détecteurs peuvent déterminer si la cible est présente ou pas en comparant le modèle spectral aux pixels d'une scène. Cependant, les HSI souffrent toujours des variations spectrales causées par le bruit ou l'environnement, ce qui augmente les variations intra-classes et dégrade les performances des détecteurs. Il est essentiel d'obtenir une précision de détection élevée même des cibles dans des scènes bruitées. Ainsi, nous souhaitons améliorer les résultats de détection de cibles en utilisant des détecteurs existants en améliorant la qualité du spectre de la cible et en exploitant les caractéristiques invariantes du spectre. L'autoencodeur de débruitage (DAE) est introduit pour reconstruire le spectre et augmenter la robustesse spectrale. Afin de faire en sorte que les spectres reconstruits contiennent autant d'informations que possible, un réseau d'autoencodeur de débruitage à plusieurs échelles dénommé MSDAE est conçu pour améliorer la détection de cibles dans les HSI. Le spectre d'entrée est codé à différentes échelles pour obtenir un ensemble de représentations d'entrée, puis ils sont décodés pour les fusionner pour obtenir le spectre final reconstruit.

Pour vérifier l'efficacité de la méthode proposée, les performances de détection et de débruitage de la méthode proposée sont testées sur l'image bruitée  $\mathbf{R}$  qui est obtenue en ajoutant du bruit aléatoire  $\mathbf{N}$  à l'image  $\mathbf{H}$ , soit  $\mathbf{R} = \mathbf{H} + \mathbf{N}$ . Un bruit blanc gaussien centré (WGN) et un bruit multiplicatif (MPN) uniformément distribué et de moyenne nulle sont utilisés pour simuler le bruit aléatoire. Courbes d'efficacité du récepteur (ROC) de la cible 4 en HSI<sub>b</sub> avec un rapport signal sur bruit (SNR) de 20 dB débruité par le filtre de Wiener (WF), et block-matching and 3D filtering (BM3D), et denoising convolutional neural network (DnCNN) sont représentés sur la figure 5.

On peut constater sur la figure 5, que les méthodes basées sur DAE et DnCNN obtiennent des valeurs de probabilité de détection ( $P_d$ ) plus élevée avec la même

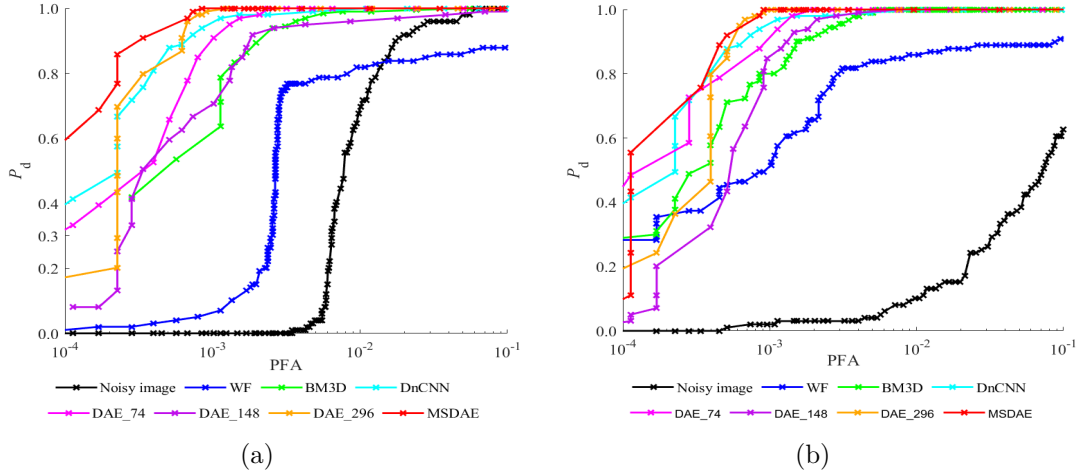


Figure 5: Courbes ROC de  $HSI_b$  sous différents bruits avec un SNR de 20 dB: (a) WGN, (b) MPN.

probabilité de fausse alarme (PFA), ce qui montre que les modèles d'apprentissage profond ont un grand potentiel pour réduire le bruit et conserver les informations utiles. De plus, avec la méthode MSDAE proposée sont obtenus de meilleurs résultats en présence des deux types de bruit WGN ou MPN.

## Contributions du travail de cette thèse

Cette thèse est consacrée à l'utilisation de méthodes basées sur les tenseurs, en se concentrant sur les méthodes d'apprentissage profond, pour traiter les HSI. Les principaux apports se résument comme suit :

- Considérant que CNN a un grand potentiel dans l'extraction de caractéristiques et peut bien préserver la structure spatiale de la cible, 2D-CNN est introduit pour extraire les caractéristiques parmi les données pour la classification hyperspectrale. Cependant, les performances de CNN sont toujours influencées par les réglages des paramètres. Pour obtenir les paramètres optimaux pour la classification HSI, nous proposons une méthode de classification basée sur un 2D-CNN avec réglage des paramètres (2D-CNN-PT). Les paramètres du réseau sont réglés à leur tour selon le principe de la variable unique et un ensemble de paramètres optimaux peut enfin être obtenu pour améliorer les performances du réseau.

- Les HSI sont représentées par des tableaux ou tenseurs 3D et comme les données multidimensionnelles peuvent être directement appliquées à l'entrée de 3D-CNN. Par conséquent, un 3D-CNN est introduit pour exploiter pleinement les informations spectrales et spatiales des données hyperspectrales et améliorer la précision

de la classification. Cependant, le manque d'échantillons étiquetés de HSI limite les performances des 3D-CNN. Pour résoudre ce problème, une méthode améliorée basée sur des 3D-CNNs combinés à un transfert d'apprentissage ou à des échantillons virtuels est proposée. Les poids dans les couches inférieures du réseau cible sont transférés d'un autre 3D-CNN bien entraîné sur une HSI (données sources) avec suffisamment d'échantillons et avec les mêmes caractéristiques spatiales que les données cibles. De plus, des échantillons virtuels générés à partir des échantillons originaux des données cibles sont utilisés pour augmenter encore le nombre d'échantillons. Le transfert d'apprentissage ou les échantillons virtuels peuvent atténuer le problème posé par la limitation des échantillons étiquetés rencontré dans de nombreuses situations.

- Le GAN est entraîné de manière antagoniste ne nécessitant aucun échantillon étiqueté, ce qui est un modèle d'entraînement non supervisé. Pour pallier la limitation des échantillons étiquetés, un extracteur de caractéristiques non supervisé est conçu sur la base d'un transfert d'apprentissage et d'un 3D-WGAN-GP. Le 3D-WGAN-GP ajoute une pénalité de gradient pour appliquer la contrainte de Lipschitz, ce qui peut résoudre le problème de l'explosion et de la disparition des gradients. De plus, compte tenu des caractéristiques spectrales et spatiales des HSI, le réseau est conçu sous forme 3D, ce qui permet d'exploiter pleinement les informations contenues dans les données hyperspectrales.

- L'AE peut être optimisé en minimisant l'erreur entre les données reconstruites et les données d'entrée sans utilisation de données étiquetées, c'est un modèle non supervisé typique. Pour apprendre simultanément les informations spectrales et spatiales des cibles et ne pas dépendre des échantillons étiquetés qui sont souvent limités, un nouveau réseau d'extraction de caractéristiques à plusieurs niveaux non supervisé basé sur un 3D-CAE est proposé. En outre, l'encodeur d'un 3D-CAE est une structure hiérarchique du bas en haut, et les caractéristiques extraites sont en forme d'une pyramide. Pour exploiter pleinement les avantages du réseau optimisé, les caractéristiques multi-niveaux obtenues à partir des couches codées avec différentes échelles et résolutions sont proposées, ce qui est plus efficace que d'utiliser plusieurs réseaux pour les obtenir. De plus, un réseau multi-taille et multi-modèle basé sur 3D-CAE, appelé 3D-M<sup>2</sup>CAE, est proposé pour l'extraction et la classification de petites cibles. La conception et l'optimisation du réseau reposent sur une croissance progressive et un transfert d'apprentissage. Bénéficiant de cette méthode d'entraînement, les caractéristiques d'une même cible sont obtenues de manière efficace aux différentes tailles et le réseau proposé présente une grande robustesse

vis-à-vis des différentes cibles.

- En raison des variations spectrales causées par le bruit ou l'environnement, la variation intra-classe est importante, ce qui dégrade les performances des détecteurs, en particulier lorsque la taille de la cible est petite. Compte tenu de la grande capacité d'extraction de caractéristiques et de représentation des modèles d'apprentissage en profondeur, le DAE est introduit pour réduire le bruit et exploiter les informations invariantes pour la détection de cibles. De plus, pour extraire entièrement les caractéristiques des spectres d'origine, un modèle MSDAE est conçu pour incorporer des informations complémentaires dans le spectre final en fusionnant les spectres reconstruits à partir de représentations à différentes échelles, ce qui fournit des informations plus complexes et des caractéristiques plus robustes pour une identification spectrale.

## Organisation de la thèse

Cette thèse est organisée en cinq chapitres.

**Chapter 1** présente une méthode supervisée d'extraction de caractéristiques basée sur 2D-CNN pour la classification en imagerie hyperspectrale. Considérant que les performances du réseau sont fortement influencées par les paramètres, un réseau 2D-CNN-PT est conçu. Neuf principaux paramètres sont réglés tour à tour selon le principe de la variable unique. Les résultats expérimentaux sur deux images réelles montrent que les réglages de paramètres appropriés sont d'une grande importance pour améliorer les performances du réseau et la précision de la classification.

**Chapter 2** décrit une méthode d'extraction de caractéristiques supervisée basée sur 3D-CNN pour la classification hyperspectrale. 3D-CNN peut traiter de manière flexible des données multidimensionnelles et extraire les informations spectrales et spatiales au même temps. En raison des caractéristiques des HSI, une importante charge de calculs est nécessaire. Par conséquent, une nouvelle méthode de réduction de dimension spectrale est proposée pour réduire la dimensionnalité des HSI. De plus, pour résoudre le problème des échantillons insuffisants et améliorer la classification des HSI, un réseau 3D-CNN-TV basé sur un 3D-CNN avec un transfert d'apprentissage et la génération des échantillons virtuels est proposé. Les résultats expérimentaux montrent que l'apprentissage par transfert ou les échantillons virtuels peuvent améliorer la précision de la classification, et le réseau 3D-CNN-TV donne les meilleurs résultats de classification par rapport aux autres méthodes considérées.

**Chapter 3** introduit une méthode d'extraction de caractéristiques non super-

visée basée sur un 3D-WGAN-GP. Dans un premier temps, certaines connaissances de base du GAN sont présentées. Ensuite, une nouvelle méthode de réduction de dimensionnalité basée sur des convolutions de  $1 \times 1$  et un pooling de  $1 \times 1$  est proposée pour obtenir des données de dimensionnalité inférieure contenant des caractéristiques plus abstraites et de haut niveau. Ensuite, un réseau d'extraction de caractéristiques non supervisé pour la classification hyperspectrale basé sur 3D-WGAN-GP et sur un transfert d'apprentissage est conçu pour pallier la limitation des échantillons étiquetés. Enfin, des expériences sont réalisées sur des données réelles pour évaluer les performances de la méthode proposée, et les résultats expérimentaux montrent la faisabilité et l'efficacité de la méthode proposée.

**Chapter 4** présente les deux autres réseaux d'extraction de caractéristiques non supervisés basés sur 3D-CAE. Dans le premier réseau, des caractéristiques multi-niveaux sont proposées pour contenir des informations détaillées et des informations sémantiques en même temps. Les caractéristiques multi-niveaux proposées sont directement obtenues à partir de différentes couches codées de l'encodeur optimisé, ce qui nous permet de tirer pleinement parti du réseau bien entraîné et à améliorer encore la qualité des caractéristiques. Dans le second réseau, un 3D-M<sup>2</sup>CAE composé de trois 3D-CAE avec une taille d'entrée différente est proposé pour équilibrer les différentes cibles et améliorer les résultats de classification des petites cibles. Bénéficiant de la méthodologie d'entraînement progressif et du transfert d'apprentissage, l'optimisation de 3D-M<sup>2</sup>CAE est facilitée. Les résultats expérimentaux montrent que les deux réseaux conçus sont prometteurs pour l'extraction non supervisée de caractéristiques.

**Chapter 5** propose un modèle MSDAE pour l'amélioration de la détection des cibles. Selon les caractéristiques spectrales, les pixels peuvent être classés en cible ou en fond d'image. Cependant, en raison des variations spectrales causées par le bruit ou l'environnement, la variation intra-classe est importante, ce qui dégrade les performances des détecteurs, en particulier lorsque la taille des cibles sont petites. Pour résoudre ce problème, la méthode DAE est proposée pour reconstruire les spectres et exploiter les informations invariantes pour la détection de cibles. De plus, pour extraire entièrement les caractéristiques des spectres d'origine, un modèle MSDAE est conçu pour incorporer des informations complémentaires dans le spectre final obtenu par fusion des spectres reconstruits à partir des représentations aux différentes échelles, ce qui fournit des informations plus complexes et des caractéristiques plus robustes pour une identification spectrale. Les résultats sur des données simulées et réelles montrent que la méthode proposée peut non seulement

améliorer la détection des cibles, mais également a un grand potentiel pour préserver les petites cibles.

# Abbreviations

1D	One-dimensional
2D-CNN	Two-dimensional convolutional neural network
2D-CNN-PT	2D-CNN with parameter tuning
3D-CAE	Three dimensional convolutional autoencoder
3D-CNN	Three-dimensional convolutional neural network
3D-CNN-TL	3D-CNN with transfer learning
3D-CNN-VS	3D-CNN with virtual samples
3D-CNN-TV	3D-CNN with transfer learning and virtual samples
3D-WGAN	3D Wasserstein generative adversarial network
3D-WGAN-GP	Improved 3D-WGAN adding gradient penalty
AA	Average accuracy
AE	Autoencoder
ACE	Adaptive coherence/cosine estimator
BCS	Band column selection
AMF	Adaptive matched filter
BM3D	Block-matching and 3D filtering
CAE	Convolutional autoencoder
CNN	Convolutional neural network
DAE	Denoising autoencoder
DBN	Deep belief network
DnCNN	Denoising convolutional neural network



DP	Douglas-Peucker
FA	Factor analysis
GAN	Generative adversarial network
HSI	Hyperspectral images
$\kappa$	Kappa coefficient
MNBS	Minimum noise band selection
MPN	Multiplicative noise
MSDAE	Multiscale denoising autoencoder
OA	Overall accuracy
PCA	Principal component analysis
$P_d$	Probability of detection
PFA	Probability of false alarm
ROC	Receiver operating characteristic
SAE	Stacked autoencoder
SAM	Spectral angle mapper
SIFT	Scale invariant feature transformation
SNR	Signal-to-noise ratio
SVM	Support vector machine
WF	Wiener filter
WGN	White Gaussian noise

# Introduction

Hyperspectral images (HSIs) are collected from the interested scene by spaceborne and airborne hyperspectral imagers with a lot of narrow electromagnetic bands as shown in Figure 6. Based on two-dimensional (2D) space domain, a third spectral dimension is added for each pixel. Thus, an HSI can be represented as a three-dimensional (3D) data block that contains not only spatial information but also spectral characteristics. Due to the characteristics of HSIs, hyperspectral imaging technology has been widely applied in many fields [1, 2], such as mineral exploration [3], agriculture [4, 5], environment management [6]. Hyperspectral classification and detection are two important techniques for these applications, and feature extraction is one of the most significant steps for classification and detection. Traditional feature extraction methods are unusually based on linear transformation, such as principal component analysis (PCA) [26] and independent component analysis (ICA) [27], which are not suitable for nonlinear hyperspectral data. Worse, most of the traditional feature extraction methods can only extract features in a shallow manner [28]. Therefore, effective feature extraction is one of the key processings to improve HSI classification and detection [29–32].

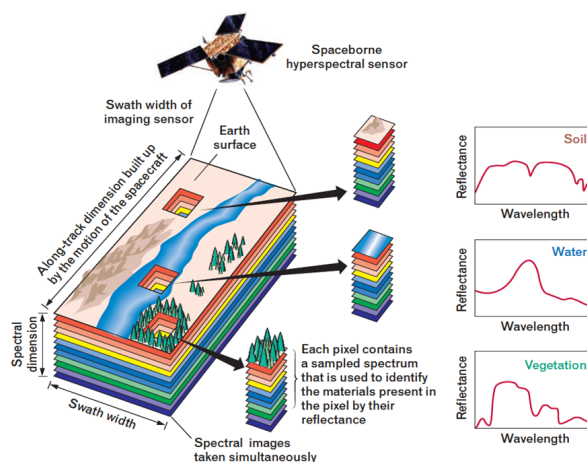


Figure 6: Schematic diagram of hyperspectral imaging.

Recently, as an important branch of machine learning [9–12], deep learning has attracted wide attention due to its strong capabilities in data analysis and feature extraction [13, 14]. By extracting features of the input data from the bottom to the top of the network, deep learning models can form the high-level abstract features [15]. Some deep learning models have been successfully applied to HSIs, such as autoencoder (AE) [16], deep belief network (DBN) [17], recurrent neural network (RNN) [18, 19], convolutional neural network (CNN) [21, 33, 34]. Among numerous deep learning models, CNN has attracted widespread attention due to its unique network structure and strong feature extraction capabilities [35].

HSIs are 3D tensor data and multidimensional data can be directly input into CNNs, which helps to preserve the original relevant information of the data and avoids complex data reconstruction [36–38]. Therefore, CNN has been introduced to extract high-level invariant features and improve the classification performance of HSIs [39, 40]. However, since the HSI usually contains hundreds of bands, the number of corresponding network parameters and the amount of calculation increase. Besides, sufficient training samples are needed to guarantee the performance of CNN. Unfortunately, labeled samples in HSIs are always limited [41–43]. Therefore, high dimensionality and insufficient labeled samples are two challenges in HSI processing. Dimensionality reduction can effectively reduce the computational difficulty. Some representative methods, for example transfer learning [25, 44, 45], manifold regularization based on semi-supervised learning [41, 43], and so on, have been studied to alleviate the problem of limited samples. The former method is suited for high-dimensional data structures, the latter being more suitable for ordinary images. Therefore, transfer learning is introduced and applied to HSIs.

Unsupervised feature extraction is another good way to get rid of limited labeled samples. In recent years, some deep learning models have been investigated for unsupervised feature extraction. Generative adversarial network (GAN) is trained in an adversarial way requiring no labeled samples. It has been one of the most promising unsupervised learning representatives [23]. In [46], a semi-supervised framework based on GAN is established for hyperspectral classification with a small number of labeled samples. But only spectral features are extracted, which are far from enough for classification. In [47], Zhang proposed a novel modified GAN whose generator and discriminator are designed in the form of fully deconvolutional network and fully convolutional network to extract the features without supervision. Nevertheless, only spatial information is taken as input when the modified GAN is trained, which can be treated as 2D convolution on multiple channels. Hyperspec-

tral data is a tensor data, which contains not only spatial information but also the spectral characteristics of the target. Fully mining the spectral-spatial features in HSIs is helpful for classifying the target. Considering 3D convolution operation is performed in space and spectrum, we want to design a framework based on three-dimensional generative adversarial network (3D-GAN) in which the generator and discriminator are built on fully 3D convolution and 3D deconvolution subnetworks to fully extract the spectral-spatial features with unsupervised learning for classification. In addition, the AE learns a representation for input data through an encoder and then decodes the representation to reconstruct data [48, 49]. The AE can be optimized by minimizing the error between the reconstructed data and the input data, and no labels are involved, which is a typical unsupervised model. Because of these characteristics of AE, some unsupervised feature extraction methods that are based on AE have been introduced in HSIs and achieved some results [50]. In [51], two variants based on stacked sparse AE are designed to the unsupervised spectral features learning and multiscale spatial features learning, respectively. The learned spectral and spatial features are stacked as a long feature vector embedded into a classifier for classification, which is more potential and robust in hyperspectral classification compared to traditional methods. However, the spectral features and spatial features are extracted separately. In [52], an unsupervised feature extraction method based on recursive AE is developed to produce high-level features. Some results have been obtained based on variants of AE, but often single-level features are considered, which affects feature performance. Therefore, we want to use AE-based models to further explore multi-scale features in HSIs.

In addition, target detection is also an important task for us. Each pixel in HSIs corresponds to a spectral curve and the spectral signature of the same category is similar, which enables identify the materials present in the pixel. Therefore, target detection can be treated as a binary classification task [53]. With a known spectral signature can also be called a spectral template, comparing the spectral template to the pixels in a scene can determine whether target is present or not. There are some detectors which are commonly used for target detection in HSIs [54], such as adaptive coherence/cosine estimator (ACE), adaptive matched filter, and spectral angle mapper. However, HSIs always suffer from spectral variations caused by noise or environment, which enlarges within-class variation and degrades the performance of detectors. It is essential to obtain high detection accuracy even targets in noisy scenes. Thus, we want to improve the target detection results using existing detectors by improving the quality of spectral signature and mining the

invariant features of the spectrum. To improve target detection results, denoising usually be done as a preprocessing step for noise removal, and then target detection is performed. Traditional denoising methods, such as PCA [55] and models based on Wiener filter (WF) [56], and block-matching and 3D filtering (BM3D) [57], have been successfully applied in image processing. However, the traditional denoising method is easy to face the problem of single task [58] or preserving small targets [59].

With the development of deep learning, some methods based on deep learning have been proposed for image denoising [60]. In [61,62], models based on denoising autoencoder (DAE) model is established for image denoising, which uses encoder to get the latent representation and then reconstructs it into the clean data through the decoder. In [63], feed-forward denoising convolutional neural network (DnCNN) is designed for image denoising and obtained effective results. In [64], deep residual network is introduced to learn a non-linear map between noisy and clean image for HSI denoising. In [65], GAN is used for estimating the noise distribution and constructing a paired training dataset to train CNN for image blind denoising. Compared with conventional denoising methods, deep learning-based methods are usually not limited to specific denoising tasks and the parameters are automatically updated according to the input. Therefore, deep learning methods are explored to remove noise and mine invariant features of targets to improve target detection results in our research.

## Thesis objective and contributions

This dissertation is devoted to using deep learning methods to process HSIs. The targets are denoised, classified and detected by analyzing hyperspectral data and fully extracting spectral-spatial characteristics with deep learning models. More specifically, the main contributions are summarized as follows:

- Considering that CNN has great potential in feature extraction and has been widely used in the field of image processing, 2D-CNN and 3D-CNN are introduced to extract the features among data for hyperspectral classification respectively. Since the CNN performance is greatly influenced by the parameter settings, a 2D-CNN with parameter tuning named 2D-CNN-PT is proposed to guarantee performance and further improve classification results.
- Limited label samples are the main challenge of applying deep learning models to HSIs. Since the optimization of GAN and CAE does not require the participation of label samples, unsupervised feature extractors based on GAN and CAE are

designed to help get rid of the limitation of labeled samples. The networks are built with fully 3D-convolutional layers to fully exploit the spectral-spatial information in HSIs. Besides, with the help of transfer learning and progressive growing training, a multi-size and multi-model framework is designed to increase the robustness of features to target size and improve the classification accuracy of small targets.

- Considering the great feature extraction and representation ability of deep learning models, DAE is introduced to remove noise and exploit the invariant information for target detection. Besides, a multiscale model is developed to incorporate complementary information, which provides more robust features for subsequent spectral identification.

## Thesis outline

This thesis is organized in five chapters.

**Chapter 1** presents a supervised feature extraction method based on 2D-CNN for hyperspectral classification. Considering that the network performance is strongly influenced by the parameter settings, a 2D-CNN-PT framework is designed. Nine main parameters are tuned in turn according to the unique variable principle. The experimental results on two real-world HSIs show that the appropriate parameter settings are of great help to improve network performance and classification accuracy.

**Chapter 2** describes a supervised feature extraction method based on 3D-CNN for hyperspectral classification. 3D-CNN can flexibly process multi-dimensional data, and mine the spectral-spatial information at the same time. Due to the high dimensionality of HSIs, a large amount of calculation is caused. Therefore, a novel band selection method is proposed to quickly select the band and reduce the dimensionality of HSIs. In addition, to solve the problem of insufficient labeled samples and improve the classification of HSIs, an improved 3D-CNN classification method based on transfer learning and virtual samples is proposed. Experimental results show that either transfer learning or virtual samples can help us further improve the classification accuracy, and the 3D-CNN combining transfer learning and virtual samples yields the best classification results.

**Chapter 3** introduces an unsupervised feature extraction method based on GAN. At first, some basic knowledge of GAN is overviewed. Next, a novel dimensionality reduction method based on  $1 \times 1$  convolution and  $1 \times 1$  pooling is proposed to obtain lower dimensionality data containing more abstract and high-level features.

Then, an unsupervised feature extraction framework for hyperspectral classification based on 3D-WGAN-GP and transfer learning is designed to get the rid of the limitation of labeled samples. Finally, experiments are performed on real-world data sets to verify the performance of the proposed method, and experimental results prove the feasibility and effectiveness of the proposed method.

**Chapter 4** introduces the other two unsupervised feature extraction frameworks based on 3D-CAE. In the first framework, multi-level features are proposed to contain detail information and semantic information at the same time. The proposed multi-level features are directly obtained from different encoded layers of the optimized encoder, which helps us to make full use of the well-trained network and further improve feature quality. In the second framework, a 3D-M<sup>2</sup>CAE consisting of three 3D-CAEs with different input sizes is proposed to balance different targets and improve classification results of small targets. Benefiting from the progressive training methodology and transfer learning, the optimization of 3D-M<sup>2</sup>CAE is facilitated and accelerated. Experimental results show that the designed two frameworks have great promise in unsupervised feature extraction.

**Chapter 5** proposes a MSDAE for improvement of target detection. In order to remove noise and retain invariant features, DAE is introduced to reconstruct spectrums and exploit the invariant information for target detection. Besides, to fully extract the features from the original spectrums, the MSDAE model is designed to incorporate complementary information. The final spectrum is fused by reconstructed spectrums from different scales representations, which provides more complex information and more robust features for subsequent spectral identification. The results on simulated and real-world data demonstrate that the proposed method can not only improve the target detection but also has great potential for preserving small targets. In addition, unsupervised segmentation is investigated to help small target detection.

# Chapter 1

## Supervised feature extraction based on 2D-CNN for hyperspectral classification

### 1.1 Introduction

As the main branch of machine learning, deep learning has shown considerable potential in the field of remote sensing classification [66, 67] owing to its strong capability for big data analysis, which enables it to extract the inherent laws and characteristics of the data [68]. CNN has a unique network structure with local connection and weight sharing, which reduces the number of parameters significantly [69] and it have been successfully used for HSI classification [70–72]. However, the CNN performance is greatly affected by the parameter settings and the loss function may reach a local minimum owing to inappropriate weights. To improve the CNN performance for HSIs, a classification method based on a CNN with parameter tuning is proposed in this Chapter.

### 1.2 Overview of 2D-CNN

A 2D-CNN mainly includes convolutional layers and pooling layers as shown in Figure 1.1, where  $C_i$ ,  $P_i$  and  $F$  represent the  $i$ th convolutional layer, the  $i$ th pooling layer, and fully connected layer respectively, and  $n @$  feature map means there are  $n$  feature maps in current layer. Each convolution kernel corresponds to an output (feature map), and different convolution kernels can extract different features.



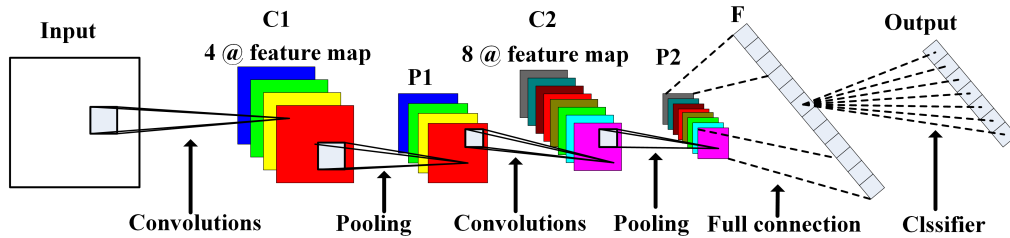


Figure 1.1: A conventional CNN structure.

Convolution operations have been widely used in signal processing and image processing. They can apply convolution kernels to an input data to produce feature maps and show great potential in feature extraction. Figure 1.2 shows a diagram of 2D convolution operation [73].

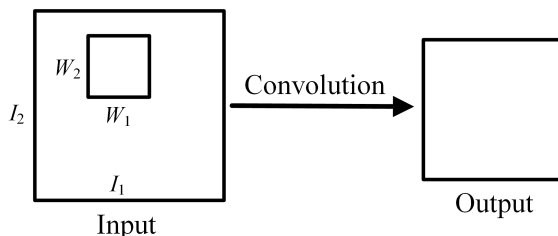


Figure 1.2: 2D convolution operation.

It can be seen from Figure 1.2 that 2D convolution is performed on 2D input data. When the convolution kernel slides over the input, compute the product of the mutually overlapping pixels and calculate their sum, then a 2D output is obtained. When 2D convolution operation is done with the stride being  $1 \times 1$ , its output  $O^{x, y}$  at position  $(x, y)$  is defined as:

$$O^{x, y} = \sum_{p=0}^{W_1-1} \sum_{q=0}^{W_2-1} W^{p, q} I^{x+p, y+q} + b \quad (1.1)$$

where  $\mathbf{I} \in \mathbb{R}^{I_1 \times I_2}$  represents the input with dimension of  $I_1 \times I_2$ ,  $\mathbf{W} \in \mathbb{R}^{W_1 \times W_2}$  is the convolution kernel, and  $b$  is the bias.

In the pooling layers, data can be subsampled by reducing the resolution of the feature maps while the number of feature maps is unchanged. Figure 1.3 shows examples of max-pooling and mean-pooling.

Max-pooling operation calculates the maximum value for patches, and mean-pooling (also called average-pooling) operation calculates the average value for patches. Both of them can be used to progressively reduce the spatial size of the representation, and reduce the amount of parameters and computation in the network.

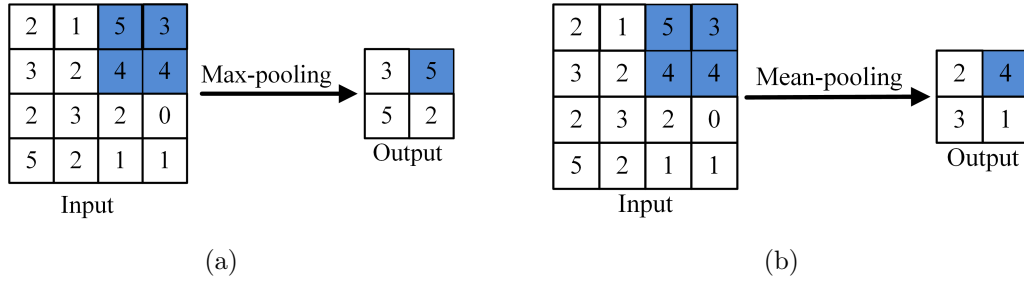


Figure 1.3: 2D pooling operation: (a) Max-pooling, (b) Mean-pooling.

### 1.3 Hyperspectral classification based on 2D-CNN

To explore the rich information in HSIs, a 2D-CNN is introduced for feature extraction and classification in this subsection. Although CNN reduces network parameters through local connection and weight sharing, its overall performance is influenced by the network parameters, such as the input size, network structure, pooling method, activation function and so on. In the existing literature, the network parameters are generally set by default. Hence, appropriate parameter selection methods are worth studying in order to improve the results.

#### 1.3.1 Optimal parameter selection based on 2D-CNN

To obtain the optimal CNN parameters for HSI classification, a classification method based on a 2D-CNN with parameter tuning (2D-CNN-PT) is proposed.

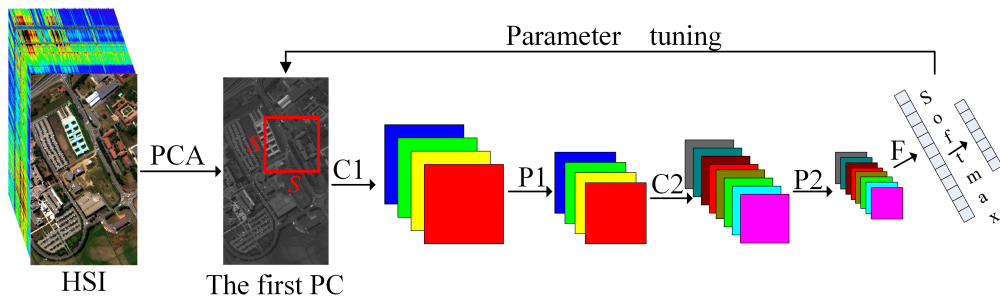


Figure 1.4: Hyperspectral classification based on a 2D-CNN with parameter tuning.

The procedure of the proposed 2D-CNN-PT method shown in Figure 1.4 can be mainly divided into four steps:

Firstly, considering that HSIs are high-dimensional data composed of numerous spectral bands and they include a significant amount of redundancy. PCA can represent the original data with a set of linearly uncorrelated variables through orthogonal

transformation. Therefore, PCA is used to reduce the number of dimensions [74,75]. It's known that HSIs are 3D data and each pixel corresponds to a spectral vector. Therefore, we unfold the 3D data into a 2D matrix. The number of rows in the 2D matrix is equal to the number of pixels in the HSI and the number of columns is equal to the number of spectral bands, i.e., each row corresponds to the spectral characteristics of the sample. This set of possibly correlated spectral characteristic variables into a set of values of linearly uncorrelated variables called principal components. The first principal component containing most of the information is preserved and reshaped into a 2D image with the same length and width as the original HSI.

Next, for each observed pixel, one block of  $S \times S$  pixels on the 2D image obtained based on The first principal component is selected as an input data. A CNN consisting of two pooling layers, two convolutional layers and one fully connected layer is constructed with input size being  $S \times S$  and classifier being softmax regression. Softmax function can converts a vector of numbers into a vector of probabilities through Eq. (1.2):

$$Out_{y_m} = \text{softmax}(y_m) = \frac{e^{y_m}}{\sum_{j=1}^N e^{y_j}} \quad (1.2)$$

where  $m = 1, 2, \dots, N$ , and  $N$  is the number of classes.  $y_m$  is the value of the  $m$ th class.  $Out_{y_m}$  which takes values between 0 and 1 is the corresponding output value after softmax.

Then, the network parameters are tuned in turn according to the unique variable principle based on the classification results during classification, which means that only one parameter is changed while the others are fixed to explore the effect of the changed parameter on the experimental results. A set of appropriate parameters can be obtained according to parameter tuning.

Finally, the 2D-CNN with optimal parameters can be used to extract features for hyperspectral classification.

## 1.3.2 Data set description and assessment criteria

### 1.3.2.1 Data set description

There are some commonly used hyperspectral datasets used to test the performance of methods and models. Different data sets may have different characteristics in resolution, land-cover classes, image quality and so on. Pavia University dataset

and Indian Pines data set are used as our target data in the following simulations.

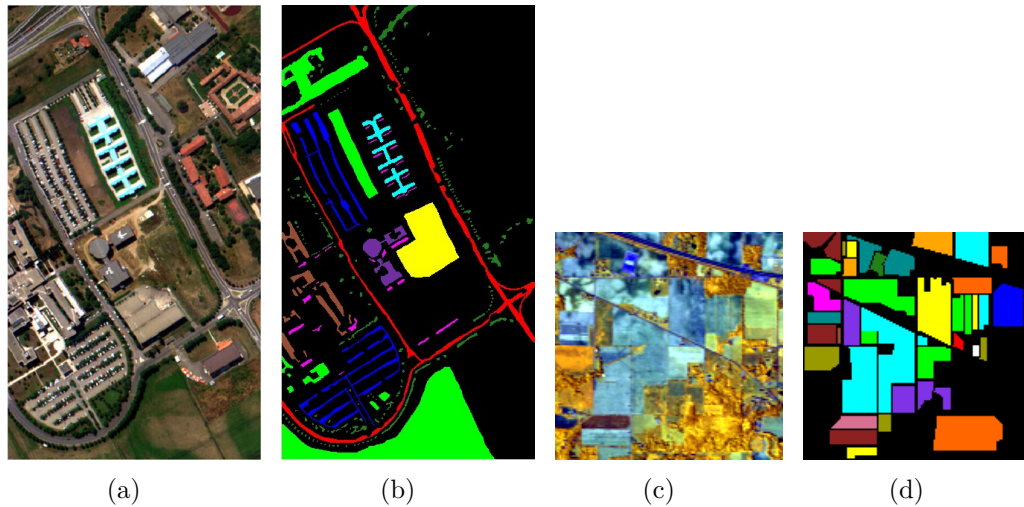











Figure 1.5: Data sets: (a) False-color image of Pavia University. (b) Ground truth of Pavia University. (c) False-color image of Indian Pines. (d) Ground truth of Indian Pines.

Pavia University data set, shown in Figure 1.5 (a), is collected by Reflective Optics System Imaging Spectrometer (ROSIS) sensor covering the University of Pavia, northern Italy. It contains  $610 \times 340$  pixels, and the spatial resolution is 1.3 meters per pixel. There are 103 useful spectral bands reserved ranging from 0.43 to  $0.86 \mu\text{m}$  after removing noise-affected bands. It can be seen from Figure 1.5 (b) that each class is color coded and pixels of the same color indicate that they are from the same class, where black represents the unlabeled area. There are 9 different land-cover classes in Pavia University data set. The details of land-cover classes and number of samples are listed in Table 1.1.
















Table 1.1: Details of land-cover classes in Pavia University data set.

Class No.	Color coding	Class name	Number of samples
1		Asphalt	6631
2		Meadows	18649
3		Gravel	2099
4		Trees	3064
5		Metal sheets	1345
6		Bare soil	5029
7		Bitumen	1330
8		Bricks	3682
9		Shadows	947

The Indian Pines data set, shown in Figure 1.5 (c), is acquired by the Airborne

Visible/ Infrared Imaging Spectrometer (AVIRIS) sensor over the Indian Pines test site in northwest Indiana, USA. There are 224 spectral bands ranging from 0.4 to 2.5  $\mu\text{m}$ . The number of spectral bands is reduced to 200 after removing bands covering the region of water absorption. The Indian Pines scene contains two-thirds agriculture, and one-third forest or other natural perennial vegetation. As shown in Figure 1.5 (d), there are 16 land-cover classes and black represents the unlabeled area. The details of land-cover classes and number of samples are listed in Table 1.2.

Table 1.2: Details of land-cover classes in Indian Pines data set.

Class No.	Color coding	Class name	Number of samples
1		Alfalfa	46
2		Corn-notill	1428
3		Corn-min	830
4		Corn	237
5		Grass-pasture	483
6		Grass-trees	730
7		Grass-pasture-mowed	28
8		Hay-windrowed	478
9		Oats	20
10		Soybean-notill	972
11		Soybean-mintill	2455
12		Soybean-clean	593
13		Wheat	205
14		Woods	1265
15		Buildings-grass-trees	386
16		Stone-stel-towers	93

### 1.3.2.2 Assessment criteria

In order to evaluate and compare the performance of different methods, overall accuracy (OA), average accuracy (AA), and kappa coefficient ( $\kappa$ ) are introduced to represent the classification results. All the values used in the experiments are average values obtained from multiple experiments.

If there are  $N$  classes in a data set and the number of samples in the  $n$ th class is  $\lambda_n$ . Thus, the total number of samples is  $\lambda$  ( $\lambda = \sum_{n=1}^N \lambda_n$ ).  $C_{nn}$  denotes the number of test samples that actually belong to the  $n$ th class, and are also classified into  $n$ th class. The OA, AA, and  $\kappa$  values can be defined as [76]:

$$\text{OA} = \frac{\sum_{n=1}^N C_{nn}}{\sum_{n=1}^N \lambda_n} \times 100\% \quad (1.3)$$

$$AA = \frac{1}{N} \sum_{n=1}^N \frac{C_{nn}}{\lambda_n} \times 100\% \quad (1.4)$$

$$\kappa = \frac{\frac{\sum_{n=1}^N C_{nn}}{\lambda} - \frac{\sum_{n=1}^N \lambda_n C_{nn}}{\lambda^2}}{1 - \frac{\sum_{n=1}^N \lambda_n C_{nn}}{\lambda^2}} \times 100\% \quad (1.5)$$

### 1.3.3 Experimental results

In the experiment, two seven-layer 2D-CNNs are built for Pavia University and Indian Pines, respectively. The initial network structure is listed in Table 1.3, where  $k_1 \times k_2 \times n$  in convolutional layer represents that there are  $n$  kernels with size of  $k_1 \times k_2$  and ReLU means rectified linear unit. The kernel size in pooling layer is  $2 \times 2$  and the stride is set to 2. The batch size and the number of units in output layer are 256 and 9 for Pavia University, and 128 and 16 for Indian Pines, respectively. Besides, the network weights are randomly initialized by a normal distribution with a mean and standard deviation of 0 and 0.5, the network weights are updated by the Adadelta algorithm with a learning rate of 1, and the number of epochs is set to 100 for the two 2D-CNNs.

Table 1.3: Initial network structure of 2D-CNN.

Layer	C1	P1	C2	P2	F
Parameter	$4 \times 4 \times 16$	$2 \times 2$	$3 \times 3 \times 32$	$2 \times 2$	128
Activation function	ReLU	Max-pooling	ReLU	Max-pooling	–

The following parameters are considered in this research: input size, network structure, number of units in the fully connected layer, activation function, pooling method, optimization method, batch size, number of convolutional kernels and number of epochs. Next, we will explore the influence of these parameters on the results.

#### 1.3.3.1 Input size

Considering the input size, i.e.,  $S \times S$ , has a great influence on the design of the whole network structure, it is optimized first in the experiment. Moreover, the proportion of the training data also has an impact on the classification results. Therefore, the training data ratio which reflects the proportion between the number of training samples and total samples and input size are considered simultaneously.

Odd input size ( $S \times S$ ) from  $13 \times 13$  to  $31 \times 31$  under the training ratio from 0.1 to 0.5 are tested in the experiment, while the other parameters are fixed. The experimental results are shown in Figure 1.6.

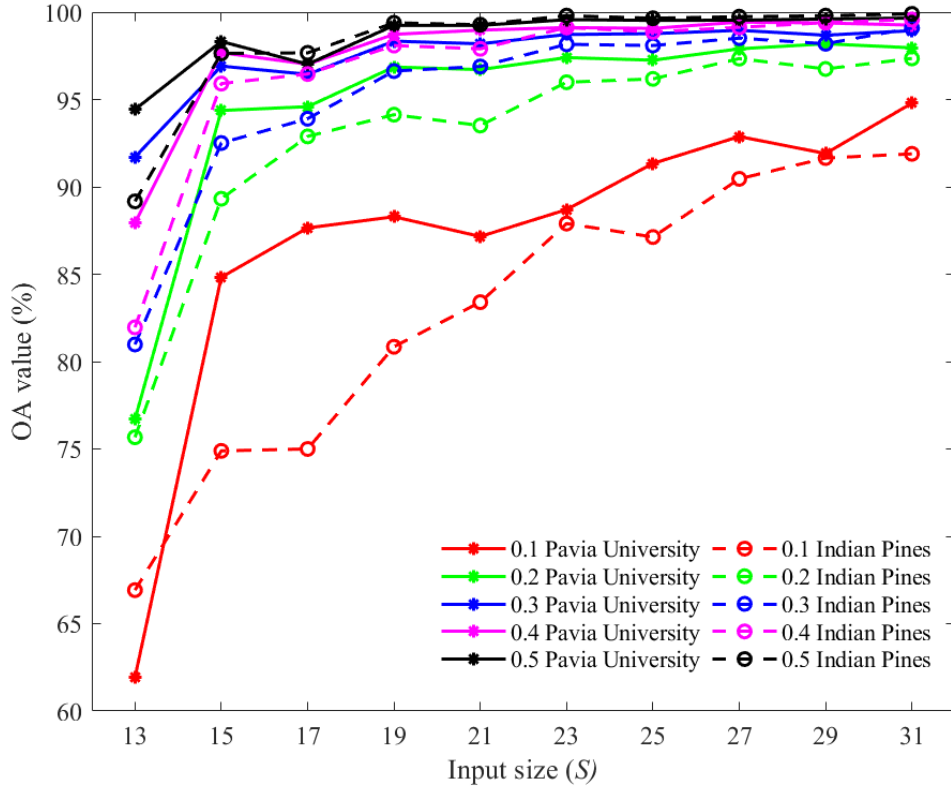


Figure 1.6: OA values under different input sizes and training set ratio.

It can be seen that when the training data ratio exceeds 0.1, the OA values of the two HSIs show an upward trend with input size at the beginning and then tend to stable under the same training set ratio. The larger the input size, the longer the training time. Therefore, when the training data ratio exceeds 0.1, it is more appropriate to set the input size to  $15 \times 15$ . When the training data ratio exceeds 0.1, the OA values show upward trend with the increase of input size. Due to the collection of the labeled samples are time consuming and labor consuming, the training data ratio of two HSIs is selected as 0.1, and  $27 \times 27$  is finally chosen as the input size because the corresponding OA value is higher and this size has been widely used in the literature which makes the experimental results more contrasting [21]. For the subsequent simulations, the input size is fixed at  $27 \times 27$  for both two data sets.

### 1.3.3.2 Network structure

The network structure is mainly affected by input size and the number of layers, and convolution kernel size, etc. Under the condition of the input size is  $27 \times 27$  and the number of units in fully connected layer is 128, the layer number and the size of the convolutional kernel are considered together in the experiment. Five 2D-CNNs with different structures are established as shown in Table 1.4.

Table 1.4: Network parameters of different networks.

Net	C1	P1	C2	P2	C3	P3
Net 1	$4 \times 4 \times 16$	$2 \times 2$	$3 \times 3 \times 32$	$2 \times 2$	—	—
Net 2	$4 \times 4 \times 16$	$2 \times 2$	$5 \times 5 \times 32$	$2 \times 2$	—	—
Net 3	$4 \times 4 \times 16$	$2 \times 2$	$3 \times 3 \times 32$	$2 \times 2$	$4 \times 4 \times 48$	—
Net 4	$4 \times 4 \times 16$	$2 \times 2$	$5 \times 5 \times 32$	$2 \times 2$	$4 \times 4 \times 48$	—
Net 5	$4 \times 4 \times 16$	$2 \times 2$	$3 \times 3 \times 32$	$2 \times 2$	$4 \times 4 \times 48$	$2 \times 2$

The OA values of the two data sets under different network structures are listed in Table 1.5. According to the experimental results in Table 1.5, the highest OA value of Pavia University is obtained under Net 3. For Indian Pines data set, Net 1 and Net 3 show better classification performance. However, the dimension of the feature vector before the fully connected layer of Net 3 is much smaller than that of Net 1 and low-dimensional features help reduce the amount of calculation and storage space. Therefore, Net 3 could be a good candidate for the Pavia University and Indian Pines.

Table 1.5: OA values of Pavia University and Indian Pines under different networks.

OA (%) \ Net	Net				
	Net1	Net 2	<b>Net 3</b>	Net 4	Net 5
Data set					
Pavia University	92.88	92.86	<b>93.45</b>	93.39	93.27
Indian Pines	90.46	88.96	<b>89.76</b>	89.65	89.48

### 1.3.3.3 Number of units in the fully connected layer

For the classification model, the fully connected layer plays the role of converting features into 1D vector form. The larger the number of units in the fully connected layer, the larger the feature dimension and the greater number of weights that needs



to be trained. The OA values under different number of units in fully connected layer are shown in Table 1.6, where  $n_f$  represents the number of units.

Table 1.6: OA values of Pavia University and Indian Pines under different number of units in fully connected layer.

OA (%) \ $n_f$	32	64	<b>128</b>	256	512	1024	2048
Data set							
Pavia University	92.74	93.27	<b>93.45</b>	93.49	93.50	93.95	93.16
Indian Pines	88.78	89.39	<b>89.76</b>	89.72	89.55	90.55	90.82

We can find from Table 1.6 that the OA value does not always increase as the number of units increases. A high number of units only increases the number of parameters, but may also introduce interference information. From the perspective of computational efficiency and accuracy, the optimal number of units in the fully connected layer for both the Pavia University and the Indian Pines is set to 128 which is more commonly used.

### 1.3.3.4 Activation function and pooling method

A neural network without an activation function becomes a linear system. Hence, the activation function allows the inclusion of nonlinear factors. An activation function is differentiable nearly everywhere [77]. Several commonly used activation functions ( $f(\cdot)$ ) are shown in Figure 1.7. A real number can be mapped to  $(0, 1)$  through the sigmoid function. However, the vanishing gradient problem occurs during back propagation. The hyperbolic tangent (tanh) function is suitable for various obvious features. But, it also suffers from the vanishing gradient problem. ReLU has been widely used in CNNs owing to its efficient computation [78]. Furthermore, it does not suffer from the vanishing gradient or exploding gradient problem. Through experiments we found that the OA values of the Pavia University and the Indian Pines are higher when ReLU is selected as activation function. Therefore, ReLU is a good choice for activation function.

The pooling layer is usually employed between the convolutional layers. Pooling operation can reduce the resolution of feature maps and prevent over-fitting to a certain degree. In CNNs, the commonly used pooling methods are max-pooling and mean-pooling [79]. Max-pooling effectively retains the texture information of images, whereas mean-pooling effectively preserves the background information of

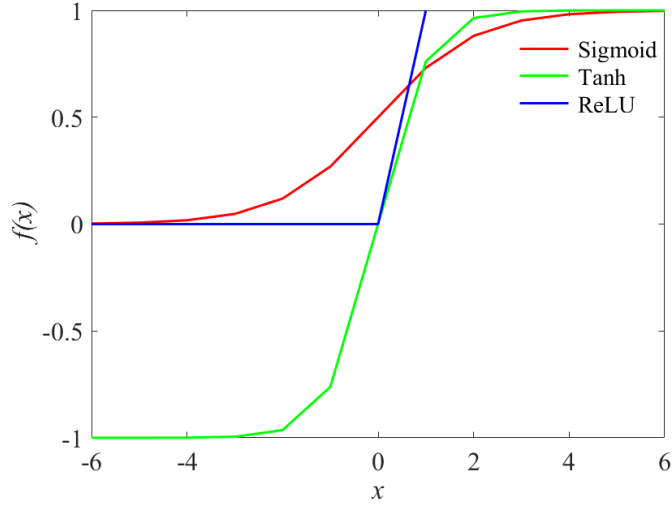


Figure 1.7: Different activation functions.

images. In theory, texture information is more useful for image classification. In the experiments, the OA values are also higher for both HSIs under max-pooling. Therefore, max-pooling is selected as the pooling method for the Pavia University and Indian Pines.

### 1.3.3.5 Optimization method

Some optimization methods [80,81]: Adadelta, stochastic gradient descent (SGD), adaptive gradient algorithm (Adagrad), root mean square prop (RMSprop), adaptive moment estimation (Adam) are compared in this part.

Table 1.7: OA values of Pavia University and Indian Pines under different optimizers.

Data set	OA (%)	Optimizer				
		SGD	Adagrad	Adadelta	RMSprop	<b>Adam</b>
Pavia University		92.87	93.03	93.45	93.95	<b>94.14</b>
Indian Pines		86.58	87.38	90.45	90.53	<b>90.61</b>

It can be seen from Table 1.7 that the highest OA value is obtained when Adam is used as the optimizer. Therefore, Adam is chosen as the optimizer for the two data sets.

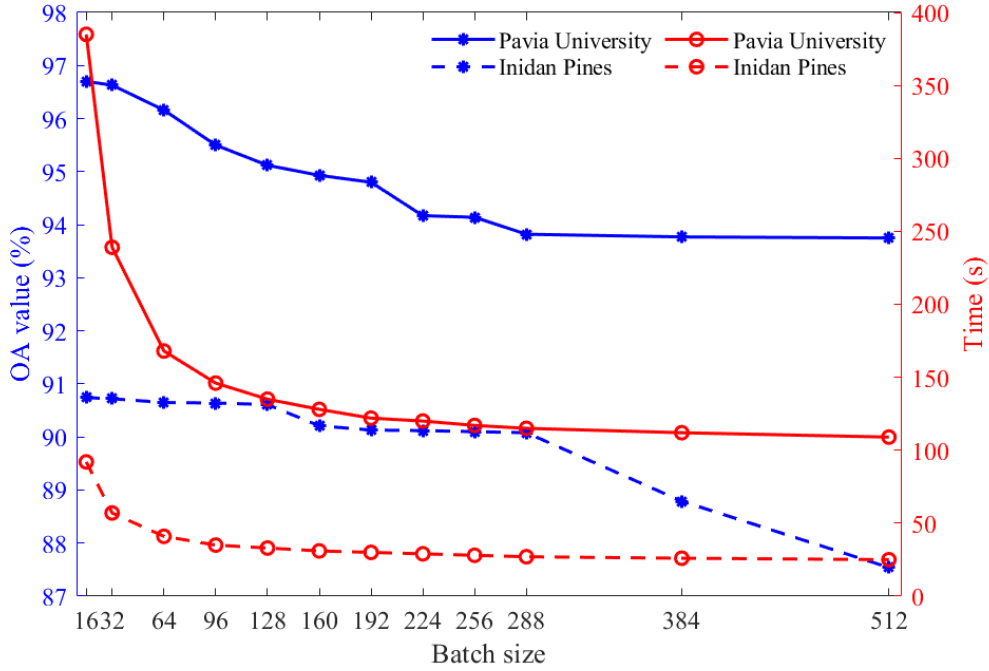


Figure 1.8: OA values and computation times under different batch sizes.

### 1.3.3.6 Batch size

In the following experiments, a mini-batch based on the Adam algorithm is used. The relationship between the OA values and the computation time under different batch sizes is shown in Figure 1.8. It can be seen that when the batch size increases, the OA values of the two HSIs decrease. But the computation is more efficient, especially for the Pavia University. To achieve a tradeoff between the OA values and the computation time, 128 is chosen as the batch size for the Pavia University and Indian Pines.

### 1.3.3.7 Number of convolutional kernels

Feature extraction is a key indicator of classification performance. Different convolutional kernels can extract different features. However, the computational complexity increases with the number of convolutional kernels. The OA values under different numbers of convolutional kernels are shown in Figure 1.9 and only the number of kernels in the first convolution layer is listed, where if the number of kernels in the first convolution layer is  $n$ , the number of kernels in the  $l$ th convolution layer is  $n \times l$ .

As seen from the Figure 1.9, for Pavia University, the OA values are lower when the number of kernels in the first convolution layer is 8 and 96, while for other kernel

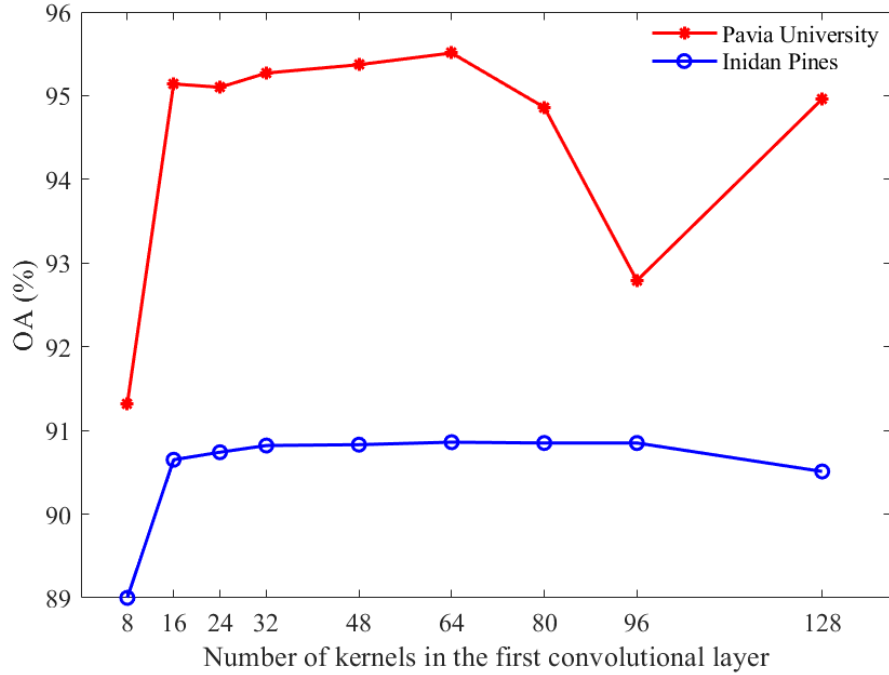


Figure 1.9: OA values under different number of convolutional kernels.

numbers, the accuracy does not change much. For the Indian Pines, the OA values do not vary significantly with the number of kernels increasing (except when it is 8). In the case of the same other parameters, the amount of calculation increases and the training time becomes longer as the kernel number increases. Therefore, considering the aforementioned factors, the number of kernels in the first convolutional layer is selected as 16 for Pavia University and Indian Pines.

### 1.3.3.8 Number of epochs

All inputs are processed one time individually of forward and backward to the network, called one epoch. The number of epochs has a significant impact on the computation time. The more the number, the longer is the time required to train the network. Therefore, the epoch number is tuned after the other parameters have been optimized. Too many epochs will not only reduce the efficiency but may also cause over-fitting. The OA values under different number of epochs are shown in Table 1.8.

It can be found from Table 1.8 that OA values slowly increase as the number of epochs increases, but the training time increases greatly. Considering OA values and training time, the number of epochs is set to 100 for both two data sets.

A set of optimal parameters can be finally obtained after parameter tuning.

Table 1.8: OA values of Pavia University and Indian Pines under different number of epochs.

OA (%) \ $n_e$	100	200	300	400	500	600
Data set						
Pavia University	<b>95.03</b>	95.10	95.14	95.14	95.36	95.37
Indian Pines	<b>90.88</b>	90.78	90.84	90.83	90.85	90.87

However, overfitting is still a problem need to be faced. To prevent complex co-adaptations on the training data, dropout can be used to reduce overfitting by randomly omitting some hidden units from the network [82]. Therefore, dropout is introduced into the optimized 2D-CNN to further improve the classification performance of the network.

### 1.3.3.9 Comparison of classification results

For better visual observation of the effectiveness of the proposed method, feature extraction methods based on factor analysis (FA) and DBN are considered for comparison to better evaluate the performance of the proposed method. FA is a linear statistical method that uses fewer numbers of factors to replace original data [83]. DBN is composed of multiple layers of latent variables and it usually takes a 1D vector as input, which learns deep features via pretraining in a hierarchal manner [84-86].

The classification maps of Pavia University and Indian Pines obtained by different methods are shown in Figure 1.10 and Figure 1.11, respectively.

It can be seen from Figure 1.10 that classification map of FA has the most misclassified pixels, especially a large number of pixels in the green are misclassified as pixels in the yellow area in the upper left and lower left of the HSI. Besides many pixels in the yellow are mistakenly classified as green in the central area of the HSI. In Figure 1.10 (c), the number of misclassified pixels in the upper left and lower left is greatly reduced compared with Figure 1.10 (b), but there are still lots of pixels in the central area. Overall, the classification map of Figure 1.10 (d) is the clearest and the misclassified pixels are the least.

For Indian Pines, we can find that the classification maps in Figure 1.11 (b) and (c) have many misclassified pixels in the upper left area. Although there are still some misclassified pixels in Figure 1.11 (d), the classification map obtained by

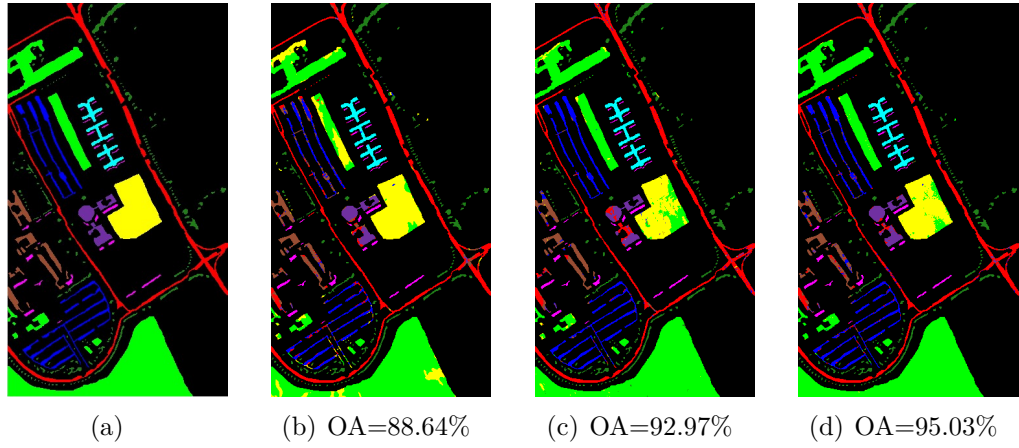


Figure 1.10: Classification maps of Pavia University under different methods: (a) Ground truth, (b) FA, (c) DBN, (d) 2D-CNN-PT.

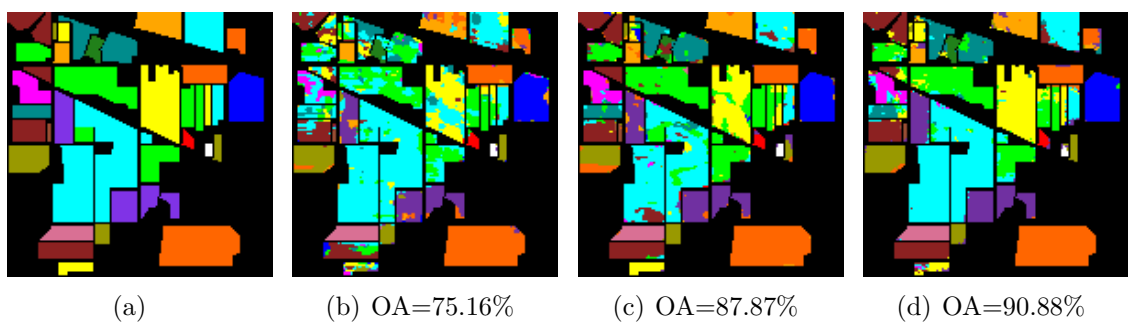


Figure 1.11: Classification maps of Indian Pines under different methods: (a) Ground truth, (b) FA, (c) DBN, (d) 2D-CNN-PT.

2D-CNN-PT is the clearest.

## 1.4 Conclusion

A 2D-CNN-PT method is proposed in this Chapter to improve the classification results of HSIs. First, PCA is introduced to reduce the HSI dimension. Second, a 2D-CNN is constructed and the parameters of the 2D-CNN are tuned in turn according to the unique variable principle on the basis of experimental results and effectiveness. Finally, classification is performed with the optimized 2D-CNN. The experimental results on two real-world HSIs show that the proposed 2D-CNN-PT method achieves better HSI classification performance compared to the other two commonly used methods.

Although the optimal parameters are obtained under a limited set of explored parameters, these results can provide a reference for parameter initialization settings of other models, and it will save time for parameter tuning of models.

# Chapter 2

## Supervised feature extraction based on 3D-CNN for hyperspectral classification

### 2.1 Introduction

In the Chapter 1, 2D-CNN is introduced to mine information in HSIs. 2D-CNN can directly take 2D data as input and has great potential in preserving spatial structure of the target. However, HSIs that usually contain hundreds of spectral channels not only contain spatial information but also provide abundant spectral information. Therefore, 2D-CNN has limitations in retaining the spectral information. In [87], a two channels deep CNN composed of a 1D-CNN and a 2D-CNN is proposed to learn jointly spectral-spatial information, but the characteristics of the spectral domain and the spatial domain are extracted separately. Considering that the 3D data can be directly input into the 3D-CNNs, which helps to fully exploit the spectral and spatial information at the same time and avoids complex data reconstruction [36–40], 3D-CNN is developed to fully exploit the spectral-spatial information of HSIs in this chapter.

### 2.2 Hyperspectral classification based on 3D-CNN

2D-CNNs mainly capture features from the spatial domain, but 3D-CNNs could help to obtain spatial-spectral features of tensors [88]. Therefore, a 3D-CNN is considered to fully exploit the information among hyperspectral data. The flow



chart of a conventional 3D-CNN for hyperspectral classification is illustrated in Figure 2.1.

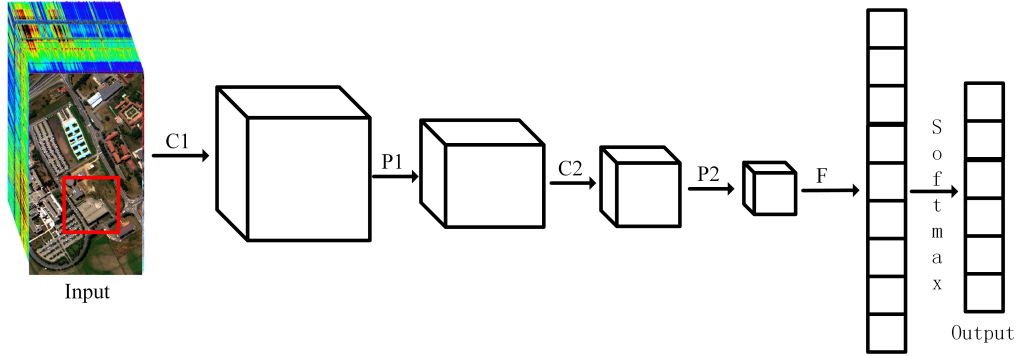


Figure 2.1: A conventional 3D-CNN for hyperspectral classification.

3D-CNN mainly obtains features through 3D convolution. It is known that 2D convolution operation can be performed on 2D input data and it has strong abilities in retaining the spatial information of the data. We can find from Figure 2.2 that 3D convolution can be performed on 3D data. The 3D filter in 3D convolution can move in three directions (width, height, and depth of data) and each movement of the filter can obtain a value by element-wise multiplication and addition. The output of 3D convolution is also a 3D data.

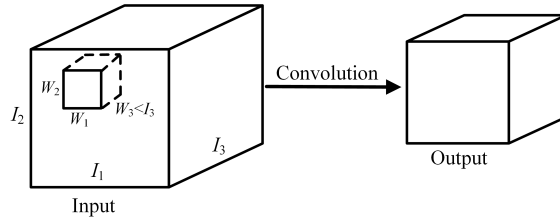


Figure 2.2: 3D convolution operation.

When 3D convolution operation is performed with stride of  $1 \times 1 \times 1$ , the output at position  $(x, y, z)$  can be calculated by:

$$O^{x, y, z} = \sum_{p=0}^{W_1-1} \sum_{q=0}^{W_2-1} \sum_{r=0}^{W_3-1} W^{p, q, r} I^{x+p, y+q, z+r} + b \quad (2.1)$$

where  $\mathbf{I} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$  represents the input with dimension of  $I_1 \times I_2 \times I_3$ ,  $\mathbf{W} \in \mathbb{R}^{W_1 \times W_2 \times W_3}$  is the convolution kernel, and  $b$  is the bias.

In order to preserve as much spatial and spectral information of the target as possible, a 3D data block centered on the target can be used as the input of 3D-CNN. However, HSIs usually contain hundreds of spectral bands and the information

of adjacent bands is highly correlated. If the input data contains all the bands, not only the amount of calculation is increased but curse of dimensionality may be caused [7]. Therefore, dimensionality reduction is an important step in hyperspectral classification based on 3D-CNN.

In subsection 1.3, PCA is used to reduce the dimensionality of HSIs. In addition to the commonly used PCA, band selection is one of the alternative ways to reduce the dimensionality of HSIs.

### **2.2.1 Band selection**

When band selection is implemented unsupervised, a commonly-used approach is to combine all possible subsets to find the most satisfactory objective value under some criterion, such as signal-to-noise ratio (SNR), optimum index factor (OIF), which may result in excessive computational complexity and cost. Moreover, the spectral signatures which are important to differentiate the materials are easy to be ignored or destroyed [89].

Selecting the bands with more invariant features and low correlation can help us improve the training efficiency and obtain better network performance. In order to efficiently select the band of HSIs, a fast band selection method based on a modified Douglas-Peucker algorithm named FMDP is proposed for hyperspectral classification. In the FMDP method, the number of invariant features calculated by scale invariant feature transformation (SIFT) algorithm and the spectral values are taken into account as evaluation criteria to ensure band information and spectral characteristics. The bands are simplified by limiting the distance between the selected adjacent bands, which not only reduces the band correlation but also reduces the number of iterations.

#### **2.2.1.1 Introduction of SIFT and DP algorithms**

The SIFT algorithm is proposed by David [90,91] which can be used to transform an image into local feature vectors. The features are invariant to image scaling, translation and rotation, which is of great help to image processing [92].

To find the distinctive features, a difference-of-Gaussian pyramid needs to be constructed at first to search for potential interest points that are brighter or darker than its surroundings. Next, the location and scale of candidate keypoints are determined by performing a detailed model fit to the nearby data and they are selected according to their stability. Then, each keypoint is assigned one or more

orientations based on its local image patch. Finally, the descriptor is built for each keypoint in its local neighborhood, which is measured by the local image gradients. The SIFT keypoints are useful for object recognition and matching due to their invariance and distinctiveness.

DP algorithm also known as the Ramer DP algorithm, is proposed by Urs Ramer [93], David Douglas and Thomas Peucker [94]. It is perfected by other scholars in the following decades, which has been widely used to compress the redundant graphical points and extract keypoints.

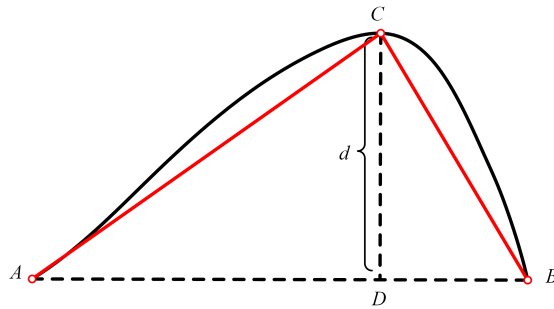


Figure 2.3: DP algorithm for line simplification.

DP algorithm produces straight-line segments to simplify the original curve, and the distance between the curve points and the segment is used as the evaluation criterion. Given a curve, connect the start point  $A$  and the end point  $B$  can get a straight-line segment  $AB$  shown in Figure 2.3. Search the farthest point  $C$  on the curve where the distance from the segment  $AB$  is the maximum. Comparing the maximum distance  $d$  with the preset threshold  $T$ , if  $d$  is greater than  $T$ ,  $C$  is selected to become the end point of the two new segments  $AC$  and  $BC$ . For each segment created, repeat the previous process until the maximum distance not satisfies the threshold value, and a polyline which can be used as the approximation of curve is eventually obtained.

### 2.2.1.2 Proposed FMDP method for band selection

The goal of selecting the band is to keep the bands containing more information and low correlation. Considering the distinctive features from scale-invariant keypoints obtained by SIFT are invariant to image scale and rotation, which is of immense benefit to image processing, the number of keypoints can be introduced to estimate the information of band. Since an HSI is a 3D data cube and each channel collects geometrical information of the same scene in the spatial domain [95],

keypoints detection can be individually performed on the 2D image corresponding to each band to estimate the number of keypoints. Besides, different dimensions of HSI correspond to different energies represented by the square of Frobenius norm (F-norm). The smaller the corresponding F-norm value of the image, the lower the corresponding energy and the less useful information contained. In order to better estimate the information contained in the band, an evaluation criteria ( $Q(h_3), h_3 = 1, 2, \dots, H_3$ ) of the  $h_3$ th band considering the aforementioned keypoint number and F-norm is defined as follows:

$$Q_{fn}(h_3) = \|H(:, :, h_3)\|_F^2 = \sum_{h_1=1}^{H_1} \sum_{h_2=1}^{H_2} |H(h_1, h_2, h_3)|^2 \quad (2.2)$$

$$Q(h_3) = Q_{kp}(h_3) \times Q_{fn}(h_3) \quad (2.3)$$

where the raw HSI with  $H_1$  rows,  $H_2$  columns and  $H_3$  spectral bands,  $Q_{fn}(h_3)$  represents the F-norm value of the  $h_3$ th band,  $Q_{kp}(h_3)$  represents the number of the keypoints of the image corresponding to the  $h_3$ th band. Both  $Q_{kp}(h_3)$  and  $Q_{fn}(h_3)$  are normalized to [0 1]. The larger the  $Q(h_3)$  value of a band, the greater the possibility that it has high quality.

In addition to the selected bands with high quality and abundant information, low correlation between the selected bands is also essential. Besides, each pixel in HSIs corresponds a spectral curve, which can be used to distinguish land-cover classes. Therefore, we expect the spectral characteristics can be reserved as much as possible and the corresponding polylines of selected bands are able to distinguish the target. To get rid of the computational complexity caused by combining all possible sub-bands to choose the least relevant one, we are looking for a new way to reduce the band correlation. It can be known that the neighboring bands are more correlative through calculating the correlation coefficient between the bands. Therefore, a modified DP algorithm is used to select bands and reduce the band correlation by controlling the distance between adjacent selected bands.

As shown in Figure 2.4, if curve  $AB$  is plotted according to  $Q(h_3)$  value calculated by the Eq. (2.3) where  $A$  and  $B$  correspond to the first and last bands of an HSI.  $C$  is the point with maximum  $Q(h_3)$  value on the curve, corresponding to the  $D$ th band. Instead of setting a threshold for  $d$  (the distance from point  $C$  to the straight-line segment  $AB$ ), two thresholds ( $T_1$  and  $T_2, T_1 < T_2$ ) are set to limit the distance ( $X_1$ ) between point  $A$  and point  $D$ , and the distance ( $X_2$ ) between point  $D$  and point  $B$ .  $X_1$  and  $X_2$  are estimated by the difference of the corresponding band number, for

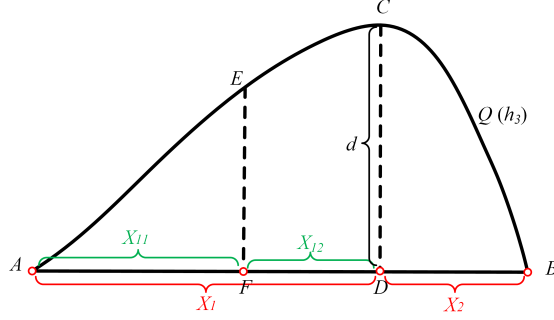


Figure 2.4: The modified DP algorithm.

example, the distance of the  $i$ th band and  $j$ th band is  $|j - i|$ . If  $T_2 < X_1$  and  $T_1 < X_2 < T_2$ , the  $D$ th band is added to the subset of selected bands. Continue to search for the possible bands to be selected between the band  $A$  and band  $D$ . If  $T_1 < X_{11} < T_2$  and  $T_1 < X_{12} < T_2$ , the  $F$ th band is added to the subset of selected bands. If the distance between all adjacent selected bands meets the condition, i.e.  $T_1 < X_1, X_2, X_{11}, X_{12} < T_2$ , the band selection ends. The  $A$ th,  $F$ th,  $D$ th,  $B$ th bands are the final selected bands. If not, repeat the previous step until all distances meet the condition.

The detailed procedure of the proposed FMDP is shown in Figure 2.5, where a stadium box indicates the beginning and ending of a process, a parallelogram box denotes the process of inputting and outputting data, a rectangular box represents a processing step, and a diamond is used to represent a decision point in the process.

For each HSI, the evaluation criteria  $Q_{kp}(h_3)$  can be calculated. Given two thresholds ( $T_1, T_2$ ) and the number of bands to select  $N$ , we can get a set of candidate bands based on the modified DP algorithm. Sort all the candidate bands according to  $Q(h_3)$  value and the top  $N$  bands with the largest  $Q(h_3)$  value are the final selection. Besides, to reduce the number of iterations and increase the difference of the selected bands, the set of the two thresholds try to make the number of candidate bands is near to  $N$  toward 0 and  $T_2$  tries to be about twice as large as  $T_1$ . When the proposed FMDP is utilized, only a few iterations are needed and the spectral information can be preserved as much as possible, which not only greatly improves the computational efficiency but also lays a solid foundation for subsequent classification.

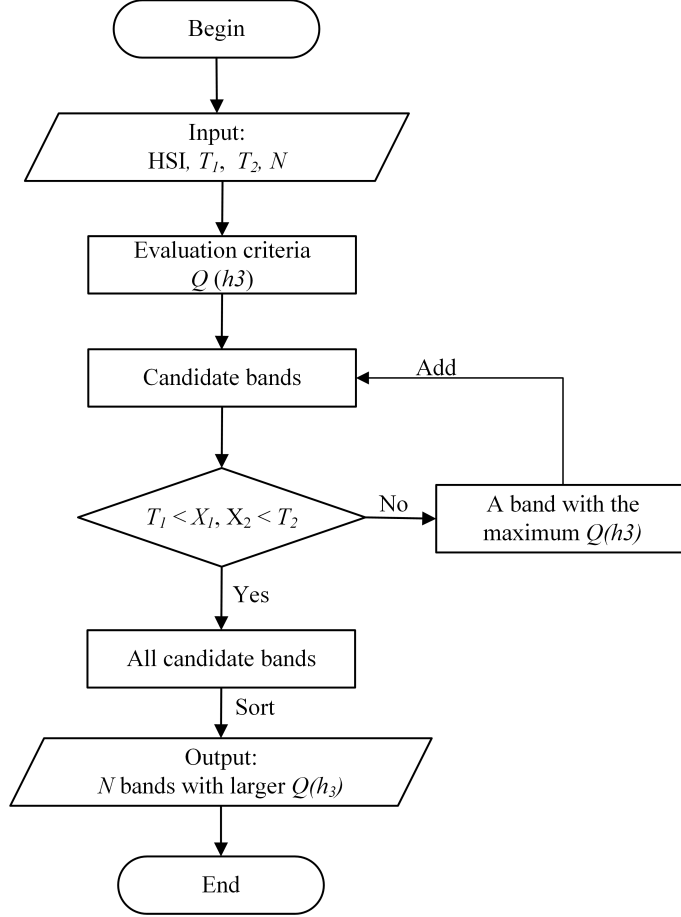


Figure 2.5: The flowchart of proposed FMDP for band selection.

### 2.2.1.3 Experimental results of band selection

To verify the effectiveness of the proposed band selection method, comparative experiments on Pavia University and Indian Pines data sets are carried out in this subsection.

When the band to be selected is 10, for Pavia University,  $T_1$  and  $T_2$  are set to 9 and 18; For Indian Pines,  $T_1$  and  $T_2$  are set to 16 and 32. The original spectral curves and fitted spectral curves of different classes in Pavia University and Indian Pines are depicted in Figure 2.6 and 2.7, respectively. The selected bands are corresponding to the red asterisk in Figure 2.6 (b) and 2.7 (b).

It can be observed from Figure 2.6 and 2.7 that the selected bands try to fill the band range instead of focusing on a narrow spectral band, which helps reduce data redundancy and data correlation. In addition, most of the selected bands are key locations that affect the contour of the spectral curve and the fitted spectral curves are closer to the original curves, which allows the selected band can provide more

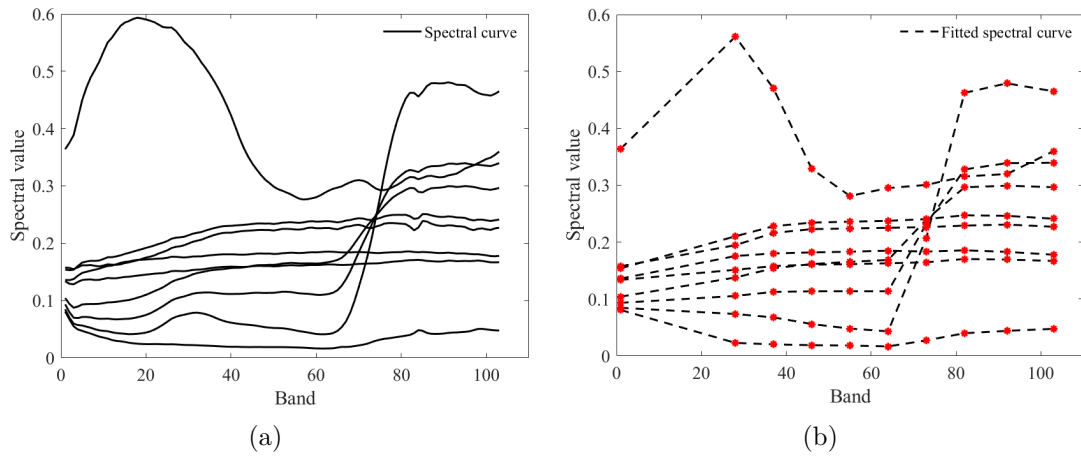


Figure 2.6: Pavia University: (a) Original spectral curve, (b) Fitted spectral curve with selected bands.

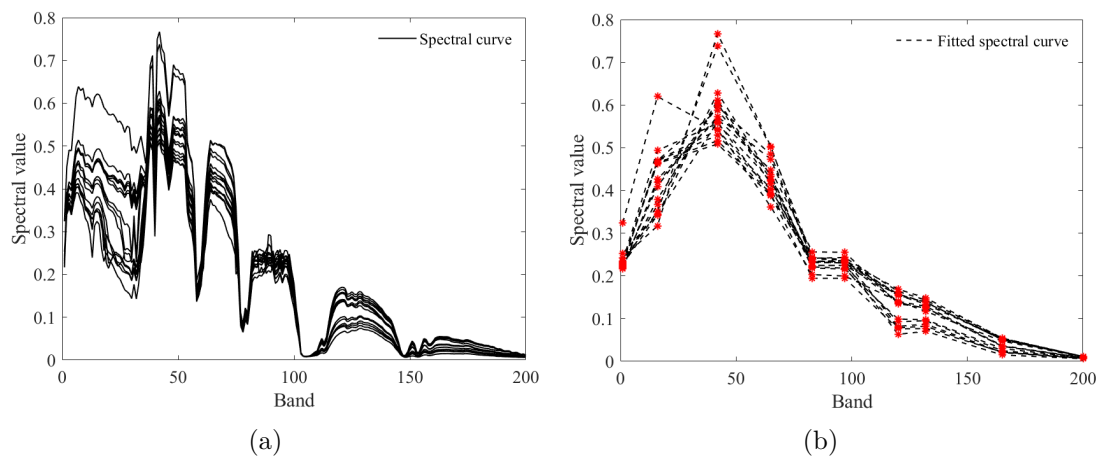


Figure 2.7: Indian Pines: (a) Original spectral curve, (b) Fitted spectral curve with selected bands.

spectral information to distinguish the materials.

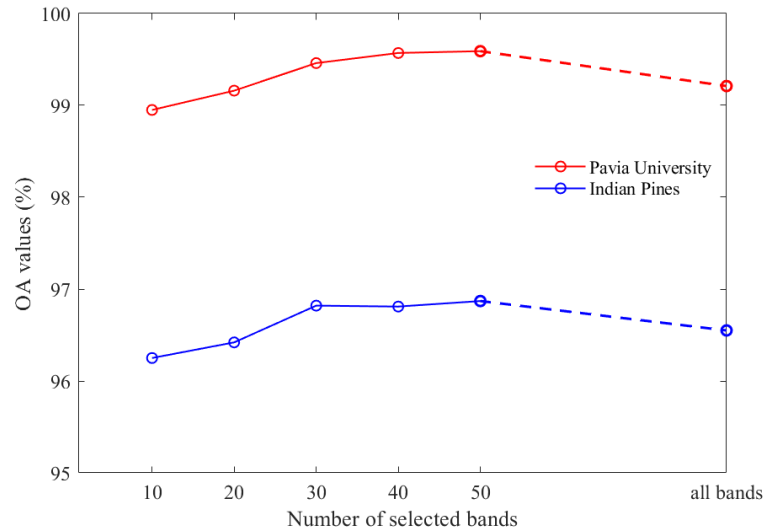


Figure 2.8: OA values under different number of selected bands.

When the number of selected band ranges from 10 to 50, the OA values based on 3D-CNN of Pavia University and Indian Pines are depicted in Figure 2.8. It can be found that OA values increase as the number of selected bands increases and then stabilize. However, according to the OA value obtained by using all the bands, it can be inferred that the OA values exceed a certain range decrease as the band increases. Overall, even in the rising range, the change in OA value is not great, but the amount of data calculation has greatly increased. Therefore, dimensionality reduction is very helpful to improve efficiency.

In order to better verify the performance of the proposed method, four other unsupervised band selection methods, maximum-variance principal component analysis (MVPCA), adaptive band selection (ABS), minimum noise band selection method (MNBS), and band column selection (BCS) are considered for comparison. MVPCA [96] prioritizes bands using loading-factors matrix via the corresponding eigenvalues and eigenvectors to achieve band reduction. ABS is developed in [97] based on OIF [98] and the high correlation between adjacent bands is also take into account. In [99], a MNBS is proposed based on data quality integrating both SNRs and correlation of bands. Because each HSI can be represented as a tensor data and selecting the most desirable column subsets is an analogy to band selection for HSIs, BCS is employed in [100] by transforming the original hyperspectral data into a matrix and then selecting the most informative and least correlative column subset. The OA values with different number of selected bands of Pavia University and Indian Pines are shown in Figure 2.9 and Figure 2.10, respectively.



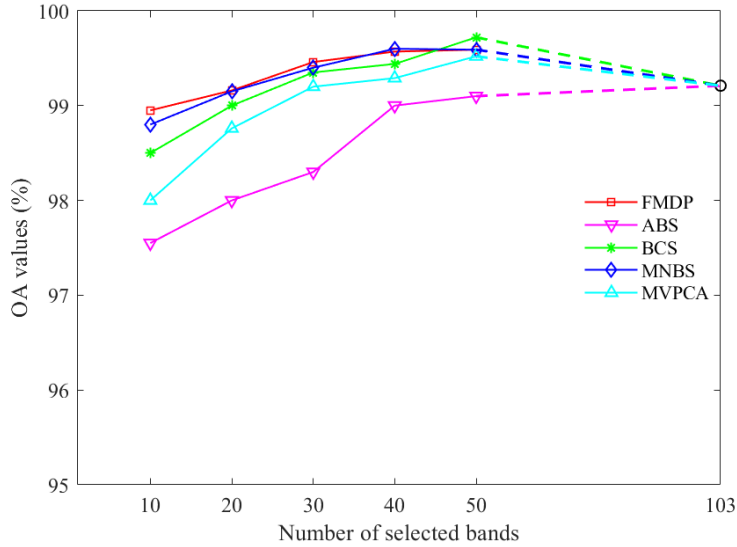


Figure 2.9: OA values of Pavia University based on different number of bands selected by different methods.

For Pavia University, we can find from Figure 2.9 that the OA values are affected by band selection methods, but the relationship between OA values and the number of selected bands is similar. At the beginning, the OA values show an upward tendency as the number of selected bands increases within a certain range. Then the OA values decrease as all bands are used except when ABS is used to select the band. Besides, when the number of selected bands is small, the proposed FMDP achieves higher accuracy compared with other mentioned methods. It's because the band selected by the proposed FMDP has a wide distribution and retains good spectral characteristics. When the number of selected bands exceeds 30, the classification results obtained by BCS, MNBS, MVPCA and the proposed FMDP are better than the results obtained with all bands, which proves that the appropriate number of selected bands can help to obtain better results compared to all bands being considered.

For Indian Pines (Figure 2.10), FMDP, BCS, MNBS and MVPCA have similar relationship between OA values and the number of selected bands. The OA values increase with the number of selected bands at the beginning, and then tend to be stable. By comparing with the OA values of all bands, it can be inferred that OA values decrease after a certain range. The OA values obtained based on ABS increase with the number of selected bands, but the OA values are low compared with the results got by other methods. When the selected band is fixed at 10, the proposed FMDP helps obtain the highest OA value. In addition, when the selected band is

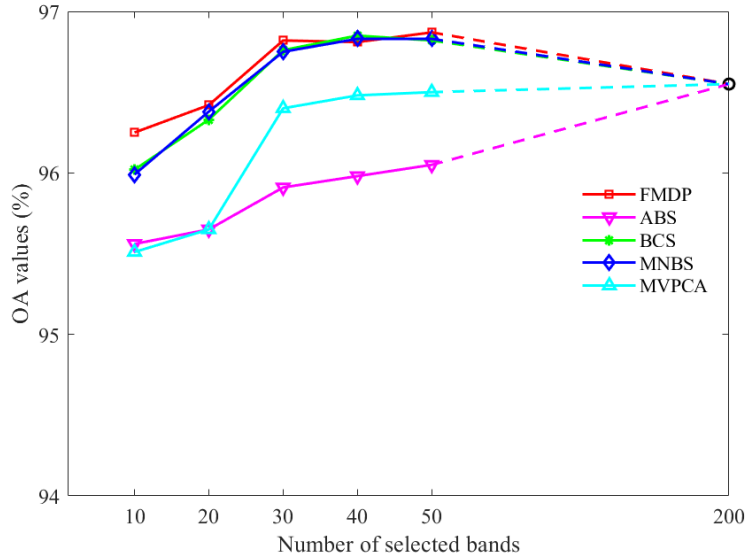


Figure 2.10: OA values of Indian Pines based on different number of bands selected by different methods.

between 30 and 50, the OA values of FMDP, BCS, and MNBS are similar and their performance surpasses the precision gained with all bands.

In general, the classification results of Pavia University and Indian Pines demonstrate that choosing appropriate number of bands instead of all bands for classifications helps obtain better results and improves efficiency. In particular, the proposed method exhibits best performance with a few iterations when the number of selected bands is small.

## 2.2.2 Solutions with limited labeled samples

After the HSI has been reduced in dimensionality, it can be input into 3D-CNN for feature extraction and classification. The designed framework is showed in Figure 2.11 where DR represents dimensionality reduction.

According to the unique variable principle described in subsection 1.3, the parameters of the 3D-CNN can be also tuned one by one to get a set of optimal parameters, which helps improve network performance. However, although the dimensionality of HSI is reduced, there are still a large number of parameters need to be optimized in a 3D-CNN, which requires sufficient labeled samples. Unfortunately, the labeled samples in hyperspectral data is limited and the collection of labeled samples is labor-consuming and time-consuming, which has a negative impact on the classification results.

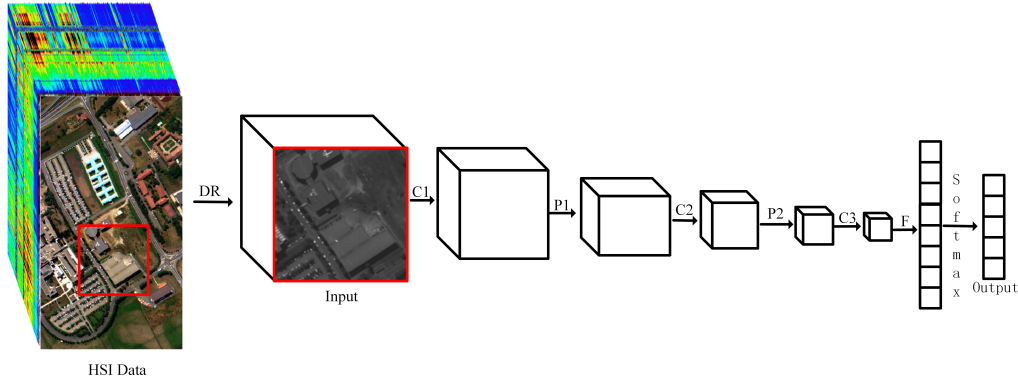


Figure 2.11: Proposed hyperspectral classification based on a 3D-CNN.

### 2.2.2.1 Transfer learning

Transfer learning (or knowledge transfer) has attracted widespread attention in recent years and has been widely applied in computer vision [101] and natural language processing [102], etc. To better understand transfer learning, the definitions of a “domain” and a “task” are given, respectively [22].

A domain  $\mathcal{D}$  consists of a feature space  $\mathcal{X}$  and a marginal probability distribution  $P(X)$ , where  $X = \{x_1, x_2, \dots, x_n\} \in \mathcal{X}$  and  $x_i$  represents the  $i$ th feature. In general, if two domains are different, then they may have different feature spaces or different marginal probability distributions. Given a specific domain,  $\mathcal{D} = \{\mathcal{X}, P(X)\}$ , a task ( $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$ ) is composed of a label space  $\mathcal{Y}$  and an objective function  $f(\cdot)$  which can be learned from the training data. The training data consists of pairs  $\{x_i, y_i\}$ , where  $x_i \in X$  and  $y_i \in \mathcal{Y}$ . For a new instance  $x$ ,  $f(\cdot)$  can be used to predict the corresponding label  $f(x)$ . From a probabilistic viewpoint,  $f(x)$  can be written as  $P(y|x)$ .

For simplicity, we only consider the condition where there is one source domain  $\mathcal{D}_S$ , and one target domain  $\mathcal{D}_T$ . The source data is denoted as  $\mathcal{D}_S = \{(x_{S_1}, y_{S_1}), (x_{S_2}, y_{S_2}), \dots, (x_{S_n}, y_{S_n})\}$ , where  $x_{S_i} \in \mathcal{X}_S$  is the data instance and  $y_{S_i} \in \mathcal{Y}_S$  is the corresponding label. Similarly, the target domain data can be denoted as  $\mathcal{D}_T = \{(x_{T_1}, y_{T_1}), (x_{T_2}, y_{T_2}), \dots, (x_{T_n}, y_{T_n})\}$ , where  $x_{T_i} \in \mathcal{X}_T$  is the input and  $y_{T_i} \in \mathcal{Y}_T$  is the corresponding output. Then, the definition of transfer learning can be defined as follows:

Given a source domain  $\mathcal{D}_S$  and learning task  $\mathcal{T}_S$ , a target domain  $\mathcal{D}_T$  and learning task  $\mathcal{T}_T$ , transfer learning aims to help improve the learning of the target predictive function  $f_T(\cdot)$  in  $\mathcal{D}_T$  using the knowledge in  $\mathcal{D}_S$  and  $\mathcal{T}_S$ , where  $\mathcal{D}_S \neq \mathcal{D}_T$ , or  $\mathcal{T}_S \neq \mathcal{T}_T$ .

From the definition of transfer learning, we can see that transfer learning has the ability of a system to recognize and apply knowledge and skills learned in previous domains/tasks to novel domains/tasks. There are some commonly used approaches such as instance-based transfer learning (or instance-transfer) approach, feature-representation-transfer approach, parameter-transfer approach, etc.

If there is another HSI (source data) with enough labeled samples and the same feature space as the HSI to be classified (target data), then transfer learning can be used to help us reduce the need for labeled samples of target data. The designed framework of 3D-CNN with transfer learning (3D-CNN-TL) is illustrated in Figure 2.12, where DR means dimensionality reduction.

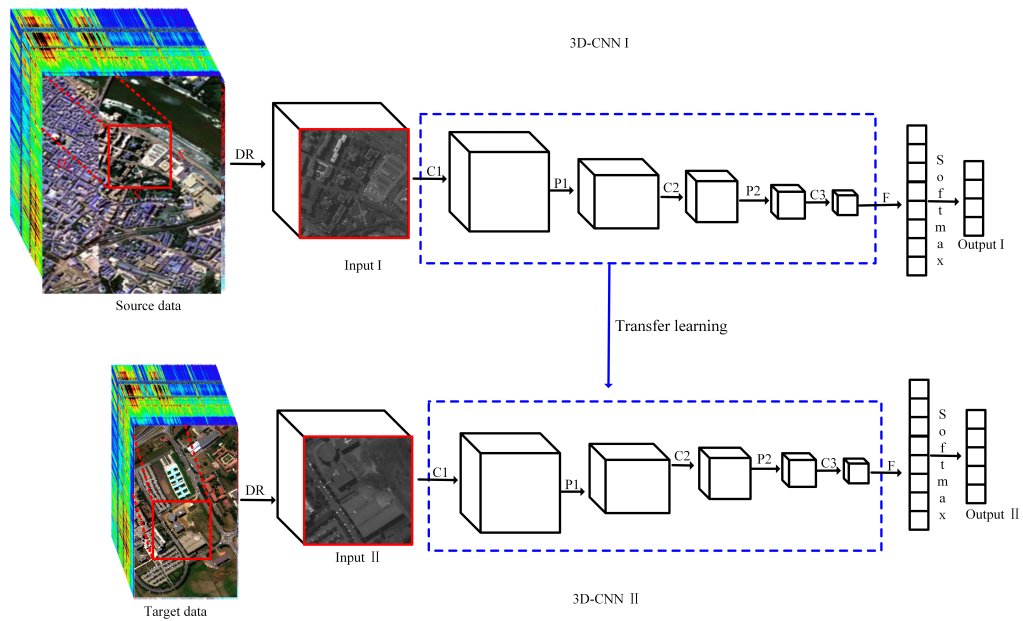


Figure 2.12: Proposed 3D-CNN-TL framework.

The training procedure of designed 3D-CNN-TL can be divided into five steps:  
Step 1: the source data and target data are reduced to the same dimension.

Step 2: two 3D-CNNs (3D-CNN I and 3D-CNN II) with same network structure except for the output layer are established, and the number of units in output layer is equal to the number of classes contained in the corresponding data set.

Step 3: the 3D-CNN I with source data as input is trained and optimized by sufficient training samples.

Step 4: knowledge transfer can be made: the weights in convolutional layers and pooling layers in the 3D-CNN II can be transferred from the same layers of the 3D-CNN I in Step 3, and the weights of other layers are initialized randomly.

Step 5: the 3D-CNN II can be further fine-tuned by the training samples from the

target data to obtain better classification performance. Besides, since the weights of the middle layers of 3D-CNN II are transferred from 3D-CNN I, the optimization of 3D-CNN II has lower requirements on the number of samples and the number of iterations.

### 2.2.2.2 Virtual samples

When the source data is available, knowledge transfer can be made from the source data to the target domain to improve the network performance by avoiding rather expensive data labeling efforts [103]. However, not all target data have corresponding source data. Without the help of source data, transfer learning cannot be implemented. If the source data is absent, as a pseudo-sample transformed from the original sample of the target data, virtual samples are also a solution to make up for the lack of HSI samples [21, 23].

If  $\varphi_o$  represents an original sample in the HSI with a size of  $S \times S \times H_{dr}$ , then the virtual sample  $\varphi_v$  can be defined as:

$$\varphi_v = \eta\varphi_o + \mathcal{N}(\mu, \sigma^2) \quad (2.4)$$

where  $\eta$  is the correlation coefficient, and  $\mathcal{N}(\mu, \sigma^2)$  denotes the Gaussian noise with a mean of  $\mu$  and a variance of  $\sigma^2$ , which is used to simulate the interference of the external environment to the samples. After mixing the virtual samples with the original ones, the overall number of training samples can be greatly increased.

### 2.2.2.3 3D-CNN with transfer learning and virtual samples

Since both transfer learning and virtual samples can make contributions to solve the problem of limited training samples, a hybrid method named 3D-CNN-TV which combines 3D-CNN, transfer learning, and virtual samples, is proposed in order to further improve HSI classification. Figure 2.13 shows the procedure of the proposed 3D-CNN-TV method, where a stadium box indicates the beginning and ending of a process, a parallelogram box denotes the process of inputting and outputting data, and a rectangular box represents a processing step.

The training procedure of designed 3D-CNN-TV can be divided into three steps:

First of all, 3D-CNN I is trained and optimized by sufficient training samples from the source data to obtain optimized weights. At the same time, virtual samples are generated from the original samples of target data according to Eq. (2.4). The

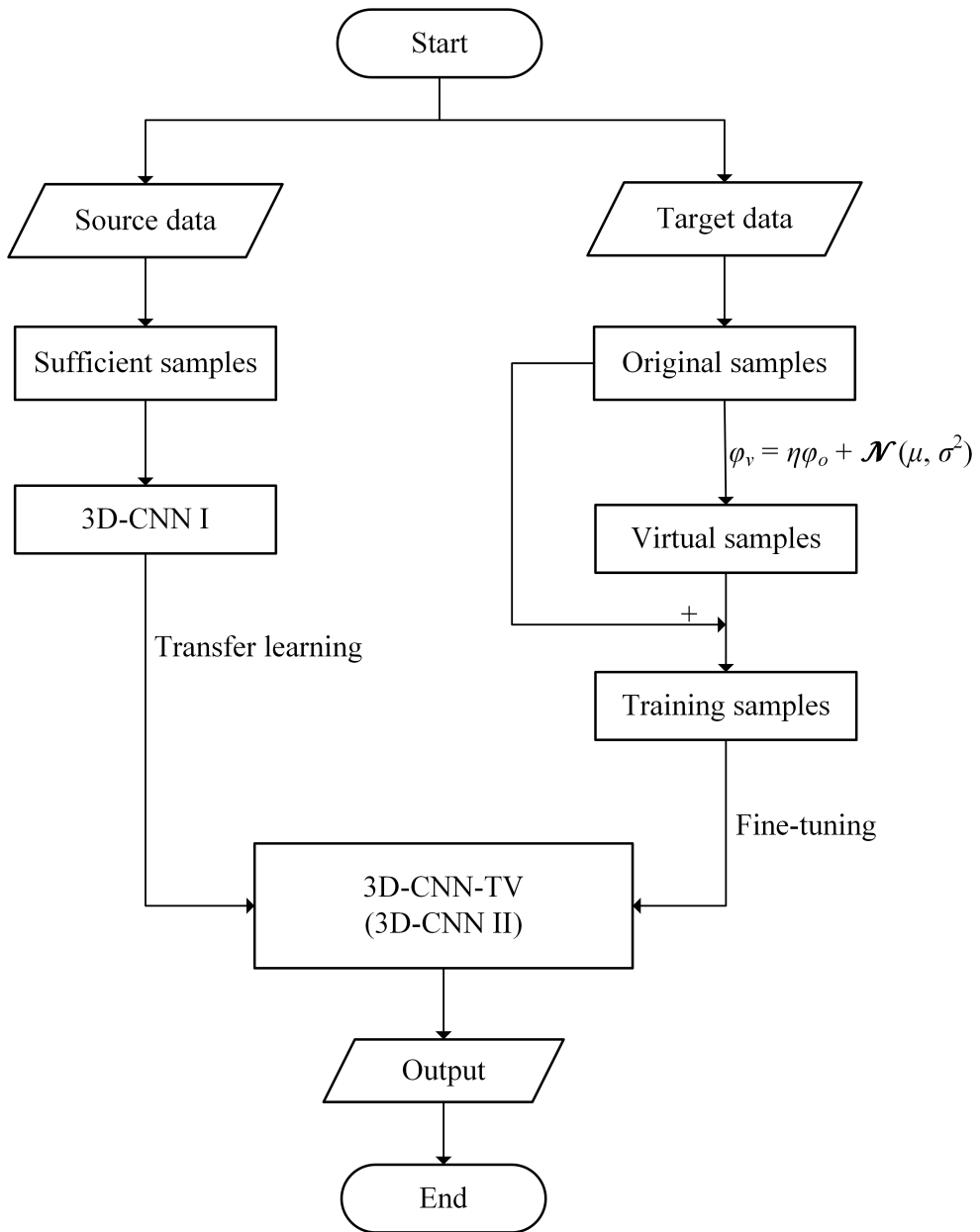


Figure 2.13: Flow chart of proposed 3D-CNN-TV framework.

generated virtual samples are mixed with the original samples as training samples of target data.

Then, as described in Figure 2.13, the weights of middle layers in 3D-CNN II (or 3D-CNN-TV) are transferred from the corresponding layers of the optimized 3D-CNN I.

At last, the training samples consisting of the original and the virtual ones can help to further fine-tune the 3D-CNN-TV model. In addition, due to the introduction of virtual samples, not only the number of training samples is greatly increased, but the robustness of the model can be improved.

## 2.3 Experimental results

In order to compare and verify the effectiveness of the proposed models, the classification results based on single 3D-CNN, 3D-CNN-TL, 3D-CNN-VS, and 3D-CNN-TV are analyzed and discussed in this subsection. In order to improve efficiency and reduce the amount of calculation, the dimensionality of the data sets involved in this subsection is reduced, and both PCA and proposed FMDP are used for comparison. Specifically, the original HSI ( $\mathbf{H} \in \mathbb{R}^{H_1 \times H_2 \times H_3}$ ) is reduced to a lower dimensional image ( $\mathbf{H}_{dr} \in \mathbb{R}^{H_1 \times H_2 \times H_{dr}}$ , ( $H_{dr} < H_3$ )). For each pixel, a 3D tensor with a size of  $S \times S \times H_{dr}$  is selected as the input of network. In the experiments, we choose Pavia University and Indian Pines as the target data to study and analyze the network performance in hyperspectral classification. The dimensionality of both two data sets is reduced to 10 by PCA and FMDP, respectively. The input size is fixed at  $27 \times 27 \times 10$ . In the following, the parameter settings of different models are introduced in detail.

### 2.3.1 Details of 3D-CNN

As described in Figure 2.11, the network structure of 3D-CNN is listed in Table 2.1 with input size being  $27 \times 27 \times 10$ . 128 is chosen as the batch size and Adam is chosen as the optimizer.

### 2.3.2 Details of 3D-CNN-TL

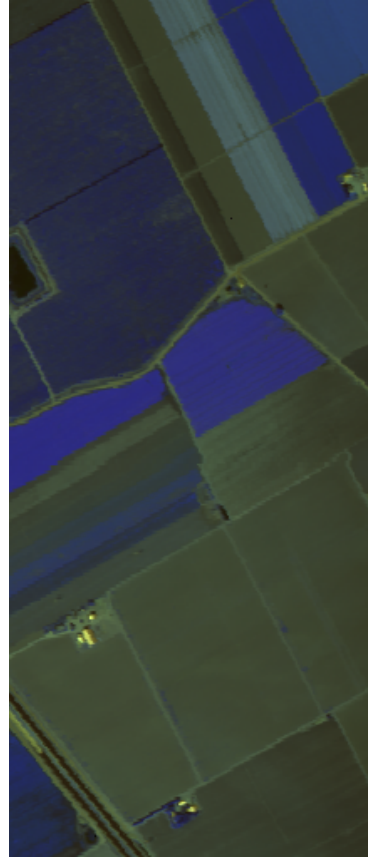
The center of Pavia city (Pavia Centre) shown in Figure 2.14 (a) is acquired by the ROSIS sensor, the same as the sensor that obtained Pavia University. After discarding the pixels containing no information, there are  $1096 \times 715$  pixels with

Table 2.1: Network structure of 3D-CNN.

Network layer	Convolutional layer	ReLU	Pooling laler	Dropout
1	$4 \times 4 \times 3 \times 16$	yes	$2 \times 2 \times 1$	-
2	$5 \times 5 \times 3 \times 32$	yes	$2 \times 2 \times 1$	0.2
3	$4 \times 4 \times 3 \times 64$	yes	-	0.2



(a)



(b)

Figure 2.14: Source data sets: (a) Pavia Centre, (b) Salinas.



102 spectral bands. Nine land-cover classes are contained in this data set and the details are listed in Table 2.2.

Table 2.2: Comparison of land-cover classes and number of samples in Pavia Centre and Pavia University.

Class No.	Pavia Centre		Pavia University	
	Name	Number	Name	Number
1	Water	65971	Asphalt	6631
2	Trees	7598	Meadows	18649
3	Asphalt	3090	Gravel	2099
4	Bricks	2685	Trees	3064
5	Bitumen	6584	Metal sheets	1345
6	Tiles	9248	Bare soil	5029
7	Shadows	7287	Bitumen	1330
8	Meadows	42826	Bricks	3682
9	Bare Soil	2863	Shadows	947

It can be found from Table 2.2 that Pavia Centre and Pavia University have seven common classes, such as Trees, Asphalt, Bricks, Bitumen, etc.

Salinas shown in Figure 2.14 (b) is collected by the 224-band AVIRIS sensor over Salinas Valley, California, the same as the sensor that obtained Indian Pines. The scene covers comprises 512 lines by 217 samples, including 16 land-cover classes. As with Indian Pines scene, 20 water absorption bands is discarded. It can be observed from Table 2.3 that although the land-over classes in Indian Pines are different from Salinas, the two data sets mainly contain agriculture, forest and vegetation.

Pavia Centre and Salinas are used as the corresponding source data for Pavia University and Indian Pines in the transfer learning experiment, respectively. We assume that both Pavia Centre and Salinas contain sufficient labeled samples.

To ensure the stability of the network performance, 70% of samples of each class in the Pavia Centre and Salinas are randomly chosen as the training set and the remaining 30% belong to the testing set. The network structures and parameter settings of 3D-CNNs are the same as in Table 2.1. When the two 3D-CNNs are well-trained by sufficient labeled samples from Pavia Centre and Salinas, respectively, the weights of the convolutional layers and the pooling layers are transferred to the corresponding 3D-CNN-TL (3D-CNN II) model. After transfer learning, 5% of samples of each class in target data are used to fine-tune the 3D-CNN-TL networks.

Table 2.3: Comparison of land-cover classes and number of samples in Indian Pines and Salinas.

Class No.	Indian Pines		Salinas	
	Name	Number	Name	Number
1	Weeds1	2009	Alfalfa	46
2	Weeds2	3726	Corn-notil	1428
3	Fallow	1976	Corn-min	830
4	Fallow-rough-plow	1394	Corn	237
5	Fallow-smooth	2678	Grass-pasture	483
6	Stubble	3959	Grass-trees	730
7	Celery	3579	Grass-pasture-mowed	28
8	Grapes-untrained	11271	Hay-windrowed	478
9	Soil-vinyard-develop	6203	Oats	20
10	Corn	3278	Soybean-notill	972
11	Lettuce-4wk	1068	Soybean-mintill	2455
12	Lettuce-5wk	1927	Soybean-clean	593
13	Lettuce-6wk	916	Wheat	205
14	Lettuce-7wk	1070	Woods	1265
15	Vinyard-untrained	7268	Buildings-grass-trees	386
16	Vinyard_vertical_trellis	1807	Stone-stel-towers	93

### 2.3.3 Details of 3D-CNN-VS

Virtual samples are introduced to the 3D-CNN model according to Eq. (2.4). To reduce the difference between the virtual samples and real samples, the value of correlation coefficient  $\eta$  should be closed to 1. Therefore,  $\eta$  is set to a uniformly distributed random number in  $[0.9, 1.1]$  in the experiment. Considering that the number of virtual samples and the interference  $\mathcal{N}$  can also influence the network performance, a sensitivity analysis is conducted in this part to achieve better network performance.

If the number of original training samples selected from among the target data is  $n_t$ , then the number of virtual samples will be  $n_v = r \times n_t$  where  $r$  represents the ratio between the number of virtual samples and the number of original samples. The mean value  $\mu$  of noise  $\mathcal{N}$  is set to 0, and the variance  $\sigma^2$  is set to 0.01 at the beginning. In the experiment, the virtual samples and the original samples are mixed together to form the training data set. When the value of the ratio  $r$  is different, i.e., when the number of virtual samples is different, Figure 2.15 (a) shows the relationship between  $r$  and OA values for Pavia University and Figure 2.15 (b) for Indian Pines.

It can be seen from Figure 2.15 that for Pavia University and Indian Pines, the

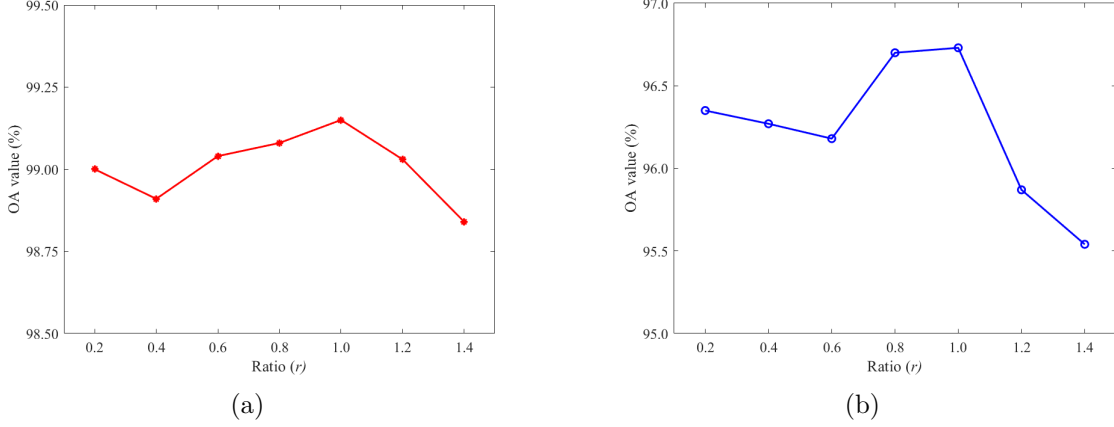


Figure 2.15: Relationship between  $r$  and OA values.

OA values increase first and then decrease with  $r$ . For both data sets, the highest value is arrived when the number of virtual samples is  $1 \times n_t$ , i.e., the number of virtual samples is equal to the number of original samples. Therefore, the number of virtual samples is set to  $n_v = n_t$  for the 3D-CNN-VS model in the classification.

When introducing virtual samples, the noise  $\mathcal{N}$  can also affect the classification performance. Keeping the number of virtual samples fixed at  $n_v = n_t$  and the mean value of  $\mathcal{N}$  at 0, and changing the variance value  $\sigma^2$ , the resulting OA values are shown in Table 2.4, where  $OA_1$  represents the OA values of Pavia University and  $OA_2$  represents the OA values of Indian Pines.

Table 2.4: OA values under different noise variances of the virtual samples.

$\sigma^2$	0.00001	0.0001	0.001	0.01	0.1	1
$OA_1$	98.31	98.52	99.15	98.15	98.55	98.22
$OA_2$	96.60	96.64	97.75	96.73	96.30	95.81

As presented in Table 2.4, the OA value of Pavia University is relatively high when the value of  $\sigma^2$  is less than 0.001. Besides, the highest OA values of Pavia University and Indian Pines are both obtained with  $\sigma^2$  being 0.001, which means that the virtual samples are more similar to the original samples at this point.

### 2.3.4 Details of 3D-CNN-TV

As mentioned in Section 2.2.2.3, a 3D-CNN-TV model combined with transfer learning and virtual samples can be constructed for the classification. Meanwhile, the virtual samples with zero mean and noise variance of 0.001 could be generated from the original samples. Then, the virtual samples are mixed with the original

ones to fine-tune the 3D-CNN-TL model with transferred weights. When the 3D-CNN-TV is well optimized, input the target data into the optimized 3D-CNN, and the category prediction can be obtained.

### 2.3.5 Comparison of classification results

In order to make a visual comparison, the classifications maps of Pavia University and Indian Pines obtained from 3D-CNN-based models are illustrated in Figure 2.16 and Figure 2.17.

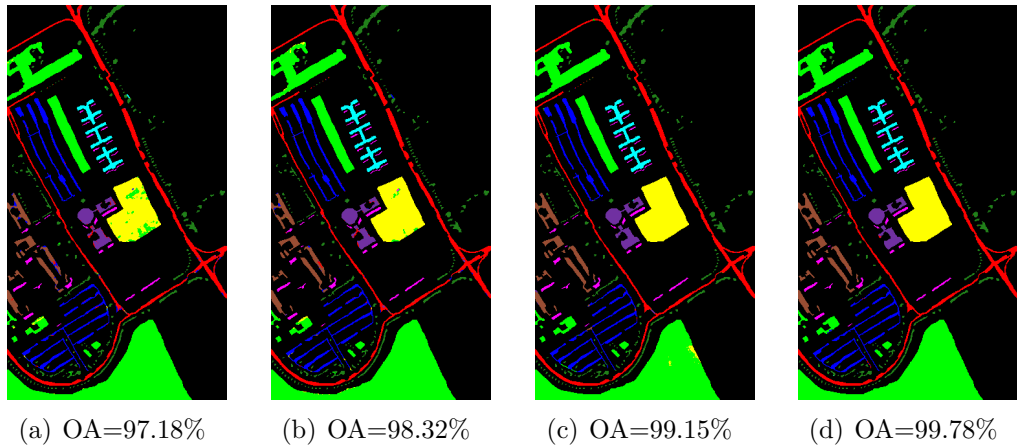


Figure 2.16: Classification maps of Pavia University under different methods: (a) 3D-CNN, (b) 3D-CNN-TL, (c) 3D-CNN-VS, (d) 3D-CNN-TV.

It can be seen from Figure 2.16 that the classification map of 3D-CNN has more pixels misclassified compared with the other three classification maps, especially the yellow area. Both the introduction of transfer learning as shown in Figure 2.16 (b) and virtual samples as shown in Figure 2.16 (c) can reduce the number of misclassified pixels. Besides, virtual samples are more helpful for improving the classification performance for Pavia University. Moreover, the proposed 3D-CNN-TV method helps to obtain the clearest map and highest OA value.

For Indian Pines, we can find from Figure 2.17 that the misclassified pixels are mainly concentrated in the upper left area. The misclassification of pixels is greatly reduced in Figure 2.17 (b) - (d), which demonstrate that transfer learning and virtual samples have great potential in further improve the network performance.

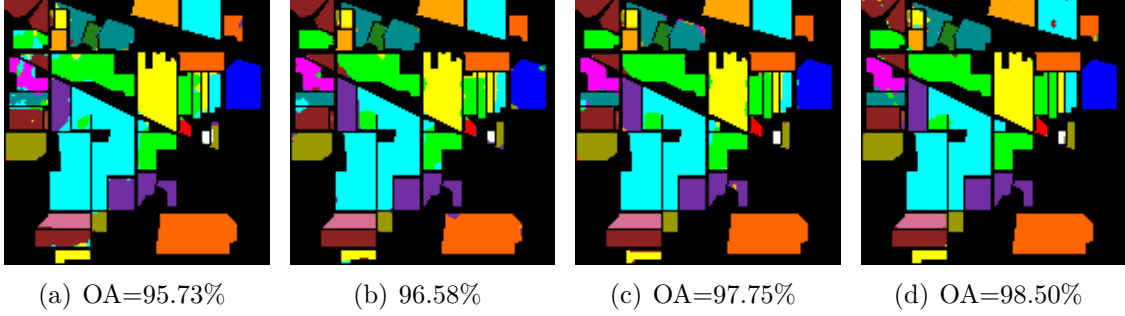


Figure 2.17: Classification maps of Indian Pines under different methods: (a) 3D-CNN, (b) 3D-CNN-TL, (c) 3D-CNN-VS, (d) 3D-CNN-TV.

## 2.4 Conclusion

When 3D-CNN is introduced for hyperspectral classification, due to the network structure and the data characteristics of HSIs, a large amount of calculation is caused. To solve this problem, we propose a new band selection method to quickly select the band and reduce the dimensionality of HSIs. In addition, to solve the problem of insufficient samples, improved 3D-CNNs based on transfer learning and virtual samples are proposed for HSI classification. On the one hand, the initial weights in the middle layers are transferred from another 3D-CNN which has been well trained by the source data. On the other hand, virtual samples are generated from the original samples in the target data to increase the number of training samples. Experimental results show either transfer learning or virtual samples can help us further improve the classification accuracy, and the OA values obtained by 3D-CNN with virtual samples are higher for both two data sets. Besides, it is relatively easy to obtain virtual samples compared with transfer learning, because training the network with source data requires a large amount of calculation and time. In general, the combination of transfer learning and virtual samples further improves network performance and achieves the highest accuracy.

## Chapter 3

# Unsupervised feature extraction based on GAN for hyperspectral classification

### 3.1 Introduction

In the previous Chapter, transfer learning and virtual samples are investigated to alleviate the problem of limited labeled samples in HSIs. However, the models mentioned above are all supervised feature extraction, which means that the training process still requires the participation of labeled samples. Unsupervised feature extraction which doesn't involve labeled samples is another good way to help us get rid of labeled data. Considering the powerful data mining capabilities of deep learning models, unsupervised feature extractors based on deep learning models can be designed to fully exploit the nonlinear and spectral-spatial features of HSIs without labeled samples.

GAN is trained in an adversarial way requiring no labeled samples. It has been one of the most promising unsupervised learning representatives [23]. In [46], a semi-supervised framework based on 1D-GAN is established for hyperspectral classification with a small number of labeled samples. But only spectral features are extracted, which are far from enough for classification. In [47], Zhang proposes a novel modified GAN whose generator and discriminator are designed in the form of fully deconvolutional network and fully convolutional network to extract the features without supervision. Nevertheless, only spatial information is taken as input when the modified GAN is trained, which can be treated as 2D convolution on

multiple channels. Hyperspectral data is a tensor data, which contains not only spatial information but also the spectral characteristics of the target. Fully mining the spectral-spatial features in HSIs is helpful for classifying the target. Considering 3D convolution operation is performed in space and spectrum, we want to design a framework based on 3D-GAN in which the generator and discriminator are built on fully 3D convolution and 3D deconvolution subnetworks to fully extract the spectral-spatial features with unsupervised learning for classification.

## 3.2 Overview of GAN

GAN is proposed by Goodfellow et al. [104], mainly including a generator and a discriminator. Generator can capture the probability distributions of real data  $x$ , by producing synthetic data from given some noise source  $z$ ; Discriminator estimates whether the sample is real or generated. The architecture of standard GAN is shown in Figure 3.1, where G represents generator and D represents discriminator.

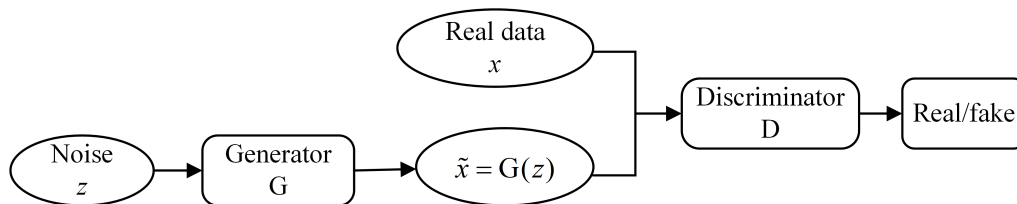


Figure 3.1: Standard GAN.

Given a low-dimensionality latent data  $z$  with the probability distribution being  $p_z$  ( $z \sim p_z$ ), and it can be projected to  $\tilde{x}$  whose probability distribution is  $p_g$ , where generator is a multilayer perceptron. Similarly, another multilayer perceptron can be represented as discriminator. Real data  $x$  with probability distribution  $p_r$  or generated data  $\tilde{x}$  is input into discriminator, where discriminator can output a probability that the input belongs to real data. Generator and discriminator play a minmax game and objective function is as follows:

$$\min_G \max_D E_{x \sim p_r} [\log D(x)] + E_{\tilde{x} \sim p_g} [\log (1 - D(G(z)))] \quad (3.1)$$

where E represents mathematical expectation.  $E_{x \sim p_x}$  represents the expectation over real data  $x$  with probability distribution being  $p_x$ .  $E_{\tilde{x} \sim p_g}$  means the expectation over noisy data  $\tilde{x}$  with probability distribution being  $p_g$ .

Through the adversarial manner and competition of two models, both the generator and the discriminator will be continuously optimized. However, in the training

procedure of standard GAN, Jensen-Shannon divergence is used to minimize the difference between the probability distributions of generated data and real data, which makes the training of GANs is well known for being delicate and unstable, and often leads to vanishing gradients as the discriminator saturates [105, 106].

Wasserstein GAN (WGAN) uses Wasserstein distances to calculate the distances between different distributions and designs weight clipping to enforce a Lipschitz constraint, which makes progress toward stable training of GANs and get rid of mode collapse [107, 108]. However, the use of weight clipping of WGAN leads to optimization difficulties, because weight clipping makes the weights of discriminator almost concentrated in the extremes of the clipping range as shown in Figure 3.2. Even worse, gradient explode or gradient vanish may be caused.

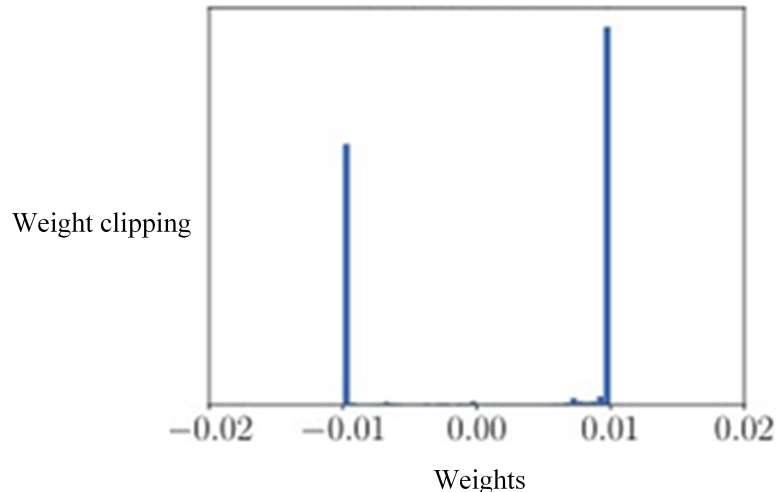


Figure 3.2: Weight distribution with weights clipping.

To solve the aforementioned problems, an improved WGAN adding gradient penalty to enforce the Lipschitz constraint, named WGAN-GP, has been developed and has been demonstrated that gradient penalty is an effective way to solve exploding and vanishing gradients. The new objective is [109]:

$$L = \underbrace{\mathbb{E}_{\tilde{x} \sim p_g} [D(\tilde{x})] - \mathbb{E}_{x \sim p_r} [D(x)]}_{\text{Original critic loss in WGAN}} + \underbrace{\lambda \mathbb{E}_{\hat{x} \sim p_{\hat{x}}} [(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]}_{\text{Gradient penalty}} \quad (3.2)$$

where  $\lambda$  is a gradient penalty coefficient, and  $p_{\hat{x}}$  samples uniformly along straight lines between pairs of points sampled from the  $p_g$  and  $p_r$ . It has been demonstrated that the training of WGAN-GP is more stable, and gradient penalty is an effective way to solve exploding and vanishing gradients.



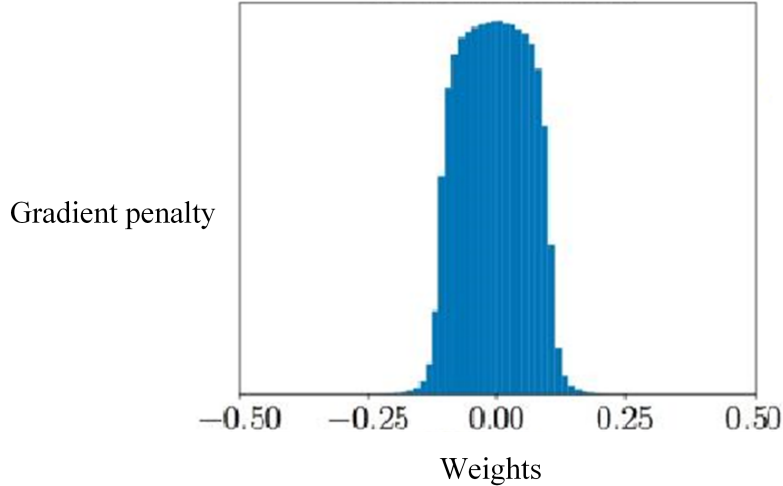


Figure 3.3: Weight distribution with gradient penalty.

### 3.3 Proposed unsupervised feature extraction method based on 3D-WGAN-GP

We have learned that HSI is a high-dimensional data which contains hundreds of spectral bands and it's difficult for generator to produce high-dimensional data, which makes the training of GAN is difficult. In Chapter 2, a band selection method is proposed to reduce the dimension of HSIs. In this Chapter, a new dimensionality reduction method is designed based on  $1 \times 1$  convolution and transfer learning, which can provide an alternative way for data dimensionality reduction.

#### 3.3.1 Proposed dimensionality reduction method

When dimensionality reduction is conducted, the spatial size (height and width) of HSI is unchanged and only the dimension corresponding to the band (depth) is reduced. Considering this characteristic of dimensionality reduction and being inspired in [110], we propose to use  $1 \times 1$  convolutions to obtain lower-dimensional and more abstract features. Furthermore, we want to reduce the dimension of target data through transfer learning with an unsupervised process, which means labeled samples of target data are not required during the dimensionality reduction process. The designed framework of dimensionality reduction is depicted in Figure 3.4.

Take Pavia University as an example, we give a specific explanation of the dimensionality reduction framework. Since the number of spectral bands is 103 for Pavia University and 102 for Pavia Centre, to make the target data and the corre-

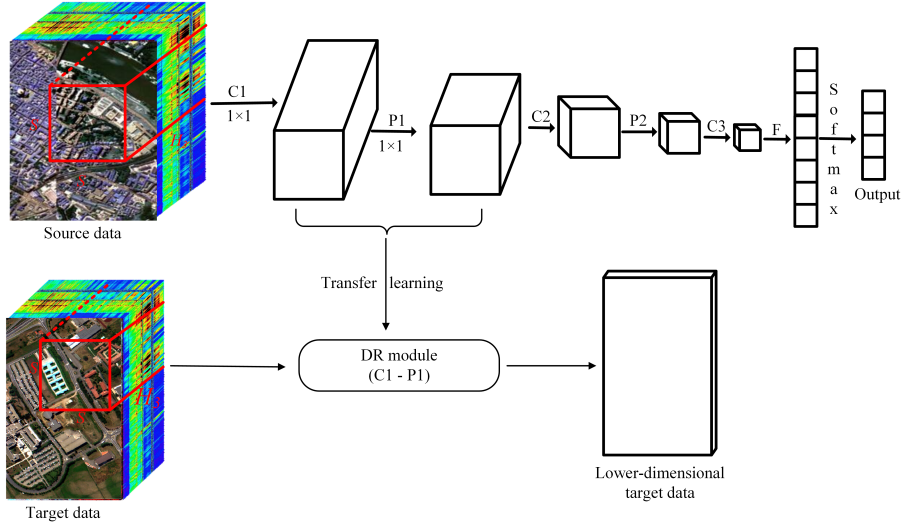


Figure 3.4: Framework of proposed dimensionality reduction method.

Corresponding source data have the same number of bands, the 103-rd spectral dimension of the Pavia University is removed. Then, the original band of Pavia University and Pavia Centre becomes 102. The network structure of 3D-CNN corresponding to the source data is listed in Table 3.1.

Table 3.1: Network structure of 3D-CNN.

Network layer	Convolutional layer	ReLU	Pooling layer	Output
1	$1 \times 1 \times 83 \times 1$	yes	$1 \times 1 \times 2$	$27 \times 27 \times 10 \times 1$
2	$4 \times 4 \times 3 \times 16$	yes	$2 \times 2 \times 2$	$12 \times 12 \times 4 \times 16$
3	$5 \times 5 \times 3 \times 32$	yes	$2 \times 2 \times 2$	$4 \times 4 \times 1 \times 32$
4	$4 \times 4 \times 1 \times 64$	yes	—	$1 \times 1 \times 1 \times 64$

The input size  $S \times S \times H_3$  is set to  $27 \times 27 \times 102$  at the beginning and the stride is 1. The output size of feature map in the first convolutional layer is  $(27 - 1 + 1) \times (27 - 1 + 1) \times (102 - 83 + 1) = 27 \times 27 \times 20$ . After pooling operation, the size of the corresponding feature maps is  $27 \times 27 \times 10$ . It can be found that the height and width of the input data are not changed after the first layer of convolution and pooling operations, and only the depth (spectral dimension) is reduced. The 3D-CNN is trained by source data which is assumed to have sufficient labeled samples. During the training process, the parameters of the network are continuously optimized. When the 3D-CNN is well-trained, the first convolutional layer and pooling layer can be transferred as the dimensionality reduction module. When the target data is input to this module, the lower-dimensional data can be obtained.

### 3.3.2 Details of proposed unsupervised feature extraction method

To extract the spectral-spatial features in HSIs with an unsupervised process, a 3D convolution model based on WGAN-GP is designed and the framework of the proposed method is shown in Figure 3.5, where deconv represents fractional-strided convolutions (or deconvolution operation), conv means strided convolution operation, G and D represent generator and discriminator, respectively.

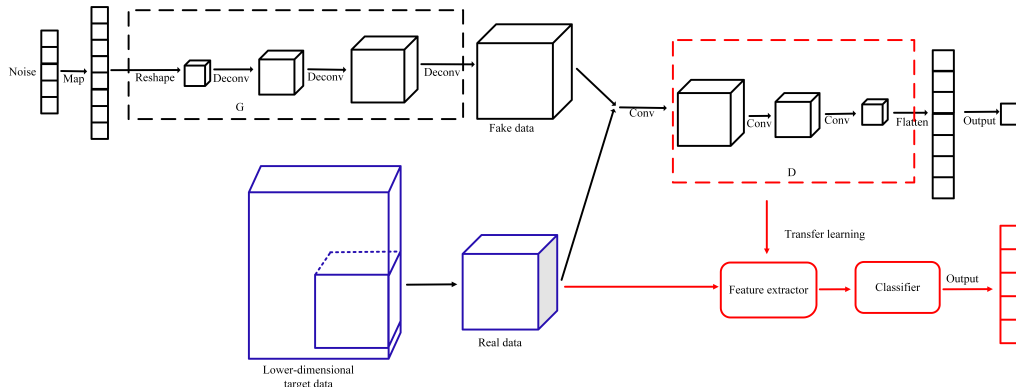


Figure 3.5: Proposed unsupervised feature extraction method based on 3D-WGAN-GP.

The process in figure 3.5 can be divided into three main steps:

First, a 3D-WGAN-GP model is constructed. Since convolution operation shows great advantages in feature extraction, the model is built in all convolutional net where pooling operation is replaced by fractional-strided convolutions (deconv) in G and strided convolution (conv) in D. No fully-connected layers are used, and the last convolution layer is flattened to fed into a single output.

Next, train and optimize the 3D-WGAN-GP to improve its performance. The real samples from lower dimensionality target data and the generated samples mapping from noise vector through generator have the same size, and they are treated as inputs of discriminator. During the training process, the fake data is getting closer to real data, and discriminator has more and more difficulties in distinguishing between real data and fake data. When the model is optimized to reach a point where the discriminator is unable to differentiate real data and generated data, we think the generator has learned the distribution of real data and the discriminator has strong feature extraction ability.

Finally, transfer the optimized discriminator as the feature extractor. It can be observed that generator can map 1D vector to high-dimensional data. Discrimina-

tor can convert high-dimensional data into low-dimensional data, which is consistent with the goal of feature extraction. Therefore, the optimized discriminator can be taken out separately as the feature extractor. Input the real target data into the feature extractor, and the corresponding low-dimensional data with high-level features are obtained which lays a foundation for subsequent classification. Considering no labeled samples are used during this process, unsupervised feature extraction is implemented.

### 3.3.3 Experimental results

To verify the performance of the proposed method, Pavia University and Indian Pines are chosen as target data, and Pavia Centre and Salinas are used as source data as Section 2.3.2. According to the previously proposed dimensionality reduction method, the dimension of the target data is reduced to 10 with the help of source data and transfer learning. For each pixel, a 3D block with size being  $27 \times 27 \times 10$ . According to the size of input data, a 3D-WGAN-GP is built and some details are described in Table 3.2, where Af represents activation function.

Table 3.2: Architectures of the 3D-WGAN-GP.

Net	Layer	Conv	Af	Net	Layer	Deconv	Af
D	1	$4 \times 4 \times 3 \times 16$	leakyReLU	G	1	$5 \times 5 \times 3 \times 128$	ReLU
	2	$5 \times 5 \times 3 \times 32$	leakyReLU		2	$5 \times 5 \times 3 \times 64$	ReLU
	3	$5 \times 5 \times 3 \times 64$	leakyReLU		3	$5 \times 5 \times 3 \times 32$	ReLU
	4	$5 \times 5 \times 3 \times 128$	leakyReLU		4	$4 \times 4 \times 3 \times 16$	ReLU

It can be seen from Table 3.2 that all activation functions of layers in discriminator are leaky rectified linear unit (leakyReLU). ReLU is chosen as the activation function of layers in generator, except the output layer, which uses hyperbolic tangent function (Tanh). Besides, in convolution-related operations, the stride is set as  $1 \times 1 \times 1$  and the padding is set as valid which means there is no zero padding operation on the boundary data. The generated samples obtained from the uniformly distributed noise with size being  $100 \times 1$  through generator are mixed with the real samples of source data, and then are fed into discriminator. In the training procedure, dropout is introduced to avoid overfitting and a mini-batch based on the root mean square prop (RMSProp) algorithm [111] which performs well even on very nonstationary problems is employed and the size of batch is set as 32.

To evaluate the performance of the proposed method, the unsupervised feature

extraction methods based on standard GAN, 3D-WGAN [47] and the proposed 3D-WGAN-GP are compared. However, it’s difficult to directly evaluate the extracted features. Therefore, the classification performance based on the extracted features is used to estimate the quality of the extracted features. Better classification results reflect that the corresponding methods have stronger feature extraction ability. In the experiment, two widely used classifiers, support vector machine (SVM) and softmax, are used to classify the features. 10% of the samples of each class are randomly chosen to train the classifier and the remaining 90% are for testing. Classification accuracy of a single land-cover class and OA values are used to assess the classification performance.

The comparison results of Pavia University and Indian Pines are listed in the Table 3.3 and Table 3.4, respectively.

Table 3.3: Classification accuracy of Pavia University under different methods.

Class \ Model	GAN - softmax	GAN - SVM	WGAN - softmax	WGAN - SVM	WGAN -GP-softmax	WGAN -GP- SVM
Asphalt	92.83	98.08	96.67	98.49	96.89	98.73
Meadows	82.52	83.04	93.09	82.70	93.76	92.14
Gravel	98.24	99.66	98.90	99.65	99.57	99.59
Trees	95.40	93.51	98.01	98.10	98.73	99.05
Metal sheets	99.48	99.90	99.92	99.85	97.99	99.78
Bare Soil	90.91	90.43	82.16	93.77	93.77	99.54
Bitumen	85.56	84.06	77.29	66.70	92.71	90.45
Bricks	88.73	95.27	90.79	98.75	95.30	98.61
Shadow	92.93	70.33	96.20	95.99	97.46	95.78
OA (%)	94.27	95.57	94.84	96.67	97.45	98.60
AA (%)	91.84	90.48	92.56	92.70	96.24	97.08
$\kappa$ (%)	92.40	94.10	93.11	95.57	96.62	98.15

From Table 3.3, it can be seen that the classification accuracy of meadows, bare soil and bitumen is relatively low compared to other categories, which may be caused by large intra-class variation and small interclass variation. The features obtained by 3D-WGAN-GP help obtain higher OA, AA and  $\kappa$  values compared with the features obtained by GAN and 3D-WGAN. Besides, SVM performs better than softmax on Pavia University.

From Table 3.4, it can be seen that when the features obtained by GAN are used for classification, the classification accuracy of many land-cover classes is less than 90%, which is unsatisfactory. When the features obtained by 3D-WGAN and 3D-WGAN-GP are used for classification, the classification results have been greatly

Table 3.4: Classification accuracy of Indian Pines under different methods.

Class \ Model	GAN - softmax	GAN - SVM	WGAN - softmax	WGAN - SVM	WGAN -GP-softmax	WGAN -GP- SVM
Weeds1	93.47	84.78	91.30	86.96	91.30	93.48
Weeds2	85.92	82.49	85.29	89.36	91.94	94.26
Fallow	80.36	91.32	90.36	89.28	94.58	98.07
Fallow-rough	77.63	69.62	78.06	81.43	89.87	87.34
Fallow-smooth	92.34	92.96	96.27	92.13	87.99	92.75
Stubble	95.62	97.67	98.77	98.22	97.67	96.44
Celery	60.71	64.29	75.00	71.43	75.00	100.00
Grapes-untrained	98.12	98.54	97.49	96.86	100.00	100.00
Soil-vinyard	60.00	85.00	80.00	40.00	100.00	75.00
Corn	86.32	91.77	95.16	92.28	90.95	96.29
Lettuce-4wk	91.32	94.70	95.64	94.91	95.93	96.01
Lettuce-5wk	81.96	79.59	87.02	89.04	89.71	95.28
Lettuce-6wk	98.05	98.54	99.89	99.89	98.05	97.56
Lettuce-7wk	97.71	95.65	96.21	98.74	98.33	98.26
Vinyard-untrained	88.86	86.01	99.48	95.60	100.00	98.45
Vinyard-vertical	91.40	79.57	78.49	77.42	93.54	97.85
OA (%)	89.72	90.89	93.20	93.21	94.63	96.16
AA (%)	86.23	87.03	90.28	87.10	93.43	94.82
$\kappa$ (%)	88.25	89.59	92.23	92.24	93.87	95.61

improved, which indicates the potential of 3D convolution-based operations in mining spatial-spectral features. Besides, the highest OA, AA, and  $\kappa$  values are obtained based on the features extracted by 3D-WGAN-GP with SVM as the classifier.

In general, the classification accuracy of two data sets is well when using the features extracted by the 3D-WGAN and 3D-GAN-GP for classification. In other words, convolution operation shows greater potential than multilayer perceptron in feature extraction. The classification accuracy of most classes based on the proposed 3D-WGAN-GP is better than other models, especially when SVM is used as a classifier.

Next, the classification results based on the proposed dimensionality reduction method are compared with the results obtained using PCA. Features extracted from dimensionality-reduced data obtained by different methods are tested with GAN-based models. The classification performance with accuracy values is shown in Table 3.3.3, where “A + B” indicates the combination of dimensionality reduction method “A” and model “B”, DR represents the proposed method.

Table 3.5: Classification results of different GANs combining PCA or proposed dimensionality reduction method.

Model	OA (%)	Pavia University			Indian Pines		
	OA(%)	AA(%)	$\kappa$ (%)	OA(%)	AA(%)	$\kappa$ (%)	
DR+GAN	95.57	90.48	94.10	90.89	87.03	92.24	
DR+WGAN	96.67	92.70	95.57	93.21	87.10	92.24	
DR+WGAN-GP	98.60	97.08	98.15	96.16	94.82	95.61	
PCA+GAN	95.26	93.57	93.37	88.74	83.40	87.05	
PCA+WGAN	95.73	95.69	94.73	92.35	89.74	91.18	
PCA+WGAN-GP	95.96	95.74	95.19	93.74	88.45	93.31	

From Table 3.3.3, we can see the differences between the proposed method (DR+WGAN-GP) and others. The proposed method improves about 2.3%, 2%, 2.4% in OA, AA and  $\kappa$  values for Pavia University and 3.5%, 5%, and 3.4% for Indian Pines compared with PCA+WGAN. When the proposed method is compared with PCA+GAN, the differences are larger, which are 2.8%, 3.8%, 3.8% for Pavia University and 7%, 12%, 7.5% for Indian Pines. In addition, since the proposed method consists the DR and the proposed feature extraction, the impact of DR or WGAN-GP on the results are analyzed separately. When PCA is used to reduce the dimension of HSIs, for Pavia University dataset, the results obtained with features extracted by GAN, WGAN or WGAN-GP are similar. For Indian Pines, the OA values improved 1-2% with the improvement of the feature extraction models.

When the DR is used to reduce the dimension of HSIs, the differences of the results got by GAN, WGAN and WGAN-GP are bigger than those got with PCA for both datasets. But the improvements are still insignificant. When the feature extraction model is designed based on WGAN-GP, the classification accuracy with DR increases about 2% for the two datasets compared with PCA. The proposed DR method is helpful for the WGAN-GP performance.

In general, if we only use the proposed DR or the proposed feature extraction method, the differences are not significant compared with other methods. But the combining of the proposed DR and the proposed unsupervised feature extraction method helps to obtain better results, which shows great potential in HSI classification.

In order to make a visual comparison, the classifications maps of Pavia University obtained by combining the proposed dimensionality reduction method and different GAN-based models with different classifiers are illustrated in Figure 3.6.

It can be seen from Figure 3.6 that the classification maps of GAN-SVM and GAN-softmax have more pixels misclassified in Figure 3.6 (c) to Figure 3.6 (d), especially the classes corresponding to yellow and green areas. In Figure 3.6 (e) to Figure 3.6 (g), some samples of the yellow area are incorrectly classified as green or blue. In Figure 3.6 (h), although there are still some points of misclassification, it's the closest to the ground truth image, which shows that the proposed method based on 3D-WGAN-GP is more promising in unsupervised feature extraction.

The classifications maps of Indian Pines obtained by combining the proposed dimensionality reduction method and different GAN-based models are illustrated in Figure 3.7.

It can be seen that the classification maps in Figure 3.7 (c) and (d) have many misclassified pixels in the upper part of the image. The classification maps of Figure 3.7 (e) - (h) have fewer misclassified pixels compared to Figure 3.7 (c) and (d), which proves the ability and potential of 3D convolution in feature extraction.

### 3.4 Conclusion

High dimensionality and limited labeled samples are two issues we have to face when we classify hyperspectral data. In order to reduce the dimension of HSIs, a novel dimensionality reduction method based on  $1 \times 1$  convolutions and  $1 \times 1$  pooling is proposed to obtain lower dimensionality data containing more abstract and high-level features. In order to get the rid of the limitation of labeled samples, an



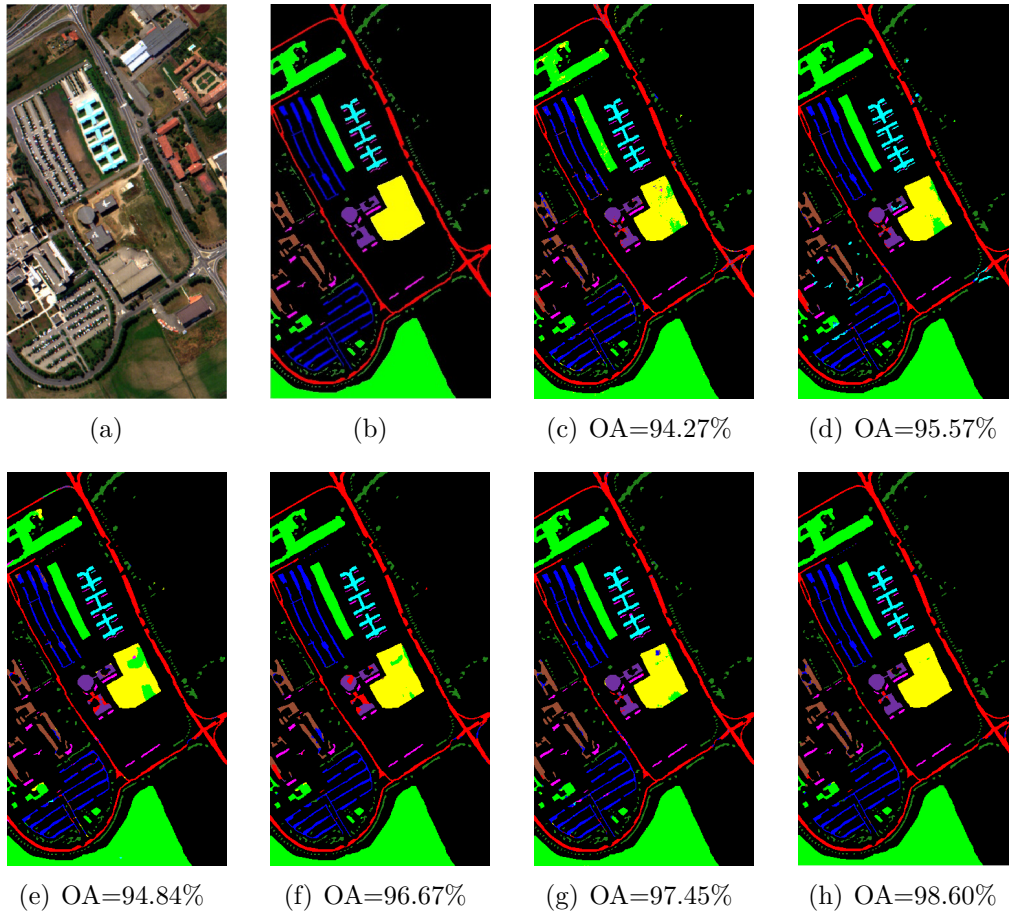


Figure 3.6: Classification maps of Pavia University under different methods: (a) False-color image, (b) Ground truth, (c) GAN-softmax, (d) GAN-SVM, (e) 3D-WGAN-softmax, (f) 3D-WGAN-SVM, (g) 3D-WGAN-GP-softmax, (h) 3D-WGAN-GP-SVM.

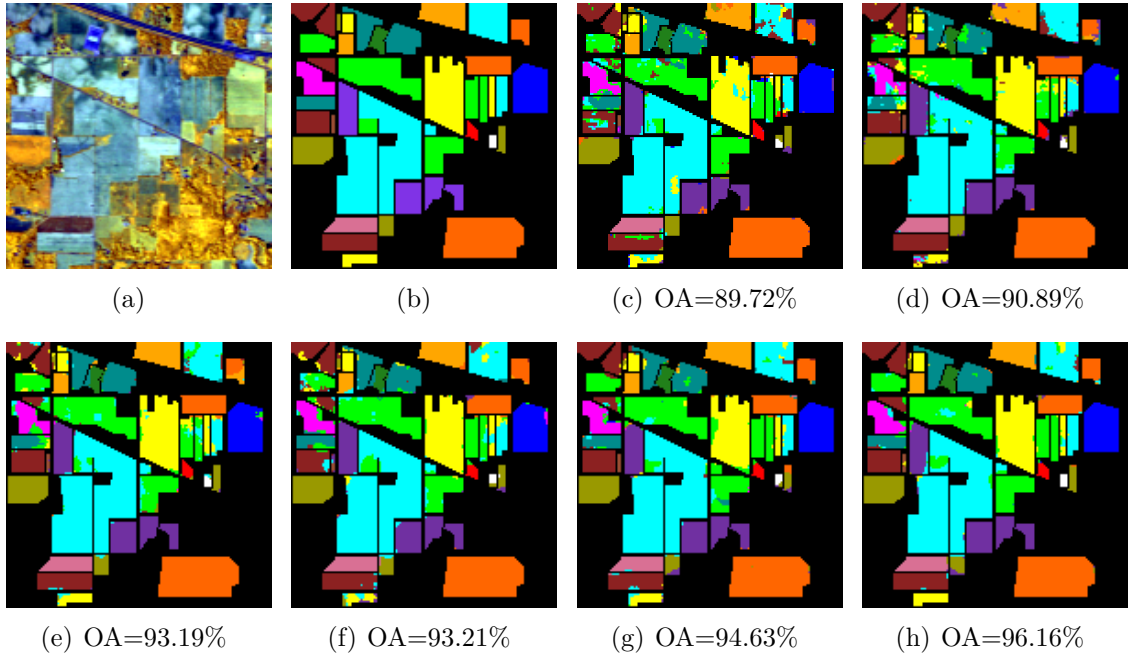


Figure 3.7: Classification maps of Indian Pines under different methods: (a) False-color image, (b) Ground truth, (c) GAN-softmax, (d) GAN-SVM, (e) 3D-WGAN-softmax, (f) 3D-WGAN-SVM, (g) 3D-WGAN-GP-softmax, (h) 3D-WGAN-GP-SVM.

unsupervised feature extraction framework based on WGAN-GP and transfer learning is designed for hyperspectral classification. The generator and the discriminator in 3D-WGAN-GP are built with 3D deconvolution operation and 3D convolution operation, respectively, which can better mine the spectral-spatial features of HSIs. The optimized discriminator is transferred and utilized as an unsupervised feature extractor to help solve the problem of insufficient labeled samples. Experimental results prove that the performance of the proposed method is better than that of GAN and 3D-WGAN. In addition, the proposed method provides an alternative way for dimensionality reduction and unsupervised feature extraction of HSIs.



# Chapter 4

## Unsupervised feature extraction based on CAE for hyperspectral classification

### 4.1 Introduction

To get rid of the limitation of labeled samples, unsupervised feature extraction method based on GAN is designed in Chapter 3. However, since GAN is trained in a confrontational manner, the optimization of the GAN is more complicated and more challenging. The AE learns a representation for input data through an encoder and then decodes the representation to reconstruct data [48, 49]. The AE can be optimized by minimizing the error between the reconstructed data and the input data, and no labels are involved, which is a typical unsupervised model. Because of these characteristics of AE, unsupervised feature extraction methods based on AE have been introduced in HSIs and achieved some results [50–52, 112]. Unfortunately, when AE-based models are developed for unsupervised feature extraction, features from the single layer are usually considered, which can lose some useful information [113]. The image pyramid framework, which uses different-scale images to independently train multiple networks to obtain multi-level features is one of the solutions [114], but training multiple networks increases the time and computational cost, which is unsatisfactory.

The encoder of a AE is a hierarchical structure from bottom to top, and it's like a feature pyramid. The bottom layer mainly corresponds to information, such as edges, texture, and contours, and the top layer mainly corresponds to semantic

information [115]. Considering the construction and training of AE is easier than GAN, an unsupervised multi-level feature extraction method based on a 3D-CAE is proposed in this Chapter. The designed 3D-CAE is composed of 3D convolutional layers and 3D deconvolutional layers, combining the advantages of CNN and AE. The 3D-CAE can not only fully mine the spectral-spatial information with 3D data as input, but it also does not require the participation of labeled samples in the training process. Besides, multi-level features are directly obtained from different encoded layers of the optimized encoder, which is more efficient when compared to training multiple networks. The full use of the detail information at the bottom layer and semantic information at the top layer can achieve complementary advantages and improve the classification results.

In addition, the input size for different targets is always same while different targets often perform differently with the same input size, especially when there are small targets. In order to solve this problem and balance different targets, a novel multi-size and multi-model framework based on three-dimensional convolutional autoencoder, called 3D-M<sup>2</sup>CAE, is proposed. Three 3D-CAEs with different input sizes centered on the observed pixel are used to build the framework and extract features. Moreover, in order to save training time, the framework is established and trained in a progressive way with the help of transfer learning [22, 24, 25]. The weights of the middle layers of the latter 3D-CAE are transferred from the former optimized 3D-CAE, which speeds up and facilitates network training. Benefiting from this training method, the features of the same target from different sizes are obtained in a more efficient way.

## 4.2 Overview of AE

Traditional AE [116] as shown in Figure 4.1 consists of fully connected layers, and it unusually contains an input layer, a hidden layer and an output layer, which constitute an encoder and a decoder. If there is an input  $\mathbf{I} \in \mathbb{R}^{I_1}$  and it's first mapped to a latent representation  $\mathbf{Y}$  by encoder during the training procedure. Then this representation  $\mathbf{Y}$  can be decoded to a reconstruction one  $\mathbf{O}$ . The output size of AE is the same as its input size. These two steps can be expressed by the formula as:

$$\mathbf{Y} = f(\mathbf{W}\mathbf{I} + \mathbf{b}) \quad (4.1)$$

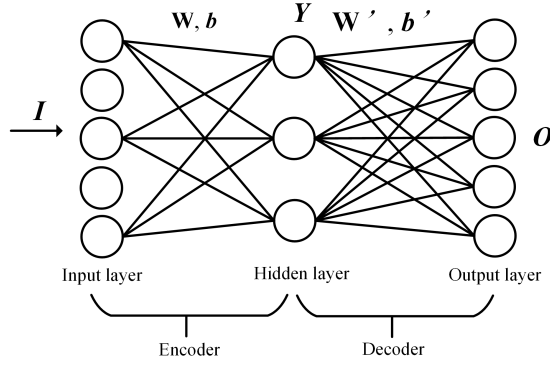


Figure 4.1: Conventional AE architecture.

$$\mathbf{O} = f(\mathbf{W}'\mathbf{Y} + \mathbf{b}') \quad (4.2)$$

where the one-dimensional (1D)  $\mathbf{W}$  of length  $W_1$  ( $\mathbf{W} \in \mathbb{R}^{W_1}$ ) denotes the weight between input layer and hidden layer and  $\mathbf{W}'$  is the weight matrix between hidden layer and output layer.  $\mathbf{b}$  and  $\mathbf{b}'$  represent the bias vectors. The weight matrix can be constrained by  $\mathbf{W}' = \mathbf{W}^T$ , in which case the AE has tied weights [117]. Generally, activation function ( $f(\cdot)$ ) is used to introduce nonlinearity into the model. The parameters of the model are optimized by minimizing the error between the input and reconstruction. Mean squared error (MSE) defined as  $\mathbf{E}(\mathbf{I}, \mathbf{Y}) = \|\mathbf{O} - \mathbf{I}\|^2$  is one of the commonly used loss functions.

Since no labeled data is needed during the whole training process, the training of AE is unsupervised. When the network can recover the input from the latent representation  $\mathbf{Y}$ , we think  $\mathbf{Y}$  preserves useful information and invariant features. The higher the quality of the reconstructed image reflects the better the extracted features. The AE can also be stacked to a deeper network, stacked autoencoder (SAE), with multiple hidden layers for learning more high-level information.

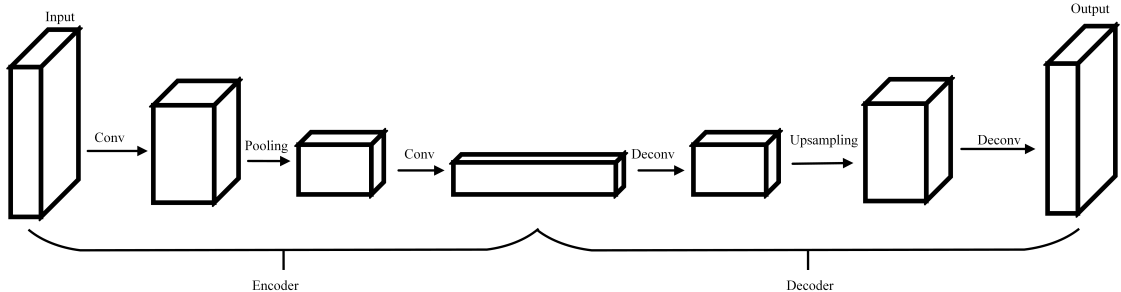


Figure 4.2: CAE architecture.

Traditional AE usually takes the form of a 1D vector as input, which has limi-

tations on retaining the spatial information of the original data. Convolution-based operation can be flexibly performed on tensor data and it has been widely used in image processing. CAEs replacing fully connected layers by convolutional layers, which can not only train the network unsupervised, but also process multi-dimensional data more flexibly. Taking 3D convolution as an example, when the input is  $\mathbf{I}$  with size of  $I_1 \times I_2 \times I_3$  ( $\mathbf{I} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ ), the convolution kernel is  $\mathbf{W}$  with size of  $W_1 \times W_2 \times W_3$  ( $\mathbf{W} \in \mathbb{R}^{W_1 \times W_2 \times W_3}$ ), and the stride is  $1 \times 1 \times 1$ , its output is defined as:

$$\mathbf{O}^{x, y, z} = \sum_{p=0}^{W_1-1} \sum_{q=0}^{W_2-1} \sum_{r=0}^{W_3-1} \mathbf{W}^{p, q, r} \mathbf{I}^{x+p, y+q, z+r} + \mathbf{b} \quad (4.3)$$

where  $\mathbf{O}^{x, y, z}$  means the output at position  $(x, y, z)$ ,  $\mathbf{W}^{p, q, r}$  denotes the kernel value of position  $(p, q, r)$ , and  $\mathbf{I}^{x+p, y+q, z+r}$  represents the input value at position  $(x+p, y+q, z+r)$ . Then, the MSE value can be calculated by Eq. (4.4).

$$\mathbf{E}(\mathbf{I}, \mathbf{Y}) = \frac{1}{I_1 \times I_2 \times I_3} \sum_{x=0}^{I_1-1} \sum_{y=0}^{I_2-1} \sum_{z=0}^{I_3-1} (\mathbf{I}^{x, y, z} - \mathbf{O}^{x, y, z})^2 \quad (4.4)$$

## 4.3 Proposed multi-level feature extraction method

### 4.3.1 Details of the proposed framework

Considering convolution-based operation has high flexibility in processing multi-dimensional data and has a strong ability in feature extraction, a 3D-CAE is introduced to extract features unsupervised. In order to better preserve the spatial and spectral characteristics of HSIs, the designed 3D-CAE is established by fully 3D convolutional layers and 3D deconvolutional layers (see Figure 4.3), where Conv- $n$  and Deconv- $n$  mean the  $n$ th convolutional layer and the  $n$ th deconvolutional layer, respectively. As in the previous two chapters, a 3D block centered on the current observed pixel is used as the input of 3D-CAE to learn its invariant characteristics. The proposed framework based on 3D-CAE for multi-level feature learning is mainly divided into three steps:

Firstly, a 3D-CAE is constructed. The 3D-CAE is designed as a symmetrical structure composed of 3D convolutional layers and deconvolutional layers, as shown in Figure 4.3. The size of feature map is gradually reduced, and the number of convolution kernels is gradually increased. The size of output is the same as the size of input.

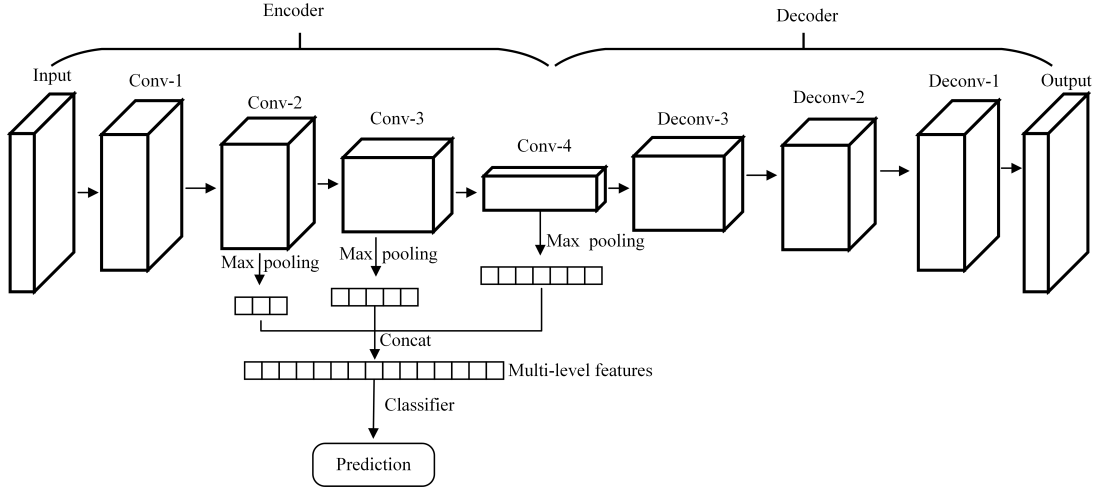


Figure 4.3: Proposed framework for multi-level feature extraction.

Secondly, 3D-CAE is trained and optimized. The data is input into the 3D-CAE and encoded as a low-dimensional representation through the encoder. The decoder is responsible for recovering the original input data from the representation. The 3D-CAE is constantly adjusted by minimizing the error between the output ( $\mathbf{O}^{x, y, z}$ ) and input ( $\mathbf{I}^{x, y, z}$ ), as described in Eq. (4.4). When the network can reconstruct the input data well, we believe that the 3D-CAE has a strong ability to mine the useful information in the data.

Thirdly, multi-level features from the optimized encoder are obtained. The hierarchical structure of the encoder from the bottom to top provides us with features of different levels and different scales. Max-pooling is introduced to reduce the feature dimension and increase feature invariance [118]. The filter size of max-pooling is set to equal to the size of the corresponding feature map. Through pooling operations, each layer can get a feature vector containing different information. The final features are concatenated by these feature vectors from multiple layers of encoder to make them contain more information and have high scale robustness. It is worth noting that the proposed multi-level features come from a single network. Compared with training multiple networks to obtain multi-level features, the proposed method is more effective and greatly saving training time. The goal of the proposed method is to make full use of the well-trained network to obtain as much information as possible, and then help to improve the subsequent classification accuracy.



## 4.3.2 Experimental results

In order to compare and study the performance of the proposed feature extraction method, experiments are performed on Pavia University and Indian Pines. Based on the experimental results of the relationship between band and accuracy in Chapter 2, the bands of the two data sets are reduced to 10 by PCA in order to reduce the amount of calculation and improve the efficiency of network training [75, 119].

### 4.3.2.1 Network Construction

For each pixel in HSIs, a 3D block with a size of  $S \times S \times H_{dr}$  centered on the observed pixel is selected as the input to construct the network, where  $S \times S$  represents the spatial neighborhood window around the observed pixel and  $H_{dr}$  means the dimension after dimensionality reduction. Taking  $13 \times 13 \times 10$  as an example, the corresponding 3D-CAE structure is given in Table 4.1. Considering that the established 3D-CAE is symmetrical, only the parameter settings of the encoder are listed.

Table 4.1: Network structures of encoder in proposed 3D-CAE.

Layer	Input Size	Kernel	Output
Conv-1	$13 \times 13 \times 10 \times 1$	$5 \times 5 \times 4 \times 16$	$9 \times 9 \times 7 \times 16$
Conv-2	$9 \times 9 \times 7 \times 16$	$5 \times 5 \times 3 \times 32$	$5 \times 5 \times 5 \times 32$
Conv-3	$5 \times 5 \times 5 \times 32$	$3 \times 3 \times 3 \times 64$	$3 \times 3 \times 3 \times 64$
Conv-4	$3 \times 3 \times 3 \times 64$	$3 \times 3 \times 3 \times 128$	$1 \times 1 \times 1 \times 128$

In Table 4.1, Conv- $n$  represents the  $n$ th convolutional layer and kernel of  $k_1 \times k_2 \times k_3 \times k_4$  means that there are  $k_4$  convolution kernels with kernel size being  $k_1 \times k_2 \times k_3$  in the current layer. Besides, the stride is set to  $1 \times 1 \times 1$  during the convolution operation. ReLU is mainly used as an activation function to introduce nonlinear mapping into the network, except for the last deconvolution layer with sigmoid. Adam [81] is selected as the optimizer to update the weights.

### 4.3.2.2 Comparison and analysis of experimental results

Classification results based on different single-level features are considered for comparison to better evaluate the effectiveness of the multi-level features. The better the classification result, the better the corresponding features. In the experiment, SVM is selected as the classifier. OA, AA, and  $\kappa$  values are introduced to evaluate

the classification results. For each class in data sets, approximately 10% is used to train the classifier and the rest is used for testing.

At first, single-level features and multi-level features from three encoded layers are compared under the condition of input size being  $13 \times 13 \times 10$ . Since the number of encoded layers used to form multi-level features may also affect the classification results, we will study the influence of this parameter on the results later. As shown in Figure 4.4, the feature map size in top three layers (the third, fourth, and fifth layers) of encoder is  $5 \times 5 \times 5$ ,  $3 \times 3 \times 3$ , and  $1 \times 1 \times 1$ , respectively. Therefore, the filter size of max-pooling in the third and fourth layers is correspondingly set as  $5 \times 5 \times 5$  and  $3 \times 3 \times 3$ . The feature map size of the fifth layer is already  $1 \times 1 \times 1$ , so we directly flatten the feature maps into a 1D vector. After max-pooling operation, three feature vectors are obtained with sizes of  $1 \times 32$ ,  $1 \times 64$ , and  $1 \times 128$ . The three feature vectors are concatenated to obtain a final feature vector with the size being  $1 \times 224$ . These features are fed into the classifier, and the prediction results can be obtained, where Prediction I represents the predicted classification results based on the final multi-level features with a size of  $1 \times 224$ , Prediction II represents the results of single-level features  $1 \times 128$  from the fifth layer, Prediction III corresponds to the single-level features with a size of  $1 \times 64$ , and Prediction IV corresponds to the single-level features with a size of  $1 \times 32$ .

Tables 4.2 and 4.3 list the classification results that are based on different features of Pavia University and Indian Pines, respectively.

Table 4.2: The classification accuracy of Pavia University based on different features.

Class \ Prediction	Single-Level			Multi-Level
	IV	III	II	I
Asphalt	95.17	96.47	97.68	98.28
Meadows	78.94	89.47	93.14	94.14
Gravel	97.45	98.69	98.47	99.46
Trees	95.98	96.96	97.98	97.75
Metal sheets	99.86	99.98	100.00	100.00
Bare soil	78.43	84.81	88.67	96.60
Bitumen	77.44	79.70	79.92	91.43
Bricks	90.71	93.45	96.06	96.79
Shadows	98.83	99.36	99.78	99.79
OA (%)	92.76	95.11	96.19	98.10
AA (%)	90.33	93.20	94.65	97.14
$\kappa$ (%)	90.33	93.49	94.93	97.48

For the Pavia University data set, it can be observed from Table 4.2 that Predic-

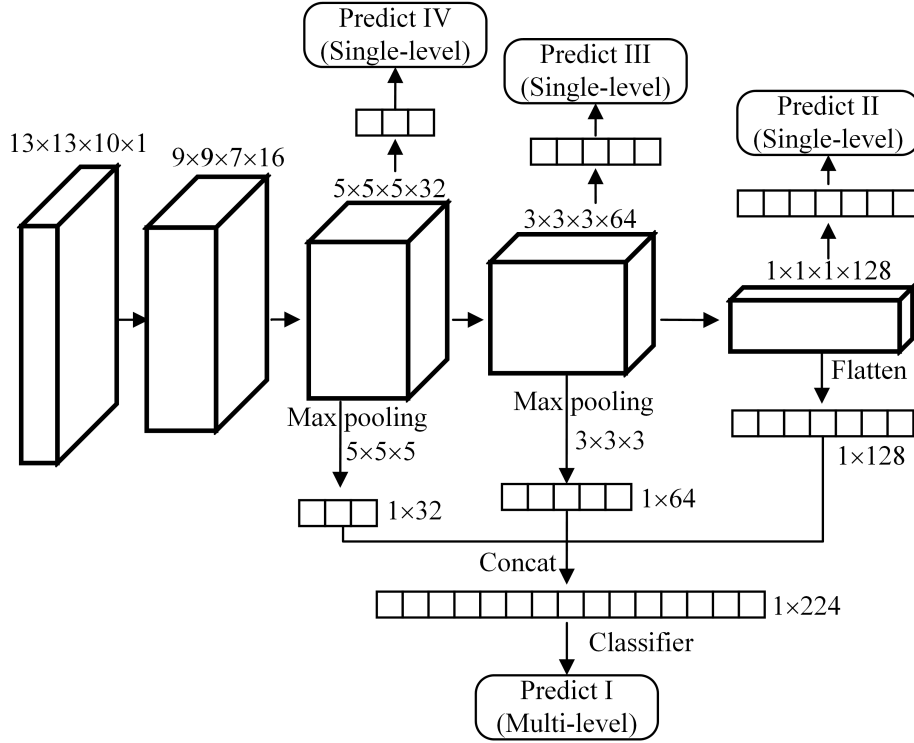


Figure 4.4: Proposed framework for multi-level feature extraction.

tion II based on features from top layer of encoder are better when compared with Prediction III and Prediction IV when only single-level features are considered. The classification accuracy of the Meadows, Bare soil and Bitumen is less than 90% in Prediction III and Prediction IV, which is not satisfactory. Although the relevant results in Prediction II are improved, the classification accuracy of the Bare soil and Bitumen is still not good. When multi-level features are used for classification, the classification accuracy of each category exceeds 90%. Moreover, the results of Prediction I are approximately 2% higher than the OA, AA, and  $\kappa$  values of Prediction II.

For Indian Pines data set, when single-level features are used for classification, it can be found from Table 4.3 that the performance of single-level features of Prediction IV and Prediction III is not as good as Prediction II. Prediction II is the best among classification results based on single-level features, but the classification accuracy of the Grass-pasture-mowed, Oats, Wheat and Buildings-grass-trees is less than 80%. When multi-level features are used for classification, the classification accuracy of the Grass-pasture-mowed, Oats, Wheat and Buildings-grass-trees is increased by 14%, 25%, 7%, and 7% compared with Prediction II. In addition, the highest OA, AA, and  $\kappa$  values are achieved when multi-level features are used. Pre-

Table 4.3: Classification accuracy of Indian Pines based on different features.

Class \ Prediction	Single-Level			Multi-Level
	IV	III	II	I
Alfalfa	80.43	82.61	84.78	89.13
Corn-notill	56.63	70.36	92.05	96.14
Corn-min	58.89	74.58	82.28	87.04
Corn	53.16	72.99	80.17	87.34
Grass-pasture	84.06	95.24	97.31	98.75
Grass-trees	93.84	96.85	97.81	98.90
Grass-pasture-mowed	82.14	75.00	53.57	67.85
Hay-windrowed	97.28	98.54	100.00	100.00
Oats	95.00	90.00	75.00	100.00
Soybean-notill	54.22	75.21	86.93	91.04
Soybean-mintill	76.86	75.89	83.29	88.39
Soybean-clean	85.36	94.63	97.07	97.56
Wheat	54.97	66.61	76.73	83.31
Woods	94.23	98.10	97.94	99.53
Buildings-grass-trees	76.69	83.68	78.76	86.27
Stone-stel-towers	92.47	95.70	96.77	97.85
OA (%)	73.77	81.70	88.17	92.08
AA (%)	77.27	84.12	86.28	91.83
$\kappa$ (%)	69.95	79.16	86.54	90.98

diction I outperforms any result based on single-level features, which proves that multi-level features allow us to obtain more useful information.

It can be seen from Table 4.2 and Table 4.3 that the classification accuracy of some classes is always lower than other classes under different features, such as Bitumen in Pavia University, and Grass-pasture-mowed and Wheat in Indian Pines. From Table 1.1 and Table 1.2 in Chapter 1, we can find that the number of Bitumen in Pavia University is the second smallest, and the number of Grass-pasture-mowed and Wheat in Indian Pines is less than the average. Besides, the within-class variation and inter-class similarity may also reduce the classification accuracy, such as Asphalt and Bitumen, and Wheat and Grass-trees. But in general, the proposed multi-level features obtain the highest OA, AA, and  $\kappa$  values for the two data sets and the classification accuracy of most land-cover classes is improved when compared to the results that are obtained by single-level features.

Both of the results shown in Table 4.2 and Table 4.3 are obtained under the condition that the input size is  $13 \times 13 \times 10$ . When the input size changes from  $13 \times 13 \times 10$  to  $19 \times 19 \times 10$ , the classification accuracy based on single-level features from top encoded layer (Prediction II) and multi-level features (Prediction I) are compared. The comparison results of Pavia University and Indian Pines are depicted in Figure 4.5 and Figure 4.6, respectively.

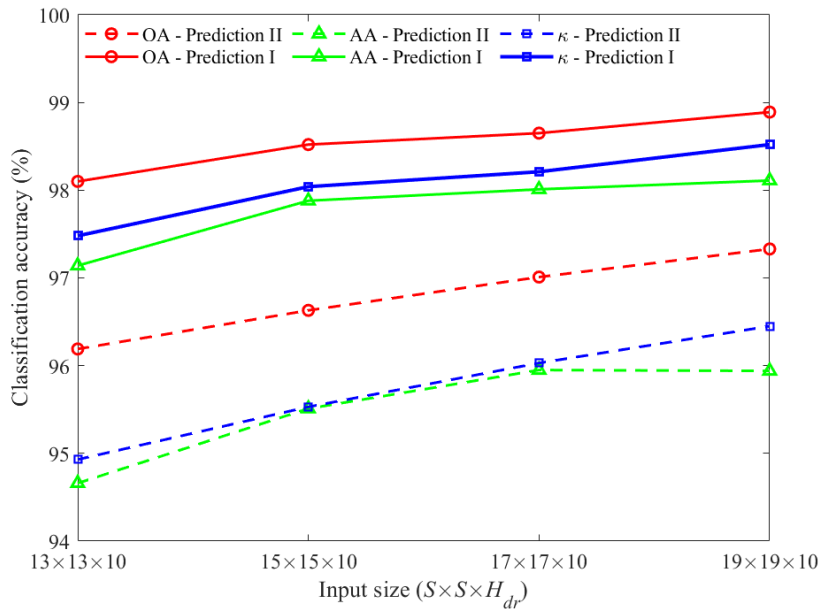


Figure 4.5: Classification accuracy of Pavia University under different input sizes.

For the Pavia University data set, we can find from Figure 4.5 that when the input size increases from  $13 \times 13 \times 10$  to  $19 \times 19 \times 10$ , whether single-level features or

multi-level features are used for classification, the OA, AA, and  $\kappa$  values gradually increase slowly. As the size increases, the amount of calculation also increases, which leads to longer training time. Therefore, we need to comprehensively consider accuracy and efficiency in practical applications. In addition, the performance of multi-level features always outperforms single-level features at any input size. The OA, AA, and  $\kappa$  values increased by about 2% to 3% on average as compared with the results of single features.

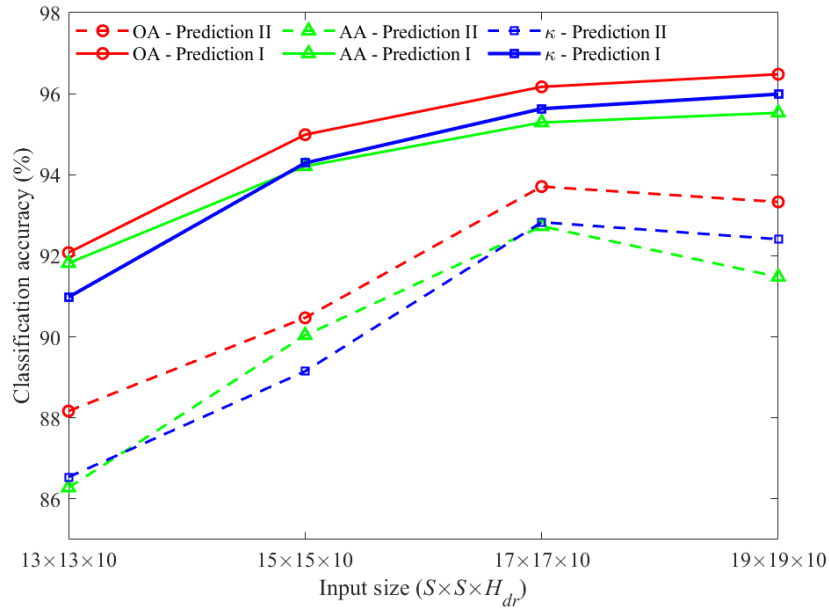


Figure 4.6: Classification accuracy of Indian Pines under different input sizes.

For Indian Pines data set (Figure 4.6), when single-level features are used for classification, we find that the input size greatly affects the classification accuracy. The classification accuracy initially increases as the input size and it reaches a peak at  $17 \times 17 \times 10$ , and then it begins to decline. When multi-level features are used for classification, the classification accuracy shows an upward trend as the input size increases. When the input size is fixed, the performance of multi-level features is much better than single-level features. Compared with the results of single-level features, the classification values of multi-level features improve about 2% to 5%. Moreover, even the peak value of a single-level features is about 2% lower than that of multi-level features.

In general, the results that are based on multi-level features are better than those of single-level features for both data sets, which proves that comprehensive consideration of the feature information of different layers can further improve the results of hyperspectral classification.

In the previous experiments, the multi-level features are obtained by concatenating the information of three encoded layers. In order to observe the impact of the number of encoded layers on the classification results, the multi-level features obtained from two, three, and four encoded layers are compared with input size being  $17 \times 17 \times 10$ . Figure 4.7 shows the comparison results of Pavia University and Figure 4.8 is the comparison results of Indian Pines.

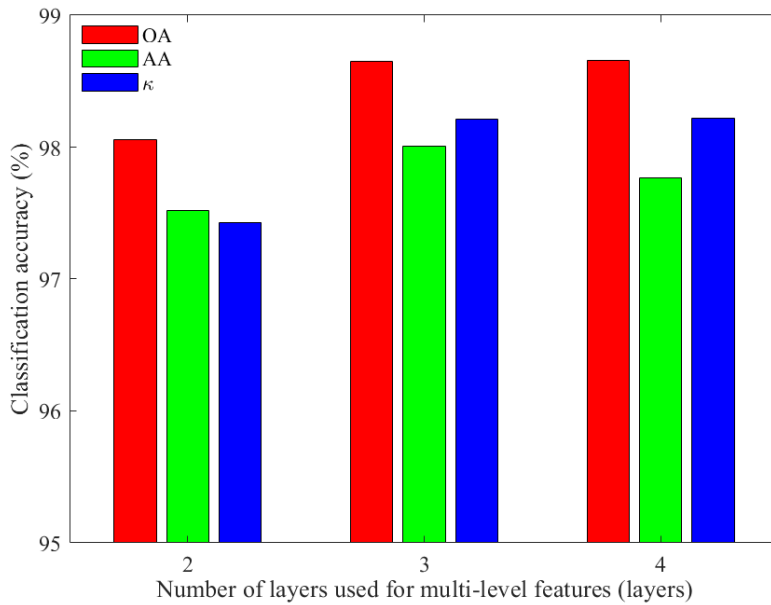


Figure 4.7: Classification accuracy of Pavia University based on multi-level features with different numbers of encoded layers.

It can be observed from Figure 4.7 that the performance of multi-level features obtained by using three and four encoded layers are better than that of two encoded layers. Considering that the results of three and four encoded layers are similar and the feature dimension obtained by three encoded layers is lower, three encoded layers used to concatenate features are more appropriate for Pavia University. Therefore, three is selected as the number of encoded layers for multi-level features in the subsequent experiments.

For the Indian Pines data set (Figure 4.8), the OA and  $\kappa$  values are slightly affected by the number of encoded layers. But the AA values based on two encoded layers and four encoded layers are relatively low. Therefore, three encoded layers are suitable for obtaining multi-level features for Indian Pines.

Next, supervised feature extraction methods based on DBN, 2D-CNN, and unsupervised feature extraction methods based on FA, SAE are considered for comparison to better evaluate the performance of the proposed method with the input

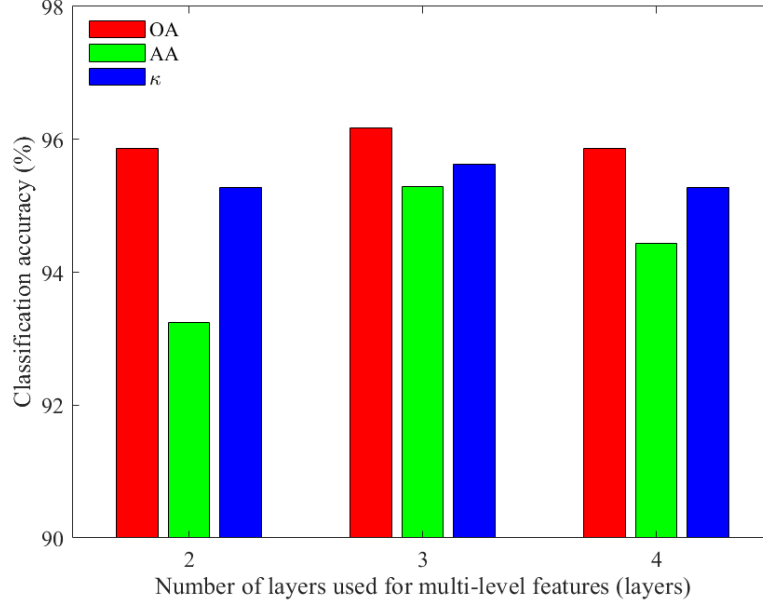


Figure 4.8: Classification accuracy of Indian Pines based on multi-level features with different numbers of encoded layers.

size being  $17 \times 17 \times 10$  and the number of encoded layers for multi-level features being three. FA, DBN, and 2D-CNN have been introduced in Chapter 1. SAE is stacked by multiple AEs that can be used to learn a higher-level representation of input data [120, 121]. The relevant results of Pavia University and Indian Pines under different methods are given in Table 4.4 and Table 4.5, where FE represents feature extraction.

For the Pavia University data set, we can see from Table 4.4 that the OA, AA, and  $\kappa$  values of FA are the lowest, which reflects that deep learning models have more strong ability in feature extraction. When DBN and SAE are used for extracting features, the classification accuracy of Class 1 (Asphalt), Class 3 (Gravel) to 5 (Metal sheets) and Class 9 (Shadows) is relatively high. When 2D-CNN is introduced to obtain features, although the classification accuracy of Class 1 (Asphalt), Class 4 (Trees), and Class 9 (Shadows) is not as good as that of DBN and SAE, the accuracy of most other classes is improved, especially the OA values. This is because the inputs of DBN and SAE are 1D vectors, while 2D-CNN can take 2D matrices as input, which can better retain the spatial information of the target. Among all of the deep models considered, the results based on 3D-CAE are more satisfactory. Besides, compared with single-level features, multi-level features can help us to further improve the classification accuracy. Especially for Class 7 (Bitumen), the accuracy obtained by other feature extraction methods is less than



Table 4.4: Classification accuracy of Pavia University based on different feature extraction methods.

Method Class No.	Supervised FE			Unsupervised FE		
	DBN	2D-CNN	FA	SAE	3D-CAE Single-Level	3D-CAE Multi-Level
1	95.85	96.74	95.88	96.26	97.48	98.58
2	75.51	73.03	79.56	73.70	93.14	94.76
3	97.88	99.63	86.47	97.55	98.76	99.68
4	96.87	96.80	95.14	95.07	98.07	97.78
5	99.78	99.14	99.03	100.00	100.00	100.00
6	76.60	88.03	94.55	66.91	92.32	97.71
7	72.93	89.02	81.65	82.78	86.99	95.49
8	95.11	89.19	69.61	90.82	96.77	98.07
9	99.79	96.72	95.88	97.88	100.00	100.00
OA (%)	92.97	95.03	88.16	91.45	97.01	98.65
AA (%)	90.03	92.13	88.64	89.00	95.94	98.01
$\kappa$ (%)	90.60	93.36	84.62	88.50	96.03	98.21

90%, but the introduction of multi-level features reaches 95%. Overall, the highest OA, AA, and  $\kappa$  values are obtained by the proposed multi-level features.

For Indian Pines data set (Table 4.5), the classification results of FA are not good and the classification accuracy of most classes is less than 90%. DBN and SAE help us to improve the classification accuracy to a certain extent, but it's still not satisfactory. The OA and  $\kappa$  values based on 2D-CNN and CAE-based models exceed 90%, which demonstrates that convolution-based operations are more flexible and have strong feature extraction capabilities. Besides, the OA, AA, and  $\kappa$  values that are based on multi-level features improved by about 3%, 1%, and 3% compared with single-level features.

For better visual comparison, classification maps of Pavia University and Indian Pines obtained by different methods are depicted in Figure 4.9 and Figure 4.10, respectively.

For the Pavia University data set, it can be seen that there are many pixels in the green area that are incorrectly classified into the yellow, and some pixels in the sienna region are misclassified into the red in Figure 4.9 (c) - (e). Besides, the misclassified pixels in the green and sienna region are greatly reduced in Figure 4.9 (f) and (g), but some pixels in the purple region are still not correctly classified, especially in Figure 4.9 (e). Overall, the classification map in Figure 4.9 (h) is the clearest.

Table 4.5: Classification accuracy of Indian Pines based on different feature extraction methods.

Class No.	Method	Supervised FE			Unsupervised FE		
		DBN	2D-CNN	FA	SAE	3D-CAE Single-Level	3D-CAE Multi-Level
1		89.13	84.78	89.13	65.22	84.78	91.30
2		92.77	82.49	61.81	86.14	93.49	94.61
3		92.36	91.32	61.90	84.59	91.25	96.98
4		87.76	69.62	43.88	83.12	91.14	94.93
5		75.77	92.96	87.78	83.85	97.10	97.51
6		92.33	97.676	81.51	95.21	99.17	99.45
7		92.86	64.28	89.29	50.00	75.00	85.71
8		98.12	98.53	93.10	94.35	99.58	100.00
9		90.00	85.00	65.00	65.00	95.00	100.00
10		77.77	87.97	53.60	88.37	88.78	94.15
11		81.02	94.70	88.96	93.60	92.67	95.47
12		98.54	83.02	99.99	88.29	94.63	91.39
13		85.67	98.53	34.23	71.50	90.05	90.73
14		98.74	95.65	95.65	92.89	98.33	99.92
15		95.34	86.15	70.47	74.35	92.75	96.89
16		46.23	79.56	68.82	55.91	99.97	95.69
OA (%)		87.87	90.88	75.16	87.85	93.71	96.17
AA (%)		87.15	87.03	74.07	79.53	92.73	95.29
$\kappa$ (%)		86.24	89.59	71.16	86.12	92.83	95.63

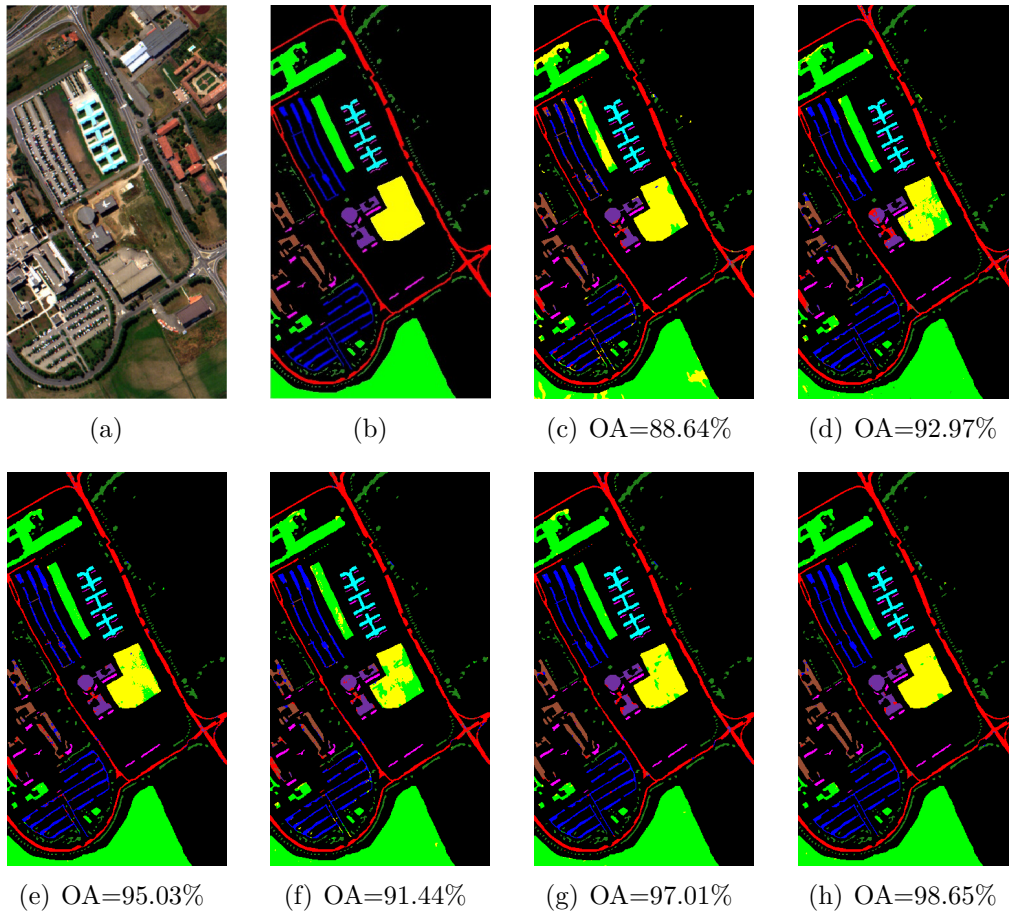


Figure 4.9: Pavia University: (a) Composite image, (b) Ground truth, (c) FA, (d) DBN, (e) 2D-CNN, (f) SAE, (g) 3D-CAE (single-level features), and (h) 3D-CAE (multi-level features).

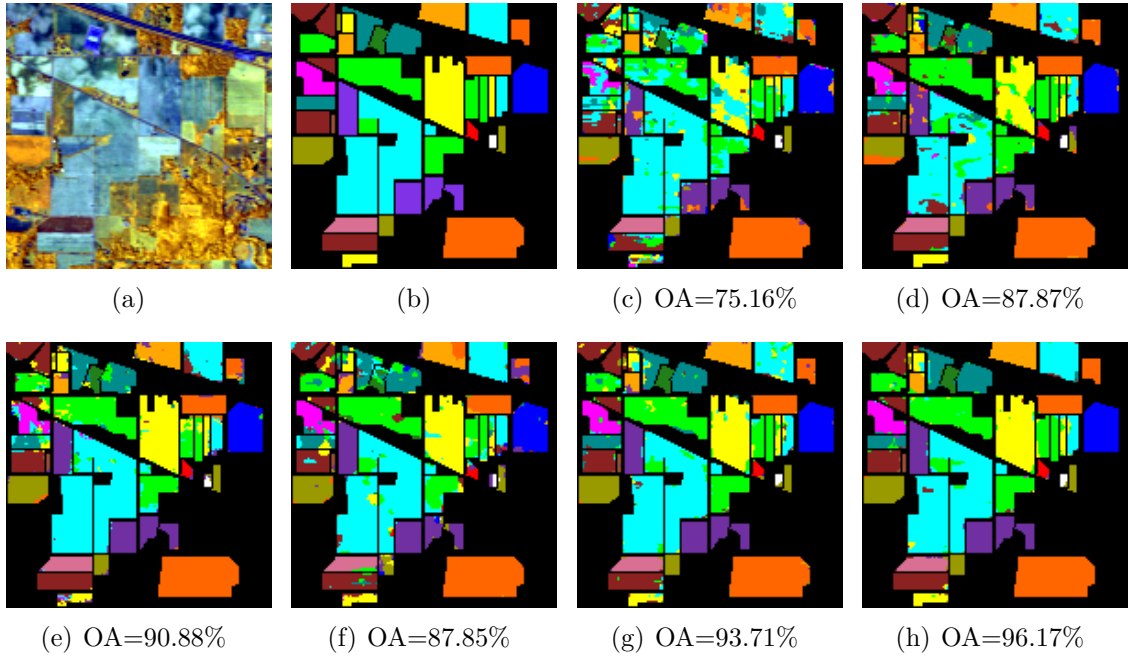


Figure 4.10: Indian Pines: (a) Composite image, (b) Ground truth, (c) FA, (d) DBN, (e) 2D-CNN, (f) SAE, (g) 3D-CAE (single-level features), and (h) 3D-CAE (multi-level features).

For the Indian Pines data set, there are many misclassified pixels in Figure 4.10 (c), (d) and (f), especially the upper left corner area. The classification maps in Figure 4.10 (e) and (g) are better. Among all of the classification maps, Figure 4.10 (h) has the least number of misclassified pixels, which demonstrates the effectiveness of the proposed method.

#### 4.4 Proposed 3D-M<sup>2</sup>CAE framework for small target feature extraction and classification

In Subsection 4.3, multi-level features from different layers of the same input data are studied. Generally, the input size of different targets in the model is the same. However, the relationship between different targets and input size may be different. Therefore, it's necessary to improve the feature robustness of the target to the input size, especially when there are small targets. In order to achieve this goal, a 3D-M<sup>2</sup>CAE framework with multi-size and multi-model is proposed in this subsection.

#### 4.4.1 Details of proposed framework

In the proposed framework, three data blocks of different input sizes centered on the observed pixel are selected as inputs to obtain features as shown in Figure 4.11.

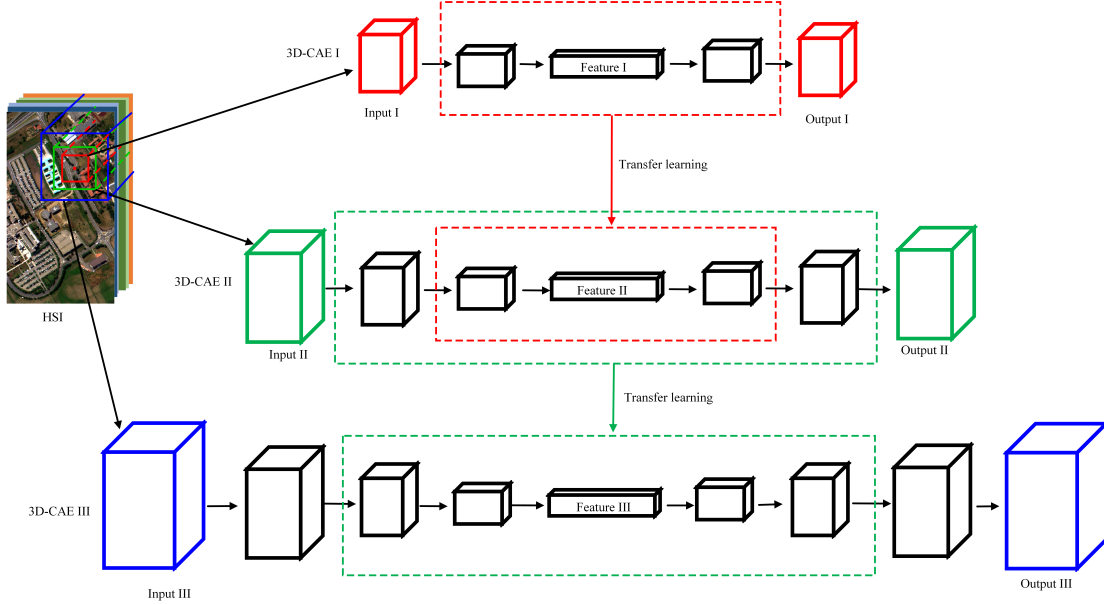


Figure 4.11: Proposed 3D-M<sup>2</sup>CAE framework for unsupervised feature extraction.

The proposed 3D-M<sup>2</sup>CAE mainly contains three 3D-CAEs with different input sizes. It can be found from Figure 4.11 that the size of the input gradually increases and the number of layers of the corresponding network also gradually increases. Since it is time-consuming to train three networks independently, the proposed framework is trained in a progressive way with the help of transfer learning. The training procedure of the proposed 3D-M<sup>2</sup>CAE framework is mainly divided into four steps:

Firstly, a five-layer 3D-CAE with input I is established, referred to as 3D-CAE I, and no labeled samples are involved. When the 3D-CAE I is well optimized, its weights are all appropriate and it has good ability in learning invariant features. The encoded representations (Feature I) retaining useful information to reconstruct images can be obtained.

Secondly, a seven-layer 3D-CAE with input II is established, referred to as 3D-CAE II. The feature map size of second layer in 3D-CAE II is the same as input I and the network structure of the middle three layers in 3D-CAE II is the same as that in 3D-CAE I. It can be seen that input I is contained in input II and closer to the observed pixel. Since input II data has a high similarity to input I, the

weights of middle layers of 3D-CAE II (in the red rectangle) are transferred from optimized 3D-CAE I to reduce training time, and the weights of other layers are initialized randomly. Then the 3D-CAE II is fine-tuned by samples with input II. Due to the weights of middle layers of 3D-CAE II are pre-adjusted, they are closer to the optimal parameters than the random weights. Therefore, the 3D-CAE II takes less number of iterations to reach stability compared with the model where all weights are initialized randomly. When the 3D-CAE II is optimized, Feature II can be obtained.

Thirdly, similar to the previous step, a nine-layer 3D-CAE III with input III is established. The weight of middle layers in 3D-CAE III are transferred from optimized 3D-CAE II and others are initialized randomly. The 3D-CAE III can be further fine-tuned by samples with input III to obtain high-quality features.

Finally, through this progressive growing training, three feature vectors are obtained from different input data. Comprehensive consideration of these characteristics can help us better analyze the target.

Compared with training multiple 3D-CAEs separately to obtain features, the proposed framework is trained in a progressive way with the help of transfer learning, which is more efficient and greatly saves training time. Besides, benefiting from progressive training, the quality of the reconstructed image can be also improved [122], especially when large-size or high-dimensional images need to be reconstructed.

## 4.4.2 Experimental results

### 4.4.2.1 Data set description

In order to verify the performance of the proposed method, two data sets are used as our target data. The first data set named HYDICE is collected by Hyperspectral Digital Imagery Collection Experiment (HYDICE). The original image has  $1280 \times 320$  pixels with 220 spectral bands. Since it's not convenient and efficient to directly process the entire image, 148 spectral bands are retained after removing low-information and noise bands [123]. In addition, we select part of the image that contains the target of interest as the new data set (Figure 4.12), which is denoted as  $HSI_a$ . The second data is Pavia University, which is described in detail in Chapter 1. Both two data sets are normalized to  $[0, 1]$  in the experiment.

$HSI_a$  data set contains  $316 \times 216$  pixels covering 148 bands, which divided into 7 land-cover classes. In the subsequent experiment, 10% samples of each class are used for training the classifier and the remaining samples for testing. The details of

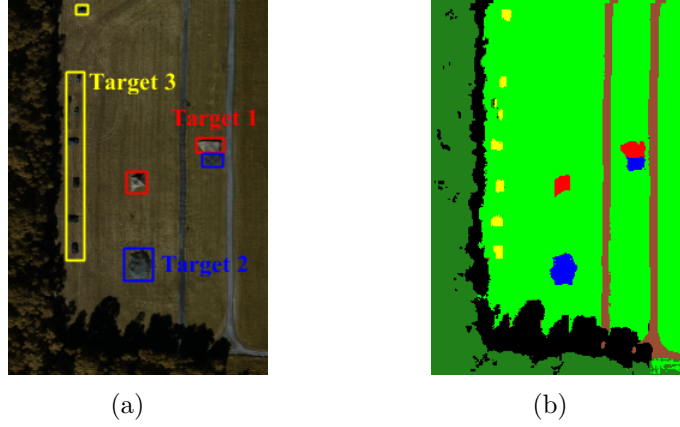


Figure 4.12:  $HSI_a$ : (a) False-color image. (b) Ground truth.

land-cover classes in  $HSI_a$  are listed in Table 4.6.

Table 4.6: Land-cover classes and color coding in  $HSI_a$ .

Class No.	Class	Total	Training	Testing
1	Field	41832	4183	37649
2	Trees	13468	1347	Road
3	Road	4212	421	3791
4	Shadow	7277	728	6549
5	Target 1	414	41	373
6	Target 2	594	60	534
7	Target 3	459	46	413

#### 4.4.2.2 Network construction

The dimensionality of spectral dimension of  $HSI_a$  is reduced by PCA to reduce the amount of calculation. The three 3D-CAEs are designed into a symmetrical structure in the experiment as shown in Figure 4.11. Taking the input size of input I is  $S \times S \times H_{dr}$ , then the size of input II is  $(S + 4) \times (S + 4) \times (H_{dr} + 2)$ , and  $(S + 8) \times (S + 8) \times (H_{dr} + 4)$  for input III. When  $S \times S \times H_{dr}$  is  $7 \times 7 \times 7$ , the corresponding structures of the three encoders are listed in Table 4.7, where  $C_i \times j$  means there are  $j$  kernels of size  $k_1 \times k_2 \times k_3$  in the  $i$ th convolutional layer.

When three 3D-CAEs are constructed, the stride is set to  $1 \times 1 \times 1$ . It can be found by calculation that the input size of 3D-CAE I is the same as the size of feature map of the first convolutional layer in the 3D-CAE II, and the input size of 3D-CAE II is the same as the size of feature map of the first convolutional layer in the 3D-CAE III. Taking the input size of  $11 \times 11 \times 9$  in the 3D-CAE II as an

Table 4.7: Encoder structures of three 3D-CAEs in proposed 3D-M<sup>2</sup>CAE framework.

Network	Input size	C1×8	C2×16	C3×32	C4×64	C5×128
3D-CAE I	$7 \times 7 \times 7$	–	–	$3 \times 3 \times 3$	$3 \times 3 \times 3$	$3 \times 3 \times 3$
3D-CAE II	$11 \times 11 \times 9$	–	$5 \times 5 \times 3$	$3 \times 3 \times 3$	$3 \times 3 \times 3$	$3 \times 3 \times 3$
3D-CAE III	$15 \times 15 \times 11$	$5 \times 5 \times 3$	$5 \times 5 \times 3$	$3 \times 3 \times 3$	$3 \times 3 \times 3$	$3 \times 3 \times 3$

example, after convolving input II with the kernel of  $5 \times 5 \times 3$ , the feature map size is  $(11 - 5 + 1) \times (11 - 5 + 1) \times (9 - 3 + 1) = 7 \times 7 \times 7$ , which is the same as the input size of the 3D-CAE I. Besides, ReLU is utilized to introduce nonlinearity in all convolutional layers and deconvolutional layers except the last layer that uses sigmoid activation. Batch normalization is introduced to normalize the features and Adam is selected for optimizing the network parameters. After the experimental test, the number of training epochs of the three 3D-CAEs are set to 20, 5 and 5 respectively, with batch size being 512.

Through progressive training, three feature vectors from different input sizes are finally obtained. In order to make full use of the extracted features, these three feature vectors are concatenated into one feature vector for subsequent classification. In order to evaluate the quality of learned features by the proposed method, the classification results based on learned features are used to measure their effectiveness. SVM with linear kernel is selected as the classifier. OA, AA and  $\kappa$  values are mainly used to evaluate the classification results.

#### 4.4.2.3 Result analysis of $HSI_a$ data set

In order to study the effect of the input size on the feature performance, we firstly focus on the results of small targets (Target 1, Target 2 and Target 3 in  $HSI_a$ ), which are more challenging in HSI processing. We gradually increase the input size and the OA values of these three small targets based on features obtained by 3D-CAE I, 3D-CAE II and 3D-CAE III are plotted in Figure 4.13 - Figure 4.15.

We can find from Figure 4.13 that when 3D-CAE I is used for feature extraction, the OA values of Target 1 is slightly affected by input size, the OA values of Target 2 increase as the input size increases and then the OA values decrease, and the OA values of Target 3 show an upward trend with the input size in the range of  $7 \times 7 \times 7$  to  $13 \times 13 \times 7$ . Overall,  $7 \times 7 \times 7$  may be more appropriate for Target 1, while  $9 \times 9 \times 7$  is more appropriate for Target 2 and  $13 \times 13 \times 7$  is more appropriate for Target 3.



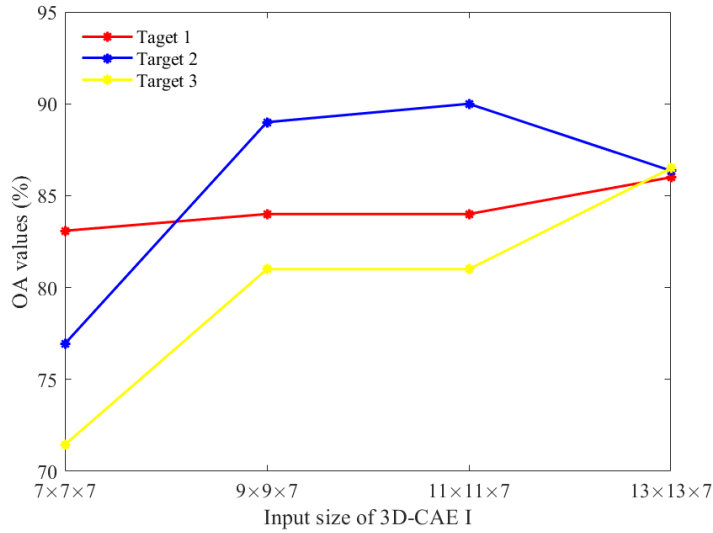


Figure 4.13: OA values of three small targets based on 3D-CAE I under different input sizes.

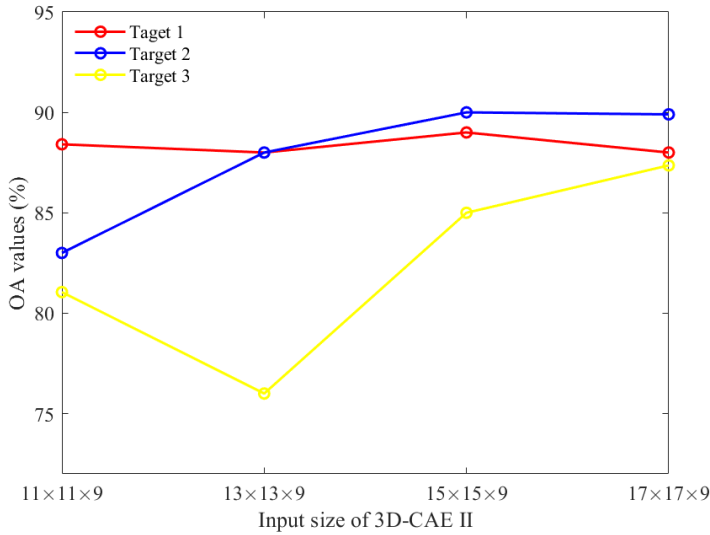


Figure 4.14: OA values of three small targets based on 3D-CAE II under different input sizes.

For 3D-CAE II (Figure 4.14), as the input size increases, the OA values of Target 1 change slightly, the OA values of Target 2 gradually increase and tend to be stable, and the OA values of Target 3 decrease at the beginning and then increase. Overall,  $15 \times 15 \times 9$  helps Target 1 and Target 2 get higher OA values, and  $17 \times 17 \times 9$  helps Target 3 get higher OA value when 3D-CAE II is used for feature extraction.

For 3D-CAE III (Figure 4.15),  $17 \times 17 \times 11$  is a good choice for three small targets and the highest OA values are obtained within the experimental range. Although the OA value is also high for Target 2 when input size is  $21 \times 21 \times 11$ , the amount

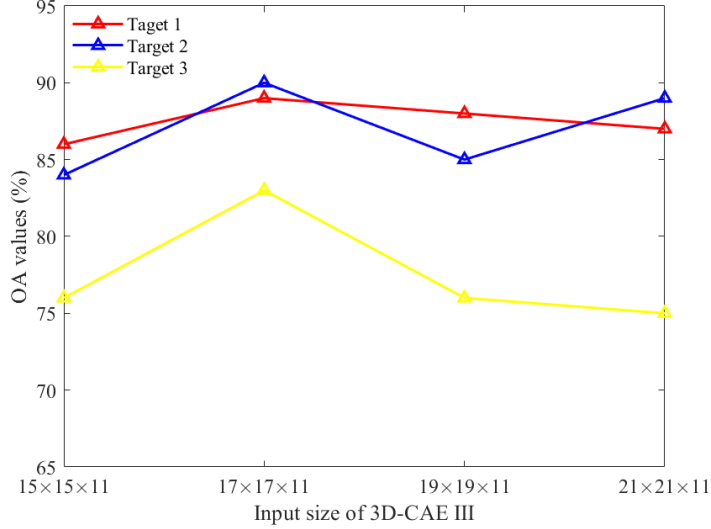


Figure 4.15: OA values of three small targets based on 3D-CAE III under different input sizes.

of calculation increases with the size.

In general, no matter which 3D-CAE model is used to extract features, the OA values are affected by input size and the relationship between different targets and input size is different. In addition, the OA values of Target 3 with input size being  $13 \times 13 \times 7$  (Figure 4.13) and  $13 \times 13 \times 9$  (Figure 4.14) are quite different, which implies that not only the width and height of the input, but also the depth can affect the classification accuracy.

In order to verify the performance of the proposed method, we compare and analyze the feature performance extracted by three single 3D-CAEs and 3D-M<sup>2</sup>CAE framework. The comparison results of three small targets are shown in Figure 4.16 - Figure 4.18, respectively, and the given input size  $H \times H \times H_{dr}$  is based on 3D-CAE I. The input size of 3D-CAE II and 3D-CAE III can be calculated according to the given size.

For Target 1, it can be seen from Figure 4.16 that when 3D-CAE I is used for feature extraction, the input size of  $13 \times 13 \times 7$  helps us obtain the highest OA value. When 3D-CAE II is used for feature extraction, the OA values are stable, but the amount of calculation increases as the input size increases. When 3D-CAE III is used for feature extraction, the highest OA value is got under the condition that the corresponding  $H \times H \times H_{dr}$  is  $9 \times 9 \times 7$ . The performance of 3D-CAE II and 3D-CAE III is better and more stable compared with 3D-CAE I for Target 1. When the proposed 3D-M<sup>2</sup>CAE is used to obtain features, the OA values of 3D-M<sup>2</sup>CAE always exceed that of three 3D-CAEs regardless of input size, and the highest OA

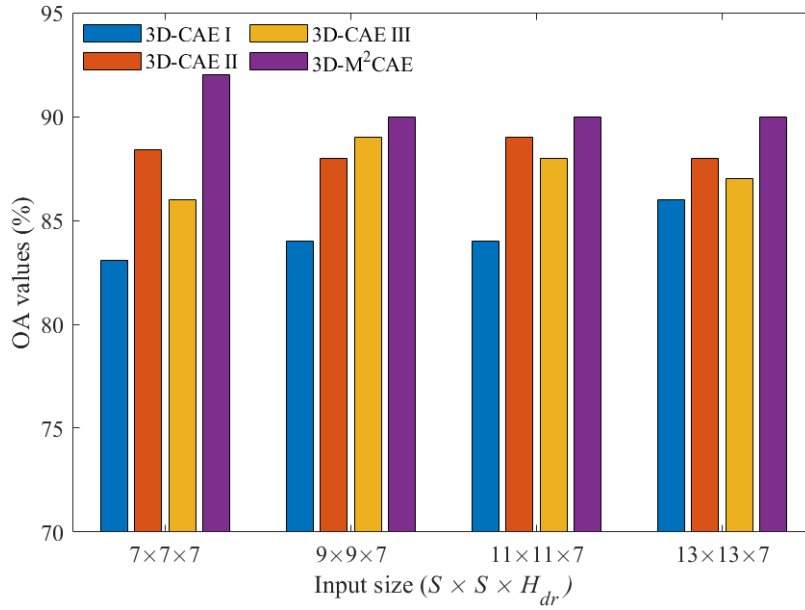


Figure 4.16: OA values of Target 1 based on features obtained from different networks.

value of Target 1 is obtained when the corresponding  $H \times H \times H_{dr}$  is  $7 \times 7 \times 7$ .

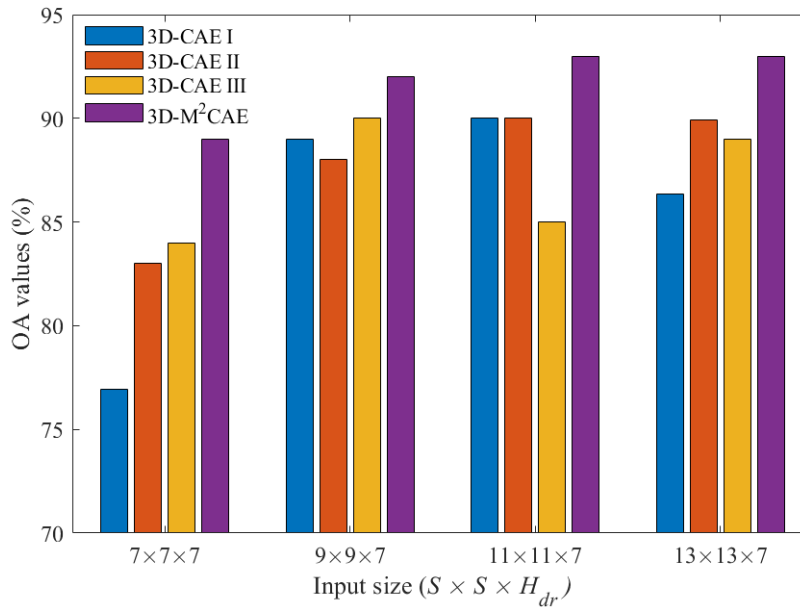


Figure 4.17: OA values of Target 2 based on features obtained from different networks.

For Target 2, we can find from Figure 4.17 that the OA values based on 3D-CAE I and 3D-CAE II rise to the peak and then decreases. When the corresponding  $H \times H \times H_{dr}$  being  $11 \times 11 \times 7$ , the OA values based on 3D-CAE I and 3D-CAE II are higher, while  $9 \times 9 \times 7$  helps 3D-CAE III get the highest OA value. When

proposed 3D-M<sup>2</sup>CAE is used for feature extraction, we can find the OA values are greatly improved compared with single 3D-CAEs, especially when the corresponding  $H \times H \times H_{dr}$  is small.

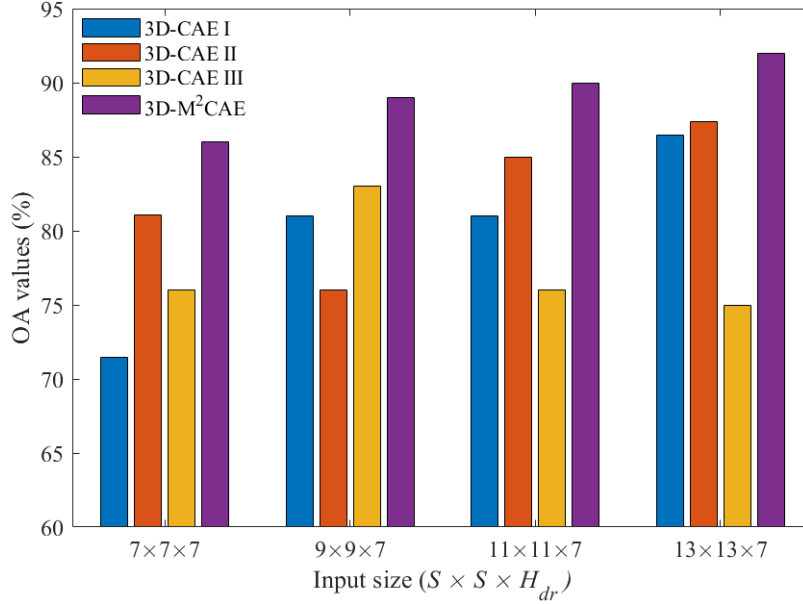


Figure 4.18: OA values of Target 3 based on features obtained from different networks.

For Target 3, we can see from Figure 4.18 that when 3D-CAE I is used for feature extraction, the OA values have an upward trend as the input size increases. For 3D-CAE II, the OA value decreases first and then increases with the input size. 3D-CAE III get the highest OA value with the corresponding  $H \times H \times H_{dr}$  being  $9 \times 9 \times 7$ , but the classification accuracy of other input sizes is low. The performance of proposed 3D-M<sup>2</sup>CAE outperforms the other three 3D-CAEs and the OA values of Target 3 all exceed 85%.

In general, when single 3D-CAEs is used to obtain features, the feature performance of different targets respond differently to the input size. The input size has a great influence on the classification results. However, the proposed 3D-M<sup>2</sup>CAE can help us obtain better results than any single 3D-CAEs without limitation of input size. Besides, the classification accuracy is also greatly improved, which proves that the proposed method can effectively help small targets improve the classification accuracy.

#### 4.4.2.4 Result analysis of Pavia University data set

In the previous experiments, we mainly focus on small targets. Next, we treat Pavia University as target data to further verify the effectiveness and generalization of the proposed 3D-M<sup>2</sup>CAE framework. The OA, AA and  $\kappa$  values of Pavia University are given in Figure 4.19 - Figure 4.21.

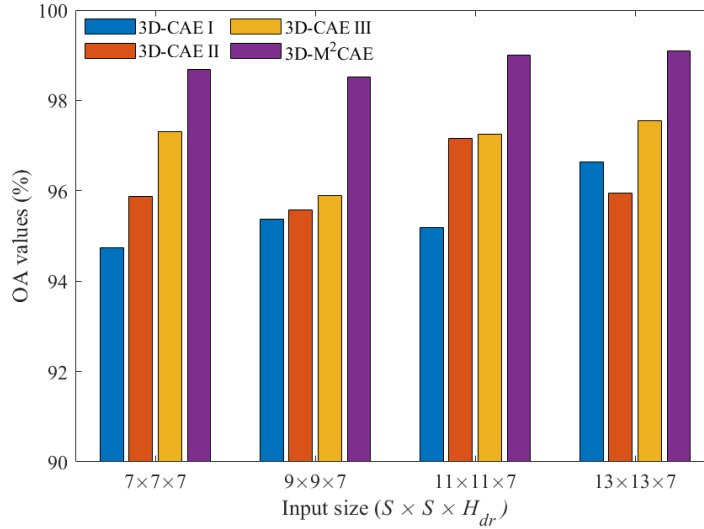


Figure 4.19: OA values of Pavia University based on features obtained from different networks.

It can be observed from Figure 4.19 that 3D-CAE III performs better than 3D-CAE I and 3D-CAE II for Pavia University data set. However, the performance of single 3D-CAE is greatly affected by the input size. When proposed 3D-M<sup>2</sup>CAE is used for feature extraction, the OA values are slightly affected by the input size. Moreover, the OA values obtained based on 3D-M<sup>2</sup>CAE are improved about 2% compared with that got based on 3D-CAE III.

From Figure 4.20, we can find that when single 3D-CAE is used to extract features, the AA values are low especially when the input size is small. The AA value got by proposed 3D-M<sup>2</sup>CAE is about 98%, which is 4% higher than 3D-CAEs I and II, and 2% higher than 3D-CAE III when the corresponding  $H \times H \times H_{dr}$  is  $7 \times 7 \times 7$ . The low AA value reflects the large difference in classification accuracy of different classes. The proposed 3D-M<sup>2</sup>CAE can not only help us further improve the classification accuracy but also narrow the difference between the results of different classes.

Figure 4.21 shows the  $\kappa$  values of different models under different sizes. It can be seen that the  $\kappa$  values of proposed 3D-M<sup>2</sup>CAE far exceed those of 3D-CAEs, which

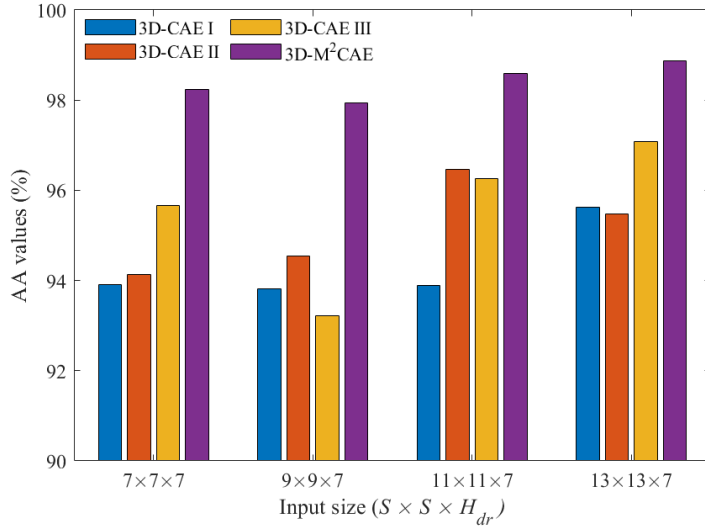


Figure 4.20: AA values of Pavia University based on features obtained from different networks.

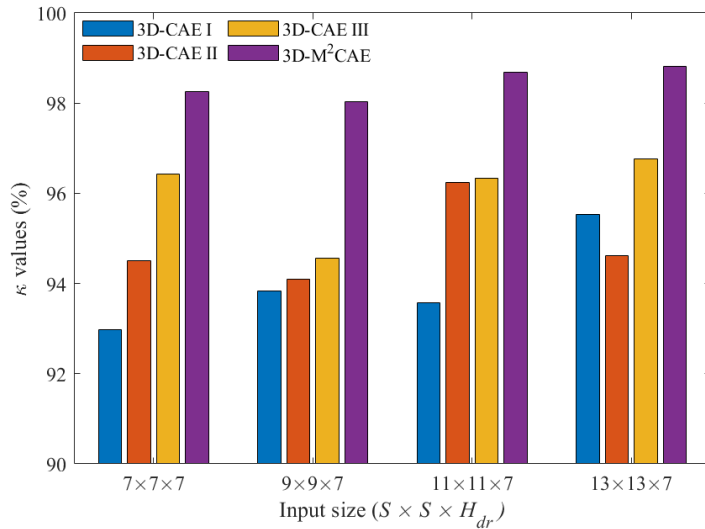


Figure 4.21:  $\kappa$  values of Pavia University based on features obtained from different networks.

demonstrates the potential of the proposed framework.

#### 4.4.2.5 Visual observation and comparison

For better visual observation of the effectiveness of the proposed method, supervised feature extraction methods based on DBN, 2D-CNN, and unsupervised feature extraction method based on FA, SAE are considered for comparison. Considering the amount of calculation and classification accuracy, the input size ( $S \times S \times H_{dr}$ ) in the proposed 3D-M<sup>2</sup>CAE is set to  $9 \times 9 \times 7$  for  $HSI_a$  data set and  $7 \times 7 \times 7$  for Pavia

University data set. The classification maps of these two data sets are depicted in Figure 4.22 and Figure 4.23.

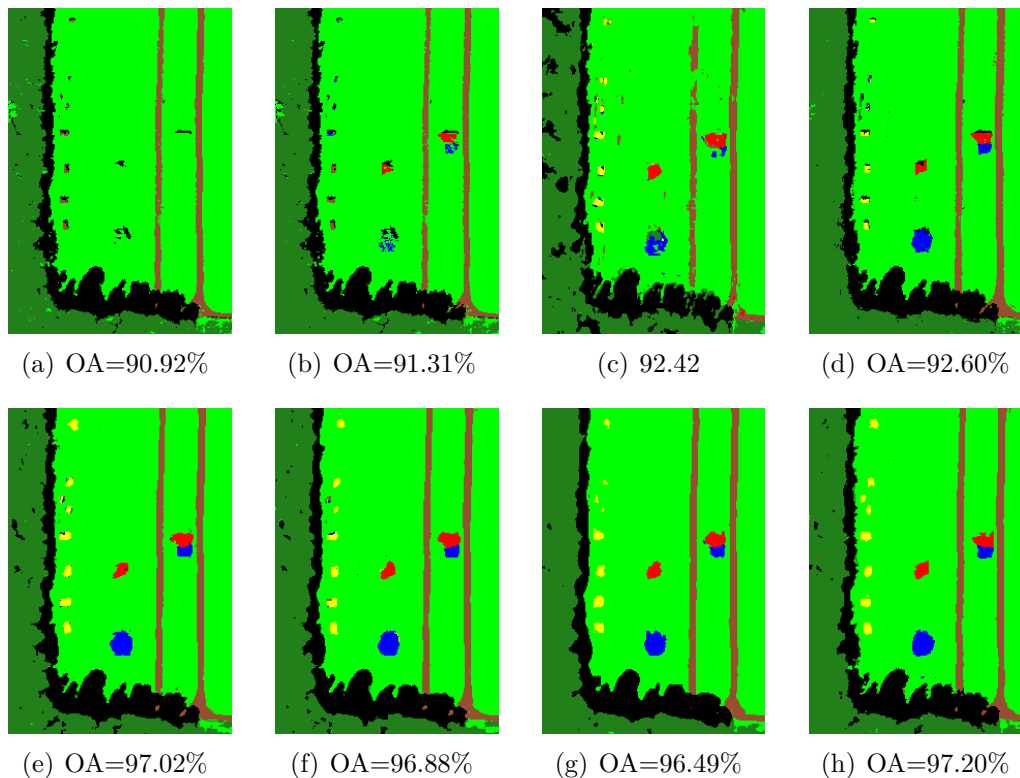


Figure 4.22: Classification maps of  $HSI_a$  obtained by different methods: (a) FA, (b) SAE, (c) DBN, (d) 2D-CNN, (e) 3D-CAE I, (f) 3D-CAE II, (g) 3D-CAE III, (h) Proposed 3D-M<sup>2</sup>CAE.

By comparing the ground truth of  $HSI_a$  in Figure 4.12 (b) and the classification maps in Figure 4.22, we can find that almost all small targets are misclassified in Figure 4.22 (a). The classification results in Figure 4.22 (b) - (d) are improved compared with Figure 4.22 (a), but there are still a lot of yellow pixels are not correctly classified. When 3D-CAE is used to obtain features, the corresponding classification maps are clearer. Although the OA values obtained by single 3D-CAE and 3D-M<sup>2</sup>CAE are not very much different, the classification results of small targets are quite different. This is because the given OA values are for the whole image, and the number of samples for small targets is much smaller than the total number of samples. The classification results of small targets have little effect on the OA value compared with other targets. From the classification map, there are few misclassified pixels in red, blue and yellow region in Figure 4.22 (h), which demonstrates that the proposed method has great potential in feature extraction and classification of small targets.

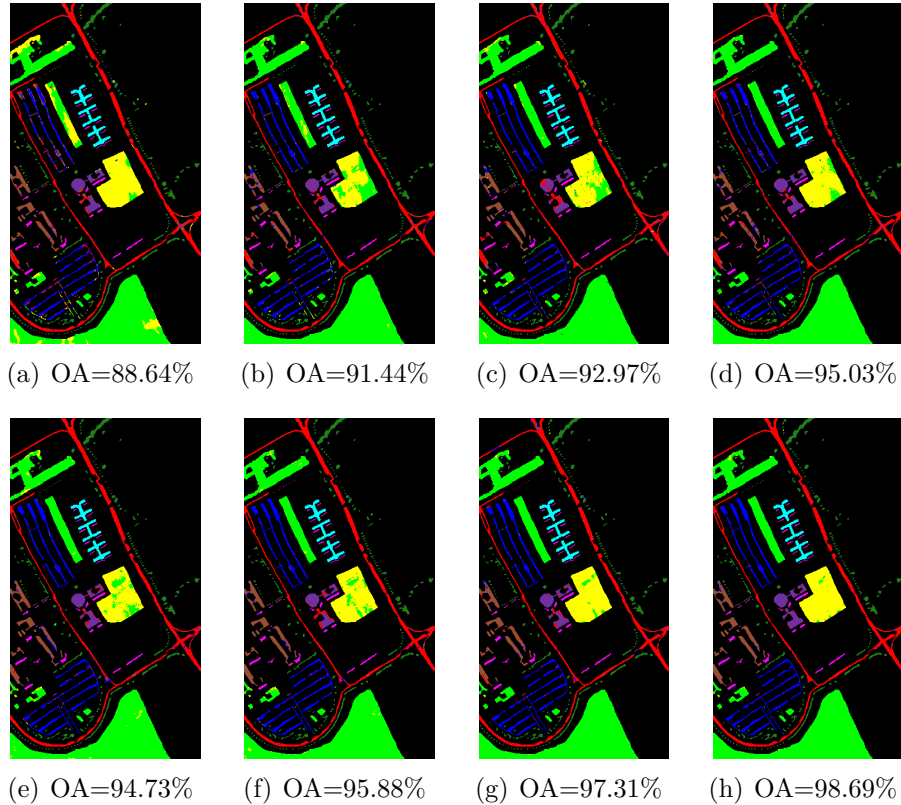


Figure 4.23: Classification maps of Pavia University obtained by different methods: (a) FA, (b) SAE (c) DBN, (d) 2D-CNN, (e) 3D-CAE I, (f) 3D-CAE II, (g) 3D-CAE III, (h) Proposed 3D-M<sup>2</sup>CAE.

It is observed from Figure 4.23 (a)-(f) that there are many yellow pixels are misclassified into green and there are lots of misclassified pixels in purple region. The classification map in Figure 4.23 (g) has less misclassified pixels compared with classification maps in Figure 4.23 (a)-(f), but the purple and yellow areas in the middle are not clear enough. The proposed 3D-M<sup>2</sup>CAE framework help us obtain the highest OA value and clearest classification map.

In general, the proposed method not only helpful for improving the classification of small targets, but also applicable to other targets.

## 4.5 Conclusion

In this chapter, two frameworks based on 3D-CAE are designed to get rid of limitations of labeled samples and further improve classification accuracy. Considering that the convolution-based operations can handle multi-dimensional data flexibly and has a strong ability in feature extraction, the designed 3D-CAEs are stacked



by fully 3D convolutional and 3D deconvolutional layers, which helps us exploit the spectral-spatial characteristics among hyperspectral data.

In the first framework, multi-level features are proposed to contain detail information and semantic information at the same time. The proposed multi-level features are directly obtained from different encoded layers of the optimized encoder, which helps us to make full use of the well-trained network and further improve feature quality. Experimental results of Pavia University and Indian Pines show that single-level features from the top encoded layer perform better when compared to single-level features from other encoded layers. The performance of the proposed multi-level features exceeds any single-level features under different input sizes. The OA, AA, and  $\kappa$  values based on proposed multi-level features increased by about 2% to 3% for Pavia University and 2% to 5% for Indian Pines compared with single-level features from top encoded layer. Besides, we find that the number of layers used to form multi-level features also affects the feature performance. The more encoded layers are selected, the larger the dimension of the multi-level features. Our goal is to use low-dimensional features to obtain high accuracy. Based on the experimental results, we choose three encoded layers for multi-layer features in the experiment. Moreover, the proposed multi-level features are compared with the features obtained by supervised DBN and 2D-CNN, as well as unsupervised FA and SAE. The experimental results show that the proposed method outperforms the considered methods. The proposed multi-level features help us to obtain the highest classification accuracy, which demonstrates that they have huge potential in hyperspectral classification.

In addition, another framework named 3D-M<sup>2</sup>CAE is proposed to balance different targets and improve classification results of small targets. The proposed 3D-M<sup>2</sup>CAE consists of three 3D-CAEs with different input size. The input size and network layers of the three 3D-CAEs are gradually increasing. Since the three inputs of the same target have high relevance and similarity, the weights of the middle layers of the second and third 3D-CAEs are transferred from the previously optimized network. Benefiting from the progressive training methodology and transfer learning, we can facilitate the training and save time of 3D-M<sup>2</sup>CAE. In addition, features from different input sizes can be obtained during the progressive training, which helps us improve the feature robustness to size variations and provide more information to better analyze targets. Since small targets are more sensitive to the input size and have fewer samples, the analysis of small target is more challenging in HSIs. In the experiment, we first focus on small targets to observe the performance

of the proposed 3D-M<sup>2</sup>CAE. Experimental results of  $HSI_a$  show that the proposed method can greatly improve the classification results of small targets compared with single-input model. Then, the experiment is executed on Pavia University data set. The results of Pavia University demonstrate that the proposed framework is not only helpful for improving the classification of small targets, but also applicable to other targets.



# Chapter 5

## Deep learning models for improvement of target detection

### 5.1 Introduction

In the previous chapters, supervised feature extraction and unsupervised feature extraction have been studied for hyperspectral classification. In this chapter, we focus on target detection of HSIs. With a known spectral signature can also be called a spectral template, comparing the spectral template with the pixels in a scene can determine whether target is present or not. There are some detectors which are commonly used for target detection [54], such as adaptive coherence/cosine estimator (ACE), adaptive matched filter (AMF), and spectral angle mapper (SAM). However, HSIs always suffer from spectral variations caused by noise or environment, which enlarges within-class variation and degrades the performance of detectors. It is essential to obtain high detection accuracy even targets in noisy scenes. Thus, we want to improve the target detection results using existing detectors by improving the quality of spectral signature and mining the invariant features of the spectrum. To achieve this goal, denoising is usually done as a preprocessing step for noise removal and then target detection is performed. Traditional denoising methods, such as PCA [55], models based on Wiener filter (WF) [56], and block-matching and 3D filtering (BM3D) [57], have been successfully applied in image processing. However, the traditional denoising methods are easy to face the problem of single task [58] or preserving small targets [59].

With the development of deep learning, some methods based on deep networks have been proposed for image denoising [60]. In [61,62], models based on denoising

autoencoder (DAE) model are established for image denoising, which uses encoder to get the latent representation and then reconstructs it into the clean data through the decoder. In [63], feed-forward denoising convolutional neural network (DnCNN) is designed for image denoising and obtained effective results. In [64], deep residual convolutional neural network (ResCNN) is introduced to learn a non-linear map between noisy and clean image for HSI denoising. In [65], GAN is used for estimating the noise distribution and constructing a paired training dataset to train CNN for image blind denoising. Compared with conventional denoising methods, deep learning-based methods are usually not limited to specific denoising tasks and the parameters are automatically updated according to the input.

In addition, the target of interest in practice is usually small compared to the background (target of non-interest). For example, when aircraft wreckage needs to be detected, its background is usually on challenging terrains, such as mountains, forests or seas [124]. In these cases, the target to be detected can be treated as small target compared to backgrounds. Based on this situation, when small target detection is focused on, we want to segment the HSI to obtain the region of interests (ROIs) and narrow the detection range. In other words, the detection process is divided into two stages. The first stage is selecting ROIs according to the result of segmentation. Some models have been developed for segmentation, such as fully convolutional network (FCN) [125], U-net [126], SegNet [127]. However, segmentation ground truth is usually required when these models are optimized, which is inconsistent with our goal of unsupervised image segmentation. Therefore, we want to find a new way to segment HSIs unsupervised and select ROIs. Then, the second stage can be executed to detect whether the selected ROIs contain targets.

## 5.2 Spectral reconstruction for denoising and target detection

### 5.2.1 Methodology

In this section, we describe some basic knowledge which will be used in the proposed method.

### 5.2.1.1 n mode unfolding

We have known that an HSI can be represented as a 3D block denoted by  $\mathbf{H} \in \mathbb{R}^{H_1 \times H_2 \times H_3}$ , where the HSI has  $H_1$  rows,  $H_2$  columns,  $H_3$  spectral bands and  $\mathbb{R}$  is the real manifold. Then the HSI  $\mathbf{H}$  can be flattened to be a  $n$  mode matrix  $\mathbf{H}_n \in \mathbb{R}^{H_n \times M_n}$  as shown in Figure 5.1, where  $M_n = H_p \times H_q$  ( $p, q \neq n$ ).

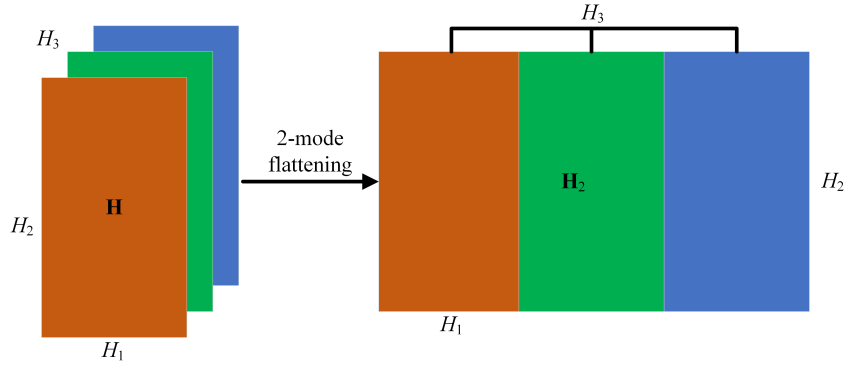


Figure 5.1: A DAE architecture.

### 5.2.1.2 Denoising autoencoder

In Chapter 4, we give a brief introduction to the AE. Compared with AE, DAE tries to reconstruct the original input from a corrupted and partial destroyed one which is usually got by randomly setting some values of input to zero while the others left untouched or adding some noise to input. The architecture of a DAE is shown in Figure 5.2.

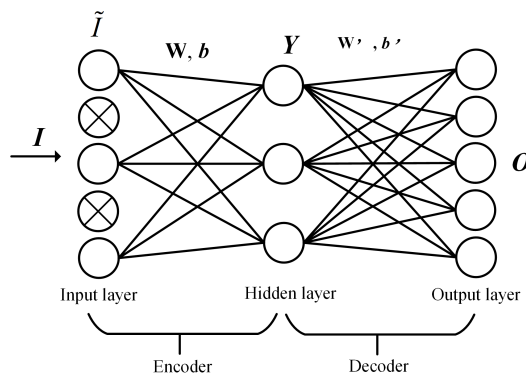


Figure 5.2: A DAE architecture.

It can be seen from Figure 5.2, for each input vector, the value of some neurons is randomly set to zero with a fixed dropout rate. The corrupted input  $\tilde{\mathbf{I}}$  is mapped

to the hidden layer  $\mathbf{Y} = f(\mathbf{W} \tilde{\mathbf{I}} + \mathbf{b})$ , and then  $\mathbf{Y}$  is mapped to the output layer  $\mathbf{O} = f(\mathbf{W}' \mathbf{Y} + \mathbf{b}')$ . During training procedure, the reconstructed  $\mathbf{O}$  from the destroyed  $\tilde{\mathbf{I}}$  is getting closer and closer to  $\mathbf{I}$ . Reconstructing the original version from the corrupted version enforces the robust of the network and can be used to denoise.

### 5.2.2 Target detection combined a multiscale denoising autoencoder

Target detection can be treated as a binary classification task [53]. According to the spectral characteristics, pixels can be classified as target or background. To improve the results of target detection, DAE is introduced to reconstruct spectrums and increase spectral robustness.

### 5.2.3 Spectral reconstruction by denoising autoencoder

Each pixel in HSI can be represented as a 1D vector. If there is a noisy HSI  $\mathbf{H}$  with size of  $H_1 \times H_2 \times H_3$ . According to the 3 mode unfolding, 3D tensor data  $\mathbf{H}$  can be unfolded to be a 2D matrix  $\mathbf{H}_3$  with size of  $H_3 \times M_3$  ( $M_3 = H_1 \times H_2$ ). Each row vector in  $\mathbf{H}_3$  corresponding to a spectral curve can be used as the input of DAE network as shown in Figure 5.3. By minimizing the error between output and input, the DAE is trained and fine-tuned. After the network is well-trained, the reconstructed spectrum that contains the useful information as much as possible can be obtained. Due to the DAE tries to recover the original one from the corrupted one, the spectrum reconstructed by DAE can remove noise while retaining the invariant features.

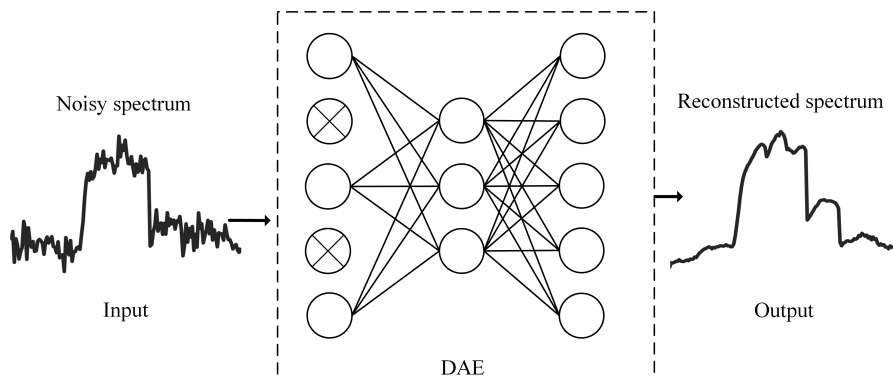


Figure 5.3: Spectral reconstruction with DAE.

### 5.2.4 Proposed model for target detection

The reconstructed spectrums removing noise can replace the original spectrums for target detection. In a DAE network, the dimensions of input layer and output layer are the same, but the dimension of hidden layer can affect the performance of reconstruction which has impact on subsequent target detection. In order to make the reconstructed spectrums contain as much information as possible, a multiscale denoising autoencoder (MSDAE) is designed for improvements of target detection in HSIs. The input spectrum is encoded to different scales to get a set of representations of input, and then they are decoded to fuse into the final reconstructed spectrum. The flowchart of proposed MSDAE model for target detection is depicted in Figure 5.4.

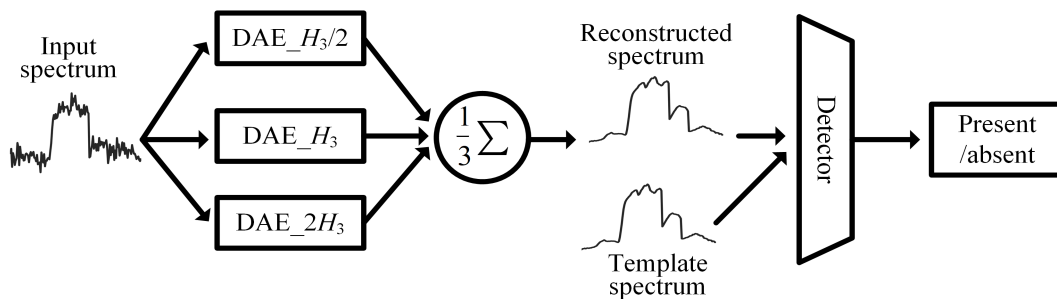


Figure 5.4: Target detection with the proposed MSDAE.

In Figure 5.4, there are three sub-denoising autoencoders ( $DAE_l$ ) ( $l = \frac{H_3}{2}, H_3, 2H_3$ ) in the MSDAE model and  $DAE_l$  means the number of units of hidden layer in current DAE is  $l$ . The proposed method mainly consists three parts. At first, the input is compressed, represented and expanded separately by the encoders. Three vectors of different scales are obtained and decoded into three reconstructed vectors later. Then, the three spectrums reconstructed from different scales are fused into a final reconstructed spectrum. Finally, the final reconstructed spectrum and the template spectrum are input to the detector for target detection. According to the probability of detection ( $P_d$ ) and the preset threshold  $\gamma$ , detection results can be divided into two categories:

$$\begin{cases} P_d > \gamma, & \text{the input is target;} \\ P_d < \gamma, & \text{the input is background.} \end{cases} \quad (5.1)$$

Due to the final reconstructed spectrum is integrated by multiple reconstruction vectors decoded from different scales, it can provide more complex information and



robust features which can help improve detection results.

### 5.2.5 Experimental results

To verify the performance of the proposed method, experiments on simulated and real-world HSIs are done and analyzed in this Section. In addition, the denoising performance of the proposed method is tested on noisy image  $\mathbf{R}$  which is obtained by adding random noise  $\mathbf{N}$  to the image  $\mathbf{H}$ , i.e.  $\mathbf{R} = \mathbf{H} + \mathbf{N}$ . Gaussian noise and multiplicative noise (MPN) are often encountered in hyperspectral imagery [128, 129]. Thus, we use zero-mean white Gaussian noise (WGN) and MPN uniformly distributed with zero-mean are introduced to model the random noise. In the experiment, we will discuss and analyze the removal effects of two kinds of noise with different signal-to-noise ratio (SNR) values. The SNR is estimated as  $\zeta = 10\log_{10} \frac{\|\mathbf{H}\|^2}{\|\mathbf{N}\|^2} = 10\log_{10} \frac{\|\mathbf{H}\|^2}{\|\mathbf{R} - \mathbf{H}\|^2}$ . Moreover, three commonly used denoising algorithms WF, BM3D and DnCNN are served as the contrastive methods.

The most commonly used ACE detector is selected to detect the target, which can be expressed as follows:

$$\text{ACE}(\mathbf{k}) = \frac{(\mathbf{s}^T \mathbf{\Gamma}^{-1} \mathbf{k})^2}{(\mathbf{s}^T \mathbf{s})(\mathbf{k}^T \mathbf{\Gamma}^{-1} \mathbf{k})} \quad (5.2)$$

where  $\mathbf{k}$  is the pixel spectrum and  $\mathbf{s}$  is the target template spectrum.  $\mathbf{\Gamma}$  is the covariance matrix estimated by  $\frac{1}{M_3} \sum_{i=1}^{M_3} (\mathbf{k}_i - \bar{\mathbf{k}})(\mathbf{k}_i - \bar{\mathbf{k}})^T$  and  $\mathbf{k}_i$  is the  $i^{\text{th}}$  column of the matrix  $\mathbf{R}$  with  $\bar{\mathbf{k}} = \frac{1}{M_3} \sum_{i=1}^{M_3} \mathbf{k}_i$ .

Receiver operating characteristic (ROC) curves formed by plotting  $P_d$  and the probability of false alarm (PFA) are mainly used to evaluate the performance of target detection. If the number of target pixels in an HSI is  $N_{\text{target}}$  and the number of pixels belonging to background is  $N_{\text{background}}$ ,  $P_d$  can be calculated by  $P_d = \frac{N_{\text{tp}}}{N_{\text{target}}}$  and the PFA is defined as  $\text{PFA} = \frac{N_{\text{fp}}}{N_{\text{background}}}$ , where  $N_{\text{tp}}$  represents that the number of target pixels is rightly detected as the target and  $N_{\text{fp}}$  means background pixels are falsely detected as targets.

Through experiments, we found that when the MSDAE model is consisted of three sub-denoising autoencoders (DAE\_ $\frac{H_3}{2}$ , DAE\_ $H_3$ , DAE\_ $2H_3$ ), five sub-denoising autoencoders (DAE\_ $\frac{I_3}{4}$ , DAE\_ $\frac{H_3}{2}$ , DAE\_ $I_3$ , DAE\_ $2H_3$ , DAE\_ $4H_3$ ) or seven sub-denoising autoencoders (DAE\_ $\frac{H_3}{8}$ , DAE\_ $\frac{H_3}{4}$ , DAE\_ $\frac{H_3}{2}$ , DAE\_ $H_3$ , DAE\_ $2H_3$ , DAE\_ $4H_3$ , DAE\_ $8H_3$ ), the change in results is not significant. More sub-denoising autoencoders and more hidden layers require more time to train the network. When the number of sub-denoising autoencoders and hidden layers is increased, it has little

influence on the detection results and causes a large amount of calculation. In addition, the detection results of the proposed MSDAE are greatly improved compared to the results of single DAE. Therefore, in order to balance training time and detection results, three sub-denoising autoencoders with one hidden layer are selected in the following experiments, which help us get the best results in less time. The input layer of each sub-denoising autoencoder is randomly dropping out units with a 10% probability. To further improve the robustness of the model, Gaussian noise with a zero mean and 0.01 variance is added to the samples before they are input to the DAE. A batch learning [130] is used to update the weights via minimizing the reconstruction error calculated by MSE, which improves the training efficiency. In the following experiments, the size of batch is setting as 128 and the number of epochs is 500.

### 5.2.5.1 Experiments on simulated data

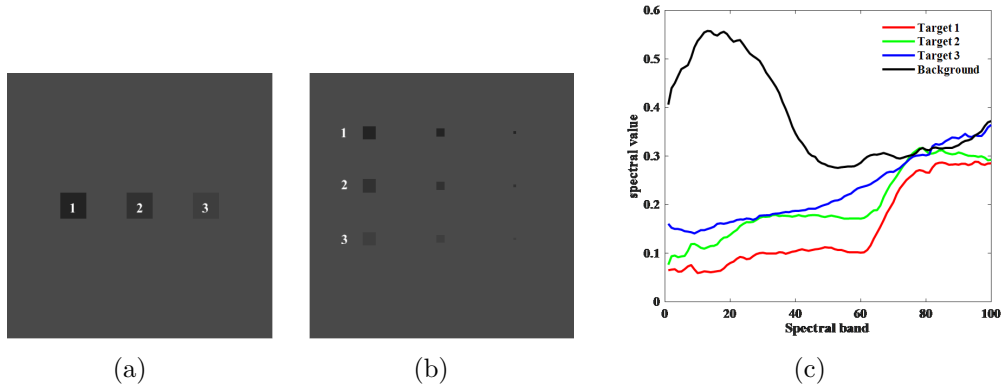


Figure 5.5: Simulated scenes: (a)  $simulated_a$  (the 80<sup>th</sup> band), (b)  $simulated_b$  (the 80<sup>th</sup> band), (c) spectral signatures.

When we generate synthetic images, spectral signatures are from Pavia University dataset and 100 bands are used after removing several low-signal bands [131]. In the experiment, two simulated data,  $simulated_a$  in Figure 5.5 (a) and  $simulated_b$  in Figure 5.5 (b) are generated with size being  $100 \times 100 \times 100$ , and the spectral signatures are shown in Figure 5.5 (c). The targets are mixed to the background according to the linear mixing model [132] when target abundance is 80%. For  $simulated_a$  HSI, the spatial size of the three targets is  $10 \times 10$ . For  $simulated_b$  HSI, the spatial size is  $5 \times 5$  along the first column,  $3 \times 3$  along the second column,  $1 \times 1$  along the last column.

Due to  $simulated_a$  and  $simulated_b$  have 100 spectral bands, two same MSDAEs

can be constructed separately. Each MSDAE consists of three DAEs with the number of units in hidden layers is 50, 100, 200, respectively. For better visual observation, the original spectrum, noisy spectrum [133] and reconstructed spectrum of Target 2 in  $simulated_a$  and  $simulated_b$  are plotted in Figure 5.6.

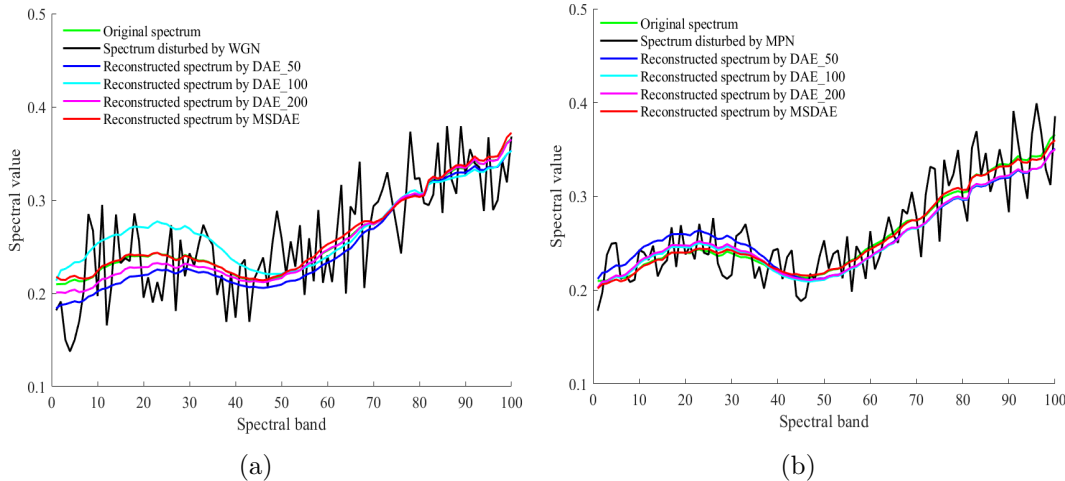


Figure 5.6: Comparison of different spectral curves of Target 2 in simulated images: (a)  $simulated_a$ , (b)  $simulated_b$ .

Figure 5.6 reflects that reconstructed spectrums have largely removed noise and are similar to the corresponding original spectrum. When the image is disturbed by WGN, DAE\_200 performs better than the other two single DAEs. When the image is disturbed by MPN, DAE\_100 and DAE\_200 perform better than DAE\_50. In general, the spectrum represented by MSDAE under two kinds of noise is closer to the original signature than the spectrum reconstructed any single DAE.

In the experiment, we range  $\zeta$  from 10 dB to 80 dB to compare the performance of the proposed MSDAE with a single DAE. We find when the SNR exceeds 30 dB for simulated images, the  $P_d$  values are always one. Other relevant  $P_d$  values under  $PFA = 10^{-3}$  of Target 2 in two simulated HSIs are given in Table 5.1.

We can see from Table 5.1 that results obtained by DAE\_200 are better for  $simulated_a$  while DAE\_100 are better for  $simulated_b$  with WGN when only one DAE is used. However, DAE\_100 performs better for both simulated image under the condition of MPN noise. Therefore, if the spectrum is reconstructed by a single DAE, the detection results are unstable and easily affected by network structure. The proposed MSDAE helps us yield best detection results than any sub-denoising autoencoder whether the image is destroyed by WGN or MPN. This is because the final reconstructed spectrum is restored from features of different scales, which provides more information for the subsequent target detection.

Table 5.1: The  $P_d$  values of different models for  $simulated_a$  and  $simulated_b$ .

Noise	Model	Simulated <sub>a</sub>				Simulated <sub>b</sub>			
		10 dB	20 dB	25 dB	30 dB	10 dB	20 dB	25 dB	30 dB
WGN	DAE_50	0.60	0.80	0.96	1.00	0.66	0.86	0.97	1.00
	DAE_100	0.54	0.58	0.96	1.00	0.71	0.80	1.00	1.00
	DAE_200	0.64	0.73	0.99	1.00	0.60	0.66	0.80	0.97
	MSDAE	0.85	0.96	1.00	1.00	0.86	0.97	1.00	1.00
MPN	DAE_50	0.74	0.96	1.00	1.00	0.77	0.94	1.00	1.00
	DAE_100	0.78	0.99	1.00	1.00	0.83	0.97	1.00	1.00
	DAE_200	0.71	0.84	0.99	1.00	0.71	0.86	1.00	1.00
	MSDAE	0.83	1.00	1.00	1.00	0.89	1.00	1.00	1.00

To evaluate the performance of the proposed model, when SNR is 20 dB, the ROC curves of Target 2 for  $simulated_a$  and  $simulated_b$  denoised by different methods are depicted in Figure 5.7 and in Figure 5.8, respectively.

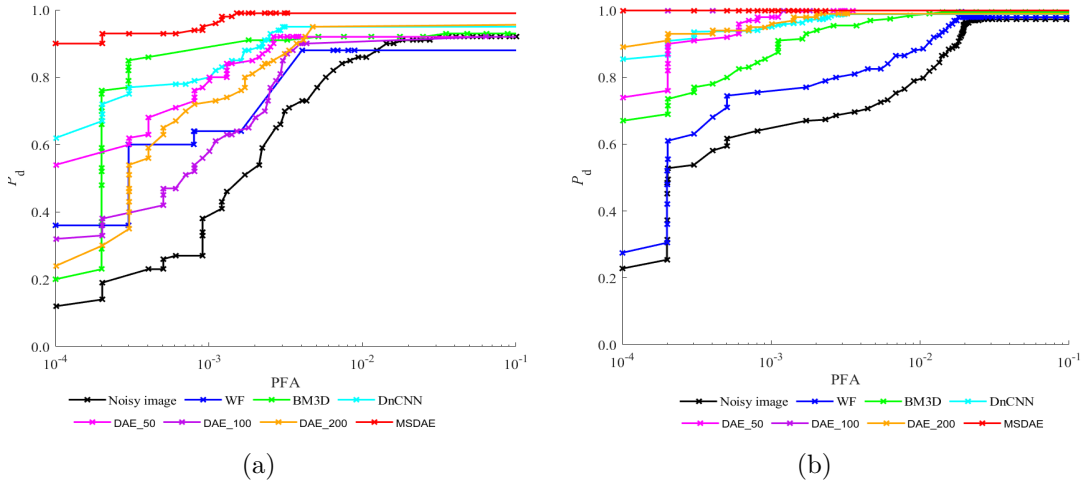


Figure 5.7: ROC curves of  $simulated_a$  under different noises with SNR being 20 dB: (a) WGN, (b) MPN.

For  $simulated_a$ , we can see that the images denoised by BM3D and DnCNN get higher  $P_d$  values compared to sub-denoising autoencoder in reducing WGN. DAE and DnCNN show great potential in removing MPN. It is worth noting that under the two noise models, the detection results obtained by the proposed MSDAE outperform other models, and the related  $P_d$  values are more stable. For  $simulated_b$ , we can find that the parameter setting of single DAE has great influence on performance. The  $P_d$  values based on DAE\_200 are lower than those by DnCNN, but DAE\_50 performs better than other models. The  $P_d$  values of reconstructed image by MSDAE under two kinds of noise are significantly improved, especially when the

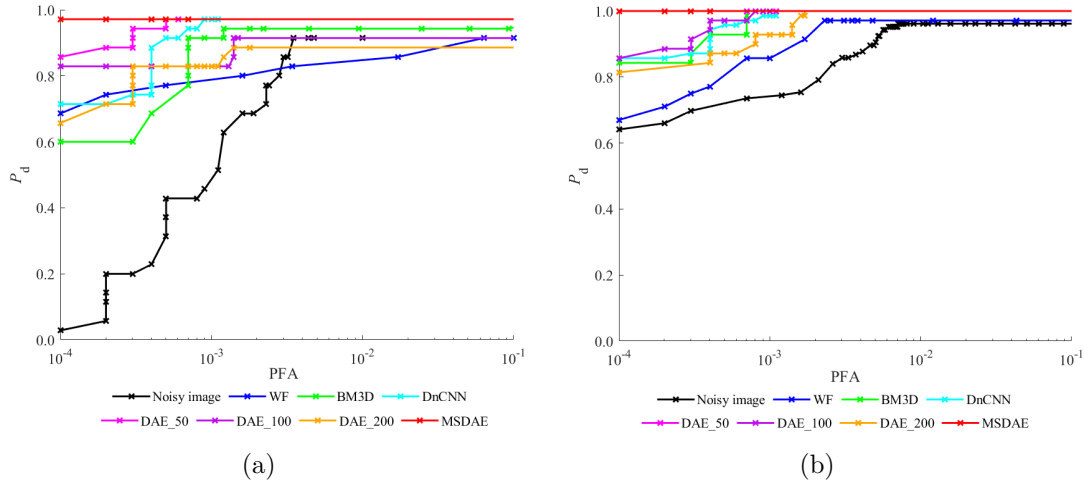


Figure 5.8: ROC curves of  $simulated_b$  under different noises with SNR being 20 dB: (a) WGN, (b) MPN.

PFA values are small.

### 5.2.5.2 Experiments on real-world data

Since the simulated HSIs are generated in an idealistic scene, two real-world HSIs, referred to as  $HSI_b$  (Figure 8 (a)) and  $HSI_c$  (Figure 8 (b)), taken from the entire HYDICE are considered in this part to test the performance of the proposed model. Both  $HSI_b$  and  $HSI_c$  have 148 spectral bands with spatial size of  $HSI_b$  being  $141 \times 126$  and  $HSI_c$  being  $140 \times 140$ . For  $HSI_b$ , the target size is  $12 \times 12$ . For  $HSI_c$ , the spatial size of each column of targets is  $5 \times 3$ ,  $3 \times 3$  and  $1 \times 1$ .

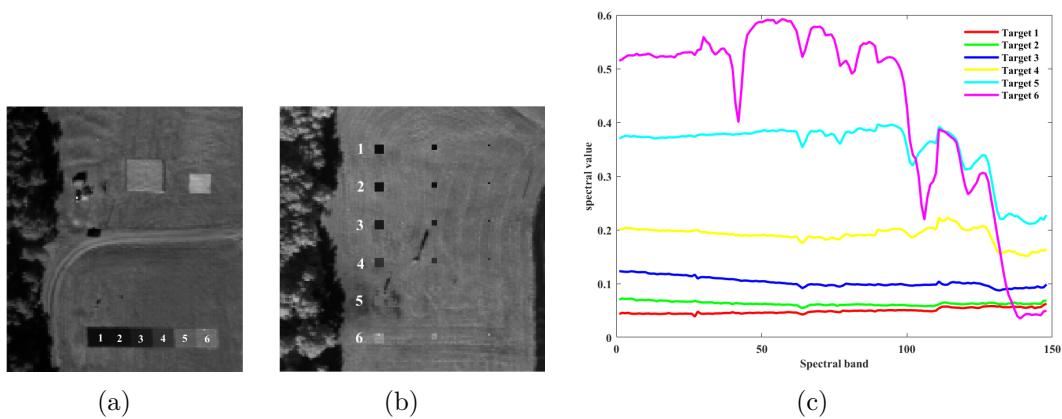


Figure 5.9: Real-world scenes: (a)  $HSI_b$  (the 60<sup>th</sup> band), (b)  $HSI_c$  (the 60<sup>th</sup> band), (c) spectral signatures.

The reconstruction process of these two HSIs is similar to the simulated HSIs.

The reconstructed spectrums of Target 4 by different DAE models are shown in Figure 5.10.

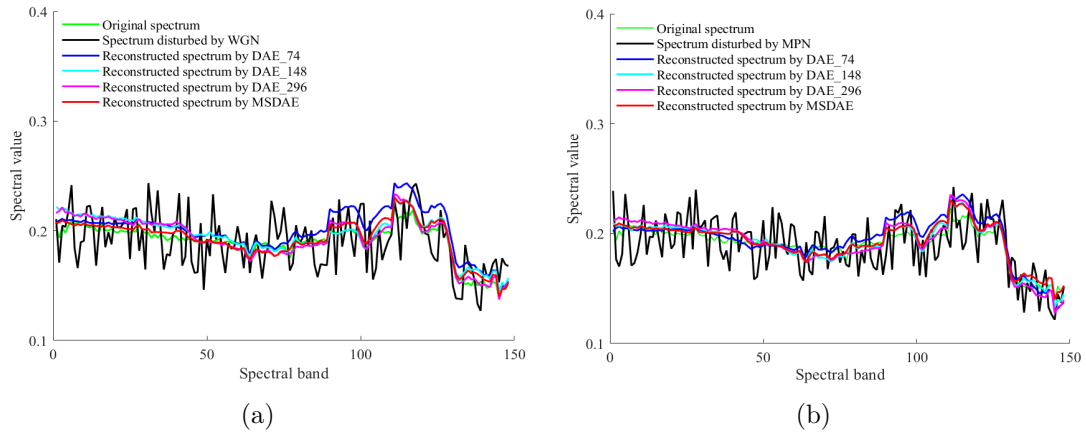


Figure 5.10: Comparison of different spectral curves of Target 4 in real-world scenes: (a)  $HSI_b$ , (b)  $HSI_c$ .

The results in Figure 5.10 demonstrate that spectrum reconstruction can reduce the noise interference and retain spectral characteristics, which lays a foundation for target detection.

When the  $\zeta$  is increased from 10 dB to 80 dB, we find when the SNR exceeds 40 dB for HYDICE images, the  $P_d$  values are always one, so we don't list them in the table. Other relevant  $P_d$  values under  $PFA = 10^{-3}$  for different reconstructed images are listed in Table 5.2.

Table 5.2: The  $P_d$  values of different models for  $HSI_b$  and  $HSI_c$ .

Noise	Model	$HSI_b$				$HSI_c$			
		10 dB	20 dB	30 dB	40 dB	10 dB	20 dB	30 dB	40 dB
WGN	DAE_74	0.63	0.94	1.00	1.00	0.60	0.74	0.88	1.00
	DAE_148	0.69	0.71	0.86	0.95	0.74	0.91	0.97	1.00
	DAE_296	0.73	0.96	1.00	1.00	0.88	0.99	1.00	1.00
	MSDAE	0.83	1.00	1.00	1.00	0.91	1.00	1.00	1.00
MPN	DAE_74	0.72	0.94	0.96	1.00	0.69	0.89	0.94	1.00
	DAE_148	0.51	0.88	0.94	0.99	0.69	0.91	1.00	1.00
	DAE_296	0.65	0.99	1.00	1.00	0.86	0.99	1.00	1.00
	MSDAE	0.90	1.00	1.00	1.00	0.94	1.00	1.00	1.00

We can see from Table 5.2 that among all single DAE models, DAE\_296 performs better under two different types of noise in real experiment. But the proposed MSDAE still has a significant improvement compared with DAE\_296, especially when the SNR value is small. Overall, the proposed MSDAE achieves the best results in different SNR values.

The ROC curves of Target 4 in  $HSI_b$  and  $HSI_c$  with SNR being 20 dB denoised by different methods are depicted in Figure 5.11 and Figure 5.12.

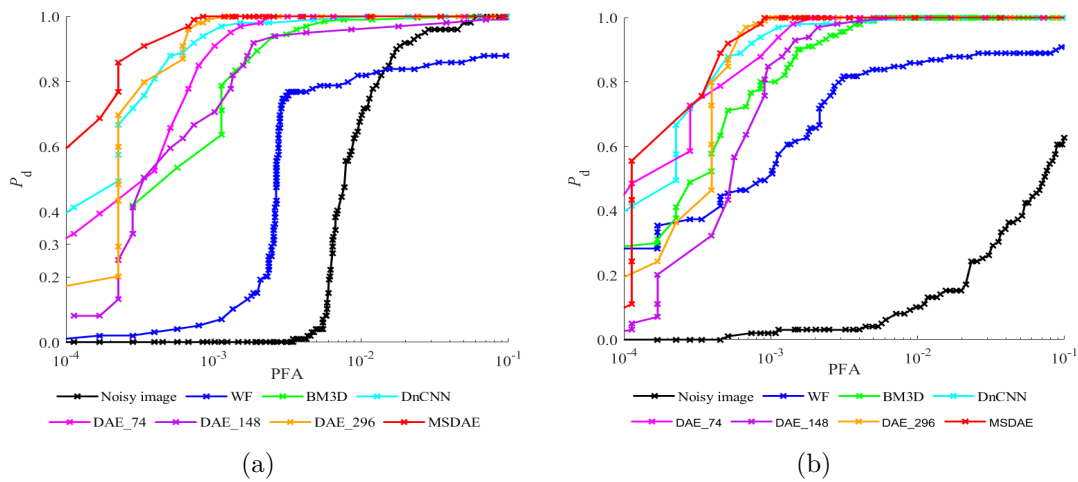


Figure 5.11: ROC curves of  $HSI_b$  under different noises with SNR being 20 dB: (a) WGN, (b) MPN.

It can be seen from Figure 5.11, MSDAE, DAE\_296 and DnCNN obtain better detection results whether the  $HSI_b$  is destroyed by WGN or MPN.

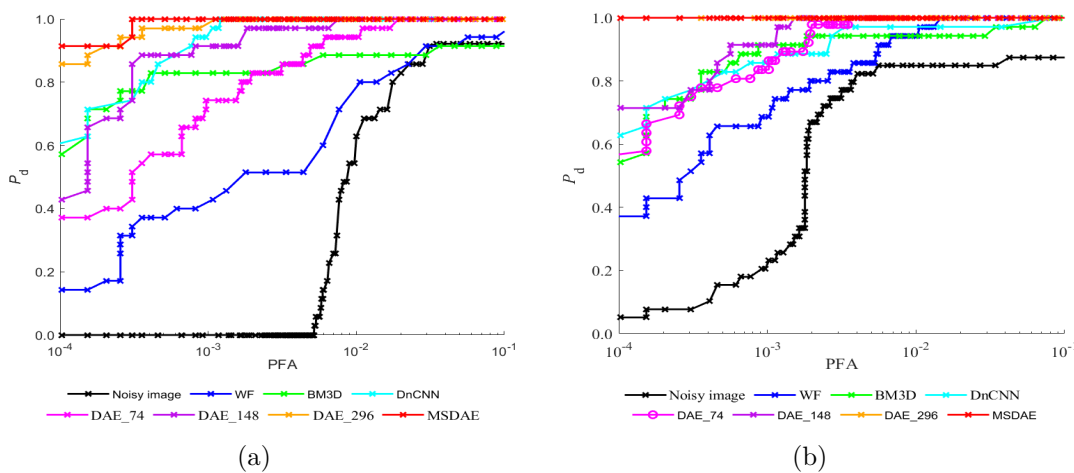


Figure 5.12: ROC curves of  $HSI_c$  under different noises with SNR being 20 dB: (a) WGN, (b) MPN.

Figure 5.12 shows the performance of the proposed MSDAE for  $HSI_c$  is far superior to other methods, which proves the potential of the proposed method in target detection.

### 5.2.5.3 Small target detection

To verify whether the proposed method has a good ability in preserving the small targets, the detection maps of  $simulated_b$  and  $HSI_c$  with the more commonly used WGN and SNR being 20dB are compared. All the detection maps are obtained under  $PFA = 10^{-3}$  after denoising or reconstructing by different models, which are shown in Figure 5.13 and Figure 5.14, respectively.

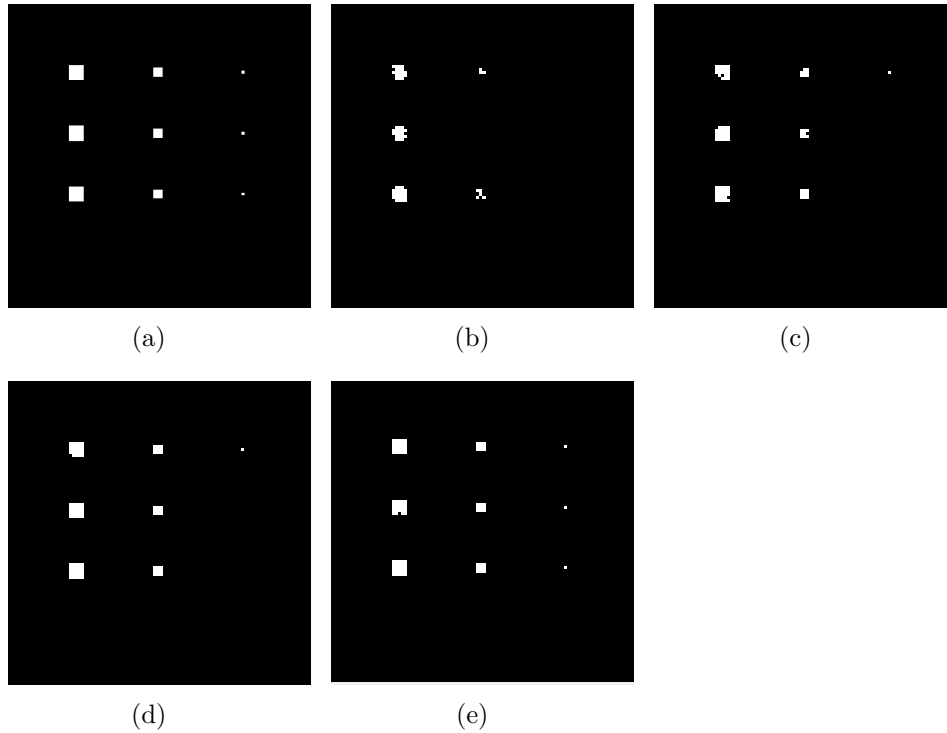


Figure 5.13: Detection maps of  $simulated_b$ : (a) ground truth, (b) denoised by WF, (c) reconstructed by BM3D, (d) reconstructed by DnCNN, (e) reconstructed by MSDAE.

From Figure 5.13 we can find that the target with size of  $1 \times 1$  are all not detected after denoising by WF. In the detection maps of denoising by BM3D and DnCNN, both of them have two small targets with size of  $1 \times 1$  are missed. However, after reconstructing by MSDAE, only one pixel is undetected and the  $1 \times 1$  targets are all detected, which indicates that the proposed MSDAE can better preserve the small targets while removing the noise.

It can be seen from Figure 5.14, when the target size is  $1 \times 1$ , only the detection result based on MSDAE is the most satisfactory. Other detection results after denoising have problems of missing small targets, especially for Target 1, Target 2 and Target 4.



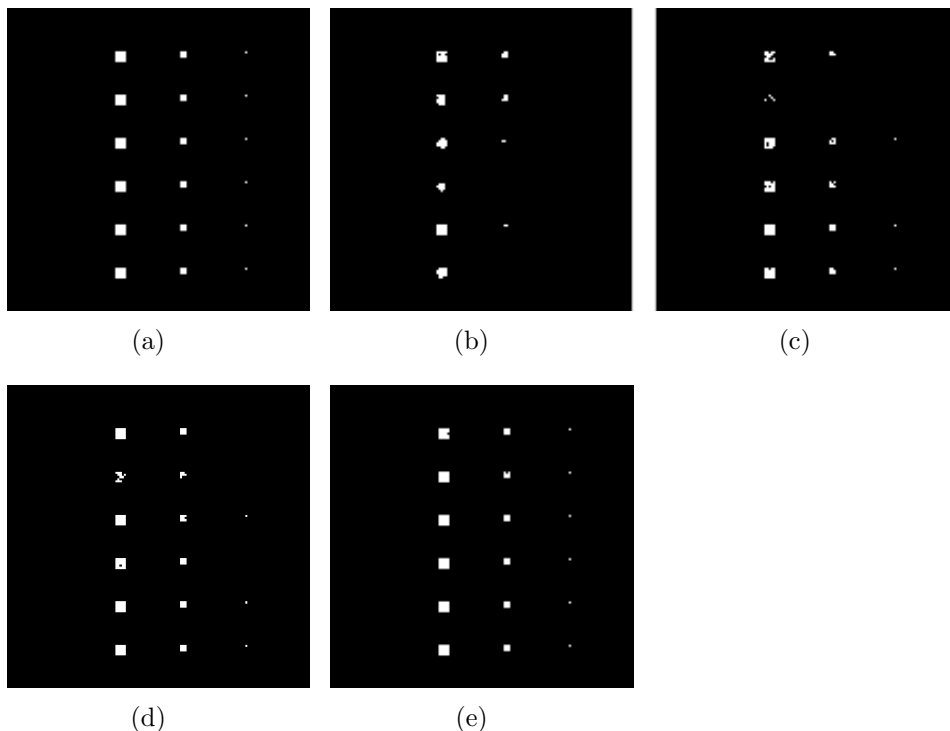


Figure 5.14: Detection maps of  $HSI_c$ : (a) ground truth, (b) denoised by WF, (c) denoised by BM3D, (d) denoised by DnCNN, (e) reconstructed by MSDAE.

The results of simulated images and real-world HSIs highlight the prospects of the proposed MSDAE for small target detection in the presence of WGN interference.

## 5.3 Unsupervised segmentation for small target detection

Image segmentation can be treated as a pixel classification problem. In order to get rid of the segmentation ground truth, unsupervised segmentation is an effective way. K-mean clustering is one of the main unsupervised segmentation methods [134], but it has limitations in mining the spectral-spatial information compared with deep learning models [135]. Therefore, unsupervised segmentation based on deep learning model is investigated for improvement of small target detection.

### 5.3.1 Unsupervised segmentation

CNN has strong feature extraction abilities and has been widely used in image processing. However, label samples are unusually required to optimize the network. In [136,137], an unsupervised image segmentation method is developed by assigning

labels to pixels based on features. Label prediction and network parameter learning are alternately iteratively trained and optimized. The illustration of unsupervised segmentation based on CNN is shown in Figure 5.15.

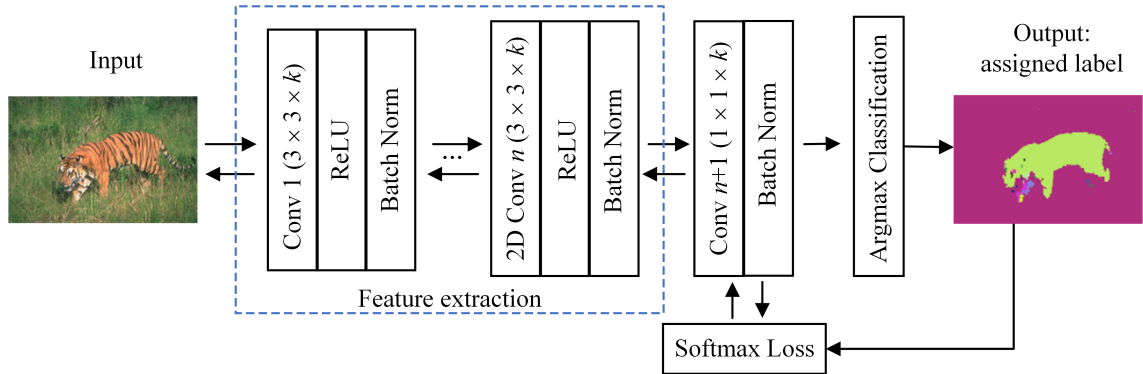


Figure 5.15: Unsupervised segmentation based on 2D-CNN.

In Figure 5.15,  $s \times s \times k$  represents that there are  $k$  convolutional kernels with size of  $s \times s$  in current layer. Zero padding is introduced to keep the input and output having the same width and height. The training procedure can be mainly divided into two parts: the forward propagation and the backward propagation. The forward propagation assigns label based on feature maps. The largest value of the feature map is the label of the corresponding pixel. The backward propagation optimizes network parameters based on gradient decent and assigned labels. The network optimization process tries to meet the following criteria:

- (1) Pixels of similar features are designed with the same label.
- (2) Pixels of spatial continuity are designed with the same label.
- (3) The number of unique cluster labels is designed to be large.

Criteria (1) and (2) are dedicated to merging neighboring pixels with similar characteristics into the same class. But there may be extreme situations where all pixels are merged into one class. In order to avoid this situation, criteria (3) is necessary.

### 5.3.2 Experimental results

In the experiment, we choose  $HSI_a$  as the target data to test the effectiveness of unsupervised segmentation for small target detection.

We have known that HSIs contain hundreds of bands. 2D convolution for multi-channel is used in 2D-CNN for unsupervised segmentation and the number of channels is equal to the number of bands. The corresponding network structure used

for segmentation is listed in Table 5.3 and SGD is used to update weights. The maximum number of segmentation classes is set to 8.

Table 5.3: Network structures of 2D-CNN.

Layer	Input Size	Kernel	Output
Conv-1	$316 \times 216 \times 148 \times 1$	$3 \times 3 \times 148 \times 24$	$316 \times 216 \times 24 \times 1$
Conv-2	$316 \times 216 \times 24 \times 1$	$3 \times 3 \times 24 \times 24$	$316 \times 216 \times 24 \times 1$
Conv-3	$316 \times 216 \times 24 \times 1$	$1 \times 1 \times 24 \times 24$	$316 \times 216 \times 24 \times 1$

It can be found from Table 5.3 that the length and width of the output have not changed and are the same as the size of the input image. After 100 epochs of training, the segmentation map is shown in Figure 5.16 (b).

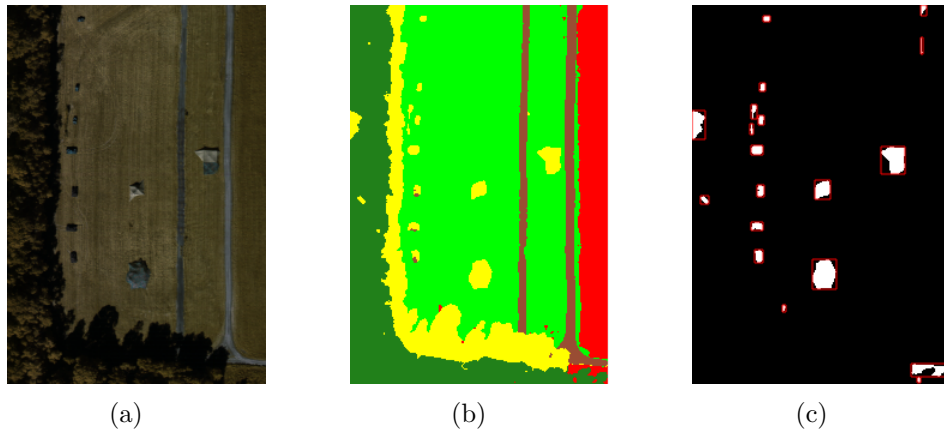


Figure 5.16:  $HSI_c$ : (a) Composite image, (b) Segmentation map, (c) Small target regions of interest.

We can see from Figure 5.16 (b) that the target image is segmented into 5 classes. Although the segmented classes are smaller than the actual classes, the neighboring targets of different classes can be distinguished well. Pixels of the same color are considered to have the same pixel value. According to the pixel value, a set of connected components can be obtained. Given a preset threshold, we select the regions where the number of pixels contained in the connected components is lower than this threshold as the ROIs. As shown in Figure 5.16 (c), the white part is the area that may contain small targets.

Taking target 3 in  $HSI_a$  as the target to be detected and ACE as the detector, the detection maps are shown in Figure 5.17 when we directly detect the target in the entire image.

As described in Eq. (5.1), by comparing the calculated  $P_d$  value with the threshold  $\gamma$ , we can determine whether the current pixel is the target. It can be seen from

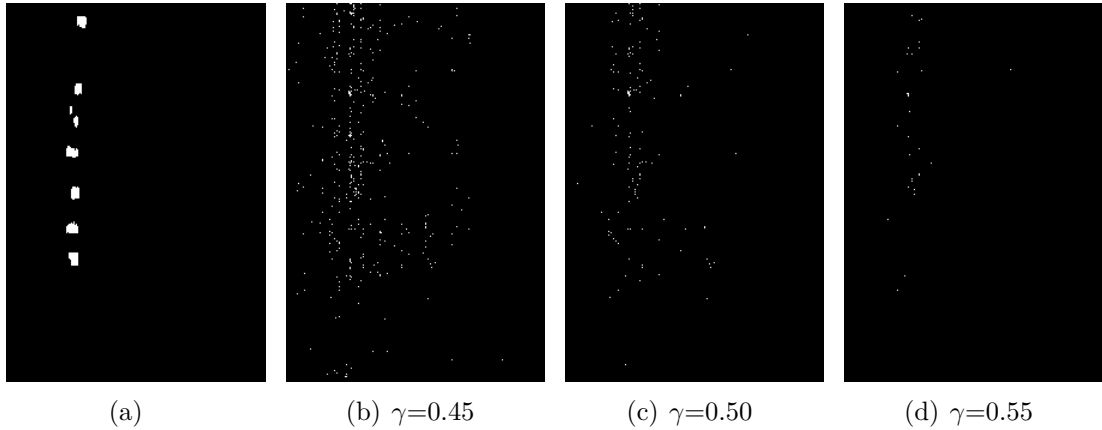


Figure 5.17: Detection maps of  $HSI_a$  without unsupervised segmentation: (a) Ground truth. (b) Detection map with  $\gamma=0.45$ , (c) Detection map with  $\gamma=0.50$ , (d) Detection map with  $\gamma=0.55$ .

Figure 5.17, when the  $\gamma$  value is 0.45 and 0.50, many pixels are mistakenly detected as targets. When the  $\gamma$  value is 0.55, most of the detected pixels are target pixels. But the pixels are scattered, which is difficult to locate the target.

Figure 5.18 shows the detection maps of Target 3 in  $HSI_a$  after selecting ROIs with unsupervised segmentation.

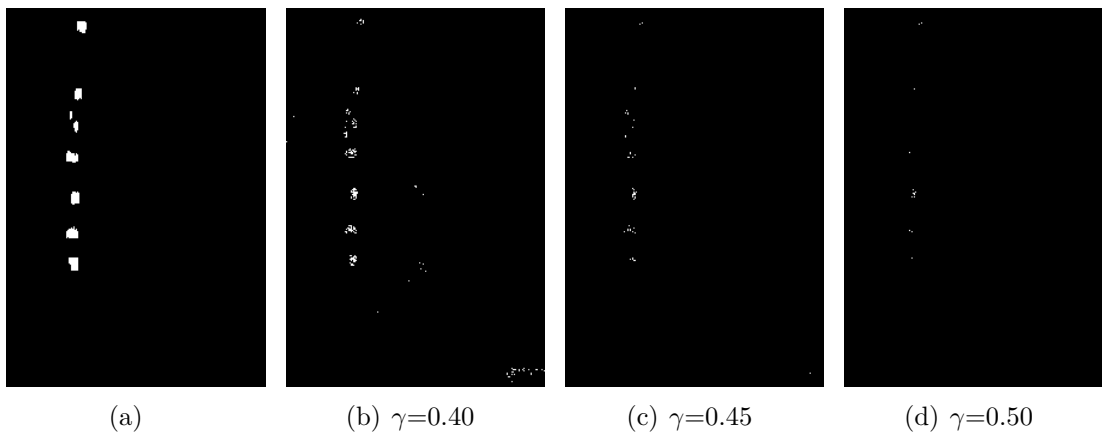


Figure 5.18: Detection maps of  $HSI_a$  with unsupervised segmentation: (a) Ground truth. (b) Detection map with  $\gamma=0.40$ , (c) Detection map with  $\gamma=0.45$ , (d) Detection map with  $\gamma=0.5$ .

It can be found that the detection maps in Figure 5.18 help detect and locate the target more accurately compared to the detection maps in Figure 5.17. In particular, when the  $\gamma$  value is set to 0.40, only a few pixels are misdetections. According to the detection map in Figure 5.18 (b), these small targets can be well located.

Experimental results prove that the first stage of unsupervised segmentation can

effectively reduce the detection range and remove interfering pixels. Besides, the detection results can be further improved. Considering that in practical applications, the background area is generally much larger than the target area, unsupervised segmentation to select ROIs for target detection is very promising.

## 5.4 Conclusion

In this Chapter, we introduce DAE to HSIs to reconstruct spectrum for removing noise interference and improving performance of target detection. In order to further improve the target detection results, we design a MSDAE model to mine spectral characteristics as much as possible. The original spectrum is compressed, represented and expanded and then they are decoded into different reconstructed spectrums. These reconstructed spectrums from different scales features are finally merged to one spectrum for target detection, which effectively exploits the spectral characteristics and invariant features. Experimental results of simulated and real-world HSIs with WGN and MPN demonstrate the effectiveness. Compared with a single DAE and other mentioned methods, the performance of proposed MSDAE is more stable and gets higher  $P_d$  values. Besides, the proposed method shows great ability in preserving small targets. The promising results show significant potential of the proposed MSDAE model for target detection. In addition, in order to further improve the detection of small targets, a two-stage method including segmentation and detection is proposed. Unsupervised segmentation is used to obtain ROIs and narrow the detection range and then target detection is performed in the ROIs. Experimental results prove that the proposed two-stage method is effective and promising for detecting small targets.

# Conclusion and future works

## Conclusion

This dissertation aims at the study of HSI processing and its applications. Target classification and target detection are key techniques for hyperspectral applications, and feature extraction is the most significant step of classification and detection. Therefore, feature extraction, classification and detection are the main research topics investigated. Since deep learning has strong capabilities in data mining and feature extraction, we are devoted to processing HSIs with deep learning models.

Considering that HSIs are 3D tensor data and CNN can handle multi-dimensional data flexibly, hyperspectral classification models based on 2D-CNN and 3D-CNN are studied. Due to the model performance is greatly influenced by the parameter settings, a parameter tuning method (2D-CNN-PT) based on 2D-CNN with unique variable principle is proposed for hyperspectral classification, and the optimal parameters are selected mainly based on classification results, which helps us further improve network performance. The parameters of the 3D-CNN or other models can also be selected as proposed 2D-CNN-PT. Experimental results on real-world HSIs demonstrate that appropriate parameter settings can help to obtain better classification results. Besides, 3D-CNN shows greater potential in fully exploiting the spectral-spatial information and helps us obtain higher classification accuracy compared to 2D-CNN. However, there are more parameters in 3D-CNN and the optimization of CNN is supervised, which means a large number of labeled samples are required to guarantee the network performance. Unfortunately, the labeled samples are limited in HSIs. To get rid of the limitation of labeled samples, a 3D-CNN-TV method based on 3D-CNNs combined with transfer learning and virtual samples is proposed. Transfer learning helps to apply knowledge learned in source data with sufficient labeled samples to novel target data. The weights of the 3D-CNN with target data are transferred from another 3D-CNN which has same network struc-

ture as the previous 3D-CNN and is optimized by source data, so that the 3D-CNN corresponding to target data has a strong feature extraction ability at the beginning. Virtual samples generated from the original samples can greatly increase the number of labeled samples. The introduction of transfer learning and virtual samples effectively alleviates the problem of insufficient labeled samples. Experimental results on real-world HSIs show either transfer learning or virtual samples can effectively alleviate the problem of insufficient labeled samples, and the combination of transfer learning and virtual samples helps to yield highest classification results.

Unsupervised learning is a type of algorithm that learns patterns from unlabeled data. If we can learn the features in the hyperspectral data in an unsupervised way, the problem of insufficient labeled samples can be solved. GAN is trained in an adversarial way requiring no labeled samples and it has been one of the most promising unsupervised learning representatives. AE can be optimized by minimizing the error between the reconstructed data and the input data, and no labels are involved. GAN and AE are typical unsupervised training networks. Therefore, unsupervised feature extractors based on GAN and AE are investigated in this thesis. Firstly, unsupervised feature extraction methods based on 3D-WGAN-GP is proposed. With the help of transfer learning, the discriminator of the optimized 3D-GAN-GP is transferred as the feature extractor. Then, a multi-level feature extraction method based on 3D-CAE is proposed. The proposed multi-level features are directly obtained from different encoded layers of the optimized encoder, which is more efficient when compared to training multiple networks and makes full use of the information at the bottom and top layers. Finally, a 3D-M<sup>2</sup>CAE framework is designed to balance different targets and improve classification results of small targets. Three 3D-CAEs with different input sizes centered on the observed pixel are used to build the framework and extract features. The framework is established and trained in a progressive way with the help of transfer learning to save training time. Benefiting from this training method, the features of the same target from different sizes are obtained in a more efficient way. Features from the same target and different sizes can be obtained, which can greatly improve the robustness of features to size changes. The experimental results verify the effectiveness of three proposed unsupervised feature extractors and show great application prospects without labeled samples.

In addition to feature extraction and classification, target detection is also studied. Target detection can be treated as a binary classification task. According to the spectral characteristics, pixels can be classified as target or background. Due to spectral variations caused by noise or environment, the within-class variation is

enlarged which degrades the performance of detectors, especially when the target size is small. We design a MSDAE model to denoise and mine the invariant features. The final spectrum input to the detector is fused by reconstructed spectrums from different scales representations, which provides more complementary information and more robust features for subsequent detection. Experiments on simulated and real-world data demonstrate that the proposed MSDAE can not only improve the target detection but also has great potential for preserving small targets in denoising. In addition, to further improve the detection of small targets, a two-stage method including segmentation and detection is proposed. Unsupervised segmentation is used to obtain ROIs and narrow the detection range, and then target detection is performed in the ROIs. Experimental results demonstrate the effectiveness of the proposed method and have promising prospects in small target detection.

## Future works

In this dissertation, we successfully apply the deep learning models to HSIs to extract features, classify and detect targets. But there are still some aspects that need to be further study. We propose here some research directions for a future extension of this work.

- In the process of feature extraction and classification, we don't separately consider and analyze the effect of noise on the results. So, the effect of different types of noise on the results and whether the noise is removed during feature extraction need to be further studied.

- In this thesis, a parameter selection method based on unique variable principle is proposed and the parameters are adjusted one by one. Taking into account the correlation between the parameters, a new algorithm needs to be developed in order to achieve the best overall performance of network.

- As described in Chapter 5, we design a MSDAE model based on DAE to reconstruct the spectrum and remove noise. The experiments show that the noise is well removed, but only spectral information is considered as input. In the future, a denoising model that consider both spectral and spatial information need to be studied.

- When a small target is a research object, its size and number of samples are much smaller compared to the background, making the processing of small targets more difficult and challenging. More work needs to be focused on small targets in the future.



- In previous research, we mainly focus on classification and detection. In the future, we want to use deep learning to analyze features for spectral unmixing.

# List of Figures

1	Schéma de principe de l'imagerie hyperspectrale. . . . .	1
2	Cartes de classification de l'Université de Pavie obtenues par différentes méthodes: (a) Vérité terrain, (b) 2D-CNN, (c) 3D-CNN, (d) 3D-CNN-TV. . . . .	4
3	Cartes de classification de l'Université de Pavie obtenues par différentes méthodes: (a) Vérité terrain, (b) 3D-WGAN-GP, (c) 3D-CAE avec fonctionnalités à un seul niveau, (d) 3D-CAE avec fonctionnalités multi-niveaux. . . . .	5
4	Cartes de classification du $HSI_a$ obtenues par différentes méthodes: (a) Vérité terrain, (b) FA, (c) DBN, (d) Proposé 3D-M <sup>2</sup> CAE. . . . .	6
5	Courbes ROC de $HSI_b$ sous différents bruits avec un SNR de 20 dB: (a) WGN, (b) MPN. . . . .	8
6	Schematic diagram of hyperspectral imaging. . . . .	15
1.1	A conventional CNN structure. . . . .	22
1.2	2D convolution operation. . . . .	22
1.3	2D pooling operation: (a) Max-pooling, (b) Mean-pooling. . . . .	23
1.4	Hyperspectral classification based on a 2D-CNN with parameter tuning. . . . .	23
1.5	Data sets: (a) False-color image of Pavia University. (b) Ground truth of Pavia University. (c) False-color image of Indian Pines. (d) Ground truth of Indian Pines. . . . .	25
1.6	OA values under different input sizes and training set ratio. . . . .	28
1.7	Different activation functions. . . . .	31
1.8	OA values and computation times under different batch sizes. . . . .	32
1.9	OA values under different number of convolutional kernels. . . . .	33
1.10	Classification maps of Pavia University under different methods: (a) Ground truth, (b) FA, (c) DBN, (d) 2D-CNN-PT. . . . .	35

1.11	Classification maps of Indian Pines under different methods: (a) Ground truth, (b) FA, (c) DBN, (d) 2D-CNN-PT. . . . .	35
2.1	A conventional 3D-CNN for hyperspectral classification. . . . .	38
2.2	3D convolution operation. . . . .	38
2.3	DP algorithm for line simplification. . . . .	40
2.4	The modified DP algorithm. . . . .	42
2.5	The flowchart of proposed FMDP for band selection. . . . .	43
2.6	Pavia University: (a) Original spectral curve, (b) Fitted spectral curve with selected bands. . . . .	44
2.7	Indian Pines: (a) Original spectral curve, (b) Fitted spectral curve with selected bands. . . . .	44
2.8	OA values under different number of selected bands. . . . .	45
2.9	OA values of Pavia University based on different number of bands selected by different methods. . . . .	46
2.10	OA values of Indian Pines based on different number of bands selected by different methods. . . . .	47
2.11	Proposed hyperspectral classification based on a 3D-CNN. . . . .	48
2.12	Proposed 3D-CNN-TL framework. . . . .	49
2.13	Flow chart of proposed 3D-CNN-TV framework. . . . .	51
2.14	Source data sets: (a) Pavia Centre, (b) Salinas. . . . .	53
2.15	Relationship between $r$ and OA values. . . . .	56
2.16	Classification maps of Pavia University under different methods: (a) 3D-CNN, (b) 3D-CNN-TL, (c) 3D-CNN-VS, (d) 3D-CNN-TV. . . . .	57
2.17	Classification maps of Indian Pines under different methods: (a) 3D-CNN, (b) 3D-CNN-TL, (c) 3D-CNN-VS, (d) 3D-CNN-TV. . . . .	58
3.1	Standard GAN. . . . .	60
3.2	Weight distribution with weights clipping. . . . .	61
3.3	Weight distribution with gradient penalty. . . . .	62
3.4	Framework of proposed dimensionality reduction method. . . . .	63
3.5	Proposed unsupervised feature extraction method based on 3D-WGAN-GP. . . . .	64
3.6	Classification maps of Pavia University under different methods: (a) False-color image, (b) Ground truth, (c) GAN-softmax, (d) GAN-SVM, (e) 3D-WGAN-softmax, (f) 3D-WGAN-SVM, (g) 3D-WGAN-GP-softmax, (h) 3D-WGAN-GP-SVM. . . . .	70

3.7	Classification maps of Indian Pines under different methods: (a) False-color image, (b) Ground truth, (c) GAN-softmax, (d) GAN-SVM, (e) 3D-WGAN-softmax, (f) 3D-WGAN-SVM, (g) 3D-WGAN-GP-softmax, (h) 3D-WGAN-GP-SVM. . . . .	71
4.1	Conventional AE architecture. . . . .	75
4.2	CAE architecture. . . . .	75
4.3	Proposed framework for multi-level feature extraction. . . . .	77
4.4	Proposed framework for multi-level feature extraction. . . . .	80
4.5	Classification accuracy of Pavia University under different input sizes. . . . .	82
4.6	Classification accuracy of Indian Pines under different input sizes. . . . .	83
4.7	Classification accuracy of Pavia University based on multi-level features with different numbers of encoded layers. . . . .	84
4.8	Classification accuracy of Indian Pines based on multi-level features with different numbers of encoded layers. . . . .	85
4.9	Pavia University: (a) Composite image, (b) Ground truth, (c) FA, (d) DBN, (e) 2D-CNN, (f) SAE, (g) 3D-CAE (single-level features), and (h) 3D-CAE (multi-level features). . . . .	88
4.10	Indian Pines: (a) Composite image, (b) Ground truth, (c) FA, (d) DBN, (e) 2D-CNN, (f) SAE, (g) 3D-CAE (single-level features), and (h) 3D-CAE (multi-level features). . . . .	89
4.11	Proposed 3D-M <sup>2</sup> CAE framework for unsupervised feature extraction. . . . .	90
4.12	$HSI_a$ : (a) False-color image. (b) Ground truth. . . . .	92
4.13	OA values of three small targets based on 3D-CAE I under different input sizes. . . . .	94
4.14	OA values of three small targets based on 3D-CAE II under different input sizes. . . . .	94
4.15	OA values of three small targets based on 3D-CAE III under different input sizes. . . . .	95
4.16	OA values of Target 1 based on features obtained from different networks. . . . .	96
4.17	OA values of Target 2 based on features obtained from different networks. . . . .	96
4.18	OA values of Target 3 based on features obtained from different networks. . . . .	97

4.19	OA values of Pavia University based on features obtained from different networks. . . . .	98
4.20	AA values of Pavia University based on features obtained from different networks. . . . .	99
4.21	$\kappa$ values of Pavia University based on features obtained from different networks. . . . .	99
4.22	Classification maps of $HSI_a$ obtained by different methods: (a) FA, (b) SAE, (c) DBN, (d) 2D-CNN, (e) 3D-CAE I, (f) 3D-CAE II, (g) 3D-CAE III, (h) Proposed 3D-M <sup>2</sup> CAE. . . . .	100
4.23	Classification maps of Pavia University obtained by different methods: (a) FA, (b) SAE (c) DBN, (d) 2D-CNN, (e) 3D-CAE I, (f) 3D-CAE II, (g) 3D-CAE III, (h) Proposed 3D-M <sup>2</sup> CAE. . . . .	101
5.1	A DAE architecture. . . . .	107
5.2	A DAE architecture. . . . .	107
5.3	Spectral reconstruction with DAE. . . . .	108
5.4	Target detection with the proposed MSDAE. . . . .	109
5.5	Simulated scenes: (a) $simulated_a$ (the 80 <sup>th</sup> band), (b) $simulated_b$ (the 80 <sup>th</sup> band), (c) spectral signatures. . . . .	111
5.6	Comparison of different spectral curves of Target 2 in simulated images: (a) $simulated_a$ , (b) $simulated_b$ . . . . .	112
5.7	ROC curves of $simulated_a$ under different noises with SNR being 20 dB: (a) WGN, (b) MPN. . . . .	113
5.8	ROC curves of $simulated_b$ under different noises with SNR being 20 dB: (a) WGN, (b) MPN. . . . .	114
5.9	Real-world scenes: (a) $HSI_b$ (the 60 <sup>th</sup> band), (b) $HSI_c$ (the 60 <sup>th</sup> band), (c) spectral signatures. . . . .	114
5.10	Comparison of different spectral curves of Target 4 in real-world scenes: (a) $HSI_b$ , (b) $HSI_c$ . . . . .	115
5.11	ROC curves of $HSI_b$ under different noises with SNR being 20 dB: (a) WGN, (b) MPN. . . . .	116
5.12	ROC curves of $HSI_c$ under different noises with SNR being 20 dB: (a) WGN, (b) MPN. . . . .	116
5.13	Detection maps of $simulated_b$ : (a) ground truth, (b) denoised by WF, (c) reconstructed by BM3D, (d) reconstructed by DnCNN, (e) reconstructed by MSDAE. . . . .	117

5.14	Detection maps of $HSI_c$ : (a) ground truth, (b) denoised by WF, (c) denoised by BM3D, (d) denoised by DnCNN, (e) reconstructed by MSDAE. . . . .	118
5.15	Unsupervised segmentation based on 2D-CNN. . . . .	119
5.16	$HSI_c$ : (a) Composite image, (b) Segmentation map, (c) Small target regions of interest. . . . .	120
5.17	Detection maps of $HSI_a$ without unsupervised segmentation: (a) Ground truth. (b) Detection map with $\gamma=0.45$ , (c) Detection map with $\gamma=0.50$ , (d) Detection map with $\gamma=0.55$ . . . . .	121
5.18	Detection maps of $HSI_a$ with unsupervised segmentation: (a) Ground truth. (b) Detection map with $\gamma=0.40$ , (c) Detection map with $\gamma=0.45$ , (d) Detection map with $\gamma=0.5$ . . . . .	121



# List of Tables

1.1	Details of land-cover classes in Pavia University data set. . . . .	25
1.2	Details of land-cover classes in Indian Pines data set. . . . .	26
1.3	Initial network structure of 2D-CNN. . . . .	27
1.4	Network parameters of different networks. . . . .	29
1.5	OA values of Pavia University and Indian Pines under different networks. . . . .	29
1.6	OA values of Pavia University and Indian Pines under different number of units in fully connected layer. . . . .	30
1.7	OA values of Pavia University and Indian Pines under different optimizers. . . . .	31
1.8	OA values of Pavia University and Indian Pines under different number of epochs. . . . .	34
2.1	Network structure of 3D-CNN. . . . .	53
2.2	Comparison of land-cover classes and number of samples in Pavia Centre and Pavia University. . . . .	54
2.3	Comparison of land-cover classes and number of samples in Indian Pines and Salinas. . . . .	55
2.4	OA values under different noise variances of the virtual samples. . . . .	56
3.1	Network structure of 3D-CNN. . . . .	63
3.2	Architectures of the 3D-WGAN-GP. . . . .	65
3.3	Classification accuracy of Pavia University under different methods. . . . .	66
3.4	Classification accuracy of Indian Pines under different methods. . . . .	67
3.5	Classification results of different GANs combining PCA or proposed dimensionality reduction method. . . . .	68
4.1	Network structures of encoder in proposed 3D-CAE. . . . .	78



4.2	The classification accuracy of Pavia University based on different features. . . . .	79
4.3	Classification accuracy of Indian Pines based on different features. . .	81
4.4	Classification accuracy of Pavia University based on different feature extraction methods. . . . .	86
4.5	Classification accuracy of Indian Pines based on different feature extraction methods. . . . .	87
4.6	Land-cover classes and color coding in $HSI_a$ . . . . .	92
4.7	Encoder structures of three 3D-CAEs in proposed 3D-M <sup>2</sup> CAE framework. . . . .	93
5.1	The $P_d$ values of different models for $simulated_a$ and $simulated_b$ . . .	113
5.2	The $P_d$ values of different models for $HSI_b$ and $HSI_c$ . . . . .	115
5.3	Network structures of 2D-CNN. . . . .	120

# Bibliography

- [1] P WT Yuen and Mark Richardson. An introduction to hyperspectral imaging and its application for security, surveillance and target acquisition. *The Imaging Science Journal*, 58(5):241–253, 2010.
- [2] Jae-Jin Park, Sangwoo Oh, Kyung-Ae Park, Tae-Sung Kim, and Moonjin Lee. Applying hyperspectral remote sensing methods to ship detection based on airborne and ground experiments. *International Journal of Remote Sensing*, 41(15):5928–5952, 2020.
- [3] Jennifer Pontius, Mary Martin, Lucie Plourde, and Richard Hallett. Ash decline assessment in emerald ash borer-infested regions: A test of tree-level, hyperspectral technologies. *Remote Sensing of Environment*, 112(5):2665–2676, 2008.
- [4] Liang Liang, Liping Di, Lianpeng Zhang, Meixia Deng, Zhihao Qin, Shuhe Zhao, and Hui Lin. Estimation of crop lai using hyperspectral vegetation indices and a hybrid inversion method. *Remote Sensing of Environment*, 165:123–134, 2015.
- [5] Caroline M Gevaert, Juha Suomalainen, Jing Tang, and Lammert Kooistra. Generation of spectral–temporal response surfaces by combining multispectral satellite and hyperspectral uav imagery for precision agriculture applications. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):3140–3146, 2015.
- [6] Gaia Vaglio Laurin, Jonathan Cheung-Wai Chan, Qi Chen, Jeremy A Lindsell, David A Coomes, Leila Guerriero, Fabio Del Frate, Franco Miglietta, and Riccardo Valentini. Biodiversity mapping in a tropical west african forest with airborne hyperspectral data. *PloS one*, 9(6):e97910, 2014.

- [7] Gordon Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE transactions on information theory*, 14(1):55–63, 1968.
- [8] Ian T Jolliffe and BJT Morgan. Principal component analysis and exploratory factor analysis. *Statistical methods in medical research*, 1(1):69–95, 1992.
- [9] Meng Wang, Xian-Sheng Hua, Richang Hong, Jinhui Tang, Guo-Jun Qi, and Yan Song. Unified video annotation via multigraph learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(5):733–746, 2009.
- [10] Xinghao Yang, Weifeng Liu, Dapeng Tao, Jun Cheng, and Shuying Li. Multi-view canonical correlation analysis networks for remote sensing image recognition. *IEEE Geoscience and Remote Sensing Letters*, 14(10):1855–1859, 2017.
- [11] Meng Wang and Xian-Sheng Hua. Active learning in multimedia annotation and retrieval: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(2):1–21, 2011.
- [12] Jie Hu, Zhi He, Jun Li, Lin He, and Yiwen Wang. 3d-gabor inspired multiview active learning for spectral-spatial hyperspectral image classification. *Remote Sensing*, 10(7):1070, 2018.
- [13] Geunseop Lee. Fast computation of the compressive hyperspectral imaging by using alternating least squares methods. *Signal Processing: Image Communication*, 60:100–106, 2018.
- [14] Lei Wang, Jing Bai, Jiaji Wu, and Gwanggil Jeon. Hyperspectral image compression based on lapped transform and tucker decomposition. *Signal Processing: Image Communication*, 36:63–69, 2015.
- [15] Wen Yang, Xiaoshuang Yin, and Gui-Song Xia. Learning high-level features for satellite image classification with limited labeled samples. *IEEE Transactions on Geoscience and Remote Sensing*, 53(8):4472–4482, 2015.
- [16] Hanane Teffahi, Hongxun Yao, Souleyman Chaib, and Nasreddine Belabid. A novel spectral-spatial classification technique for multispectral images using extended multi-attribute profiles and sparse autoencoder. *Remote Sensing Letters*, 10(1):30–38, 2019.
- [17] Yushi Chen, Xing Zhao, and Xiuping Jia. Spectral-spatial classification of hyperspectral data based on deep belief network. *IEEE Journal of Selected*

- Topics in Applied Earth Observations and Remote Sensing*, 8(6):2381–2392, 2015.
- [18] Renlong Hang, Qingshan Liu, Danfeng Hong, and Pedram Ghamisi. Cascaded recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8):5384–5394, 2019.
- [19] Feng Zhou, Renlong Hang, Qingshan Liu, and Xiaotong Yuan. Hyperspectral image classification using spectral-spatial lstms. *Neurocomputing*, 328:39–47, 2019.
- [20] Yenan Jiang, Ying Li, and Haokui Zhang. Hyperspectral image classification based on 3-d separable resnet and transfer learning. *IEEE Geoscience and Remote Sensing Letters*, 16(12):1949–1953, 2019.
- [21] Yushi Chen, Hanlu Jiang, Chunyang Li, Xiuping Jia, and Pedram Ghamisi. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):6232–6251, 2016.
- [22] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.
- [23] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [24] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.
- [25] Jingxiang Yang, Yong-Qiang Zhao, and Jonathan Cheung-Wai Chan. Learning and transferring deep joint spectral-spatial features for hyperspectral classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(8):4729–4742, 2017.
- [26] Xudong Kang, Xuanlin Xiang, Shutao Li, and Jón Atli Benediktsson. Pca-based edge-preserving features for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12):7140–7151, 2017.

- [27] Jing Wang and Chein-I Chang. Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis. *IEEE transactions on geoscience and remote sensing*, 44(6):1586–1600, 2006.
- [28] José M Bioucas-Dias, Antonio Plaza, Gustavo Camps-Valls, Paul Scheunders, Nasser Nasrabadi, and Jocelyn Chanussot. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geoscience and remote sensing magazine*, 1(2):6–36, 2013.
- [29] Renlong Hang, Qingshan Liu, Huihui Song, and Yubao Sun. Matrix-based discriminant subspace ensemble for hyperspectral image spatial–spectral feature fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 54(2):783–794, 2015.
- [30] Renlong Hang, Qingshan Liu, Yubao Sun, Xiaotong Yuan, Hucheng Pei, Javier Plaza, and Antonio Plaza. Robust matrix discriminative analysis for feature extraction from hyperspectral images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(5):2002–2011, 2017.
- [31] Yonghao Xu, Liangpei Zhang, Bo Du, and Fan Zhang. Spectral–spatial unified networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(10):5893–5909, 2018.
- [32] Huihui Song, Qingshan Liu, Guojie Wang, Renlong Hang, and Bo Huang. Spatiotemporal satellite image fusion using deep convolutional neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(3):821–829, 2018.
- [33] Shutao Li, Weiwei Song, Leyuan Fang, Yushi Chen, Pedram Ghamisi, and Jón Atli Benediktsson. Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):6690–6709, 2019.
- [34] Mercedes E Paoletti, Juan Mario Haut, Javier Plaza, and Antonio Plaza. A new deep convolutional neural network for fast hyperspectral image classification. *ISPRS journal of photogrammetry and remote sensing*, 145:120–147, 2018.
- [35] Stevica Cvetković, Miloš B Stojanović, and Saša V Nikolić. Multi-channel descriptors and ensemble of extreme learning machines for classification of

- remote sensing images. *Signal Processing: Image Communication*, 39:111–120, 2015.
- [36] Fan Zhao, Guizhong Liu, and Xing Wang. An efficient macroblock-based diverse and flexible prediction modes selection for hyperspectral images coding. *Signal Processing: Image Communication*, 25(9):697–708, 2010.
- [37] MI Vakil, DB Megherbi, and JA Malas. A robust multi-stage information-theoretic approach for registration of partially overlapped hyperspectral aerial imagery and evaluation in the presence of system noise. *Signal Processing: Image Communication*, 52:97–110, 2017.
- [38] Zhongling Huang, Zongxu Pan, and Bin Lei. Transfer learning with deep convolutional neural network for sar target classification with limited labeled data. *Remote Sensing*, 9(9):907, 2017.
- [39] Yushi Chen, Zhouhan Lin, Xing Zhao, Gang Wang, and Yanfeng Gu. Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected topics in applied earth observations and remote sensing*, 7(6):2094–2107, 2014.
- [40] Shaohui Mei, Xin Yuan, Jingyu Ji, Yifan Zhang, Shuai Wan, and Qian Du. Hyperspectral image spatial super-resolution via 3d full convolutional neural network. *Remote Sensing*, 9(11):1139, 2017.
- [41] Weifeng Liu, Zheng-Jun Zha, Yanjiang Wang, Ke Lu, and Dacheng Tao.  $p$ -laplacian regularized sparse coding for human activity recognition. *IEEE Transactions on Industrial Electronics*, 63(8):5120–5129, 2016.
- [42] Weifeng Liu, Hongli Liu, Dapeng Tao, Yanjiang Wang, and Ke Lu. Manifold regularized kernel logistic regression for web image annotation. *Neurocomputing*, 172:3–8, 2016.
- [43] Meiting Yu, Ganggang Dong, Haiyan Fan, and Gangyao Kuang. Sar target recognition via local sparse representation of multi-manifold regularized low-rank approximation. *Remote Sensing*, 10(2):211, 2018.
- [44] Pierluigi Casale, Marco Altini, and Oliver Amft. Transfer learning in body sensor networks using ensembles of randomized trees. *IEEE Internet of Things Journal*, 2(1):33–40, 2015.

- [45] Jianzhe Lin, Chen He, Z Jane Wang, and Shuying Li. Structure preserving transfer learning for unsupervised hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 14(10):1656–1660, 2017.
- [46] Ying Zhan, Dan Hu, Yuntao Wang, and Xianchuan Yu. Semisupervised hyperspectral image classification based on generative adversarial networks. *IEEE Geoscience and Remote Sensing Letters*, 15(2):212–216, 2017.
- [47] Mingyang Zhang, Maoguo Gong, Yishun Mao, Jun Li, and Yue Wu. Unsupervised feature extraction in hyperspectral images based on wasserstein generative adversarial network. *IEEE Transactions on Geoscience and Remote Sensing*, 57(5):2669–2688, 2018.
- [48] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.
- [49] Eftychios Protopapadakis, Anastasios Doulamis, Nikolaos Doulamis, and Evangelos Maltezos. Stacked autoencoders driven by semi-supervised learning for building extraction from near infrared remote sensing imagery. *Remote Sensing*, 13(3):371, 2021.
- [50] Shaohui Mei, Jingyu Ji, Yunhao Geng, Zhi Zhang, Xu Li, and Qian Du. Unsupervised spatial–spectral feature learning by 3d convolutional autoencoder for hyperspectral classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):6808–6820, 2019.
- [51] Chao Tao, Hongbo Pan, Yansheng Li, and Zhengrou Zou. Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. *IEEE Geoscience and remote sensing letters*, 12(12):2438–2442, 2015.
- [52] Xiangrong Zhang, Yanjie Liang, Chen Li, Ning Huyan, Licheng Jiao, and Huiyu Zhou. Recursive autoencoders-based unsupervised feature learning for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 14(11):1928–1932, 2017.

- [53] Xuefeng Liu, Salah Bourennane, and Caroline Fossati. Reduction of signal-dependent noise from hyperspectral images for target detection. *IEEE Transactions on Geoscience and Remote Sensing*, 52(9):5396–5411, 2013.
- [54] Nadine Renard and Salah Bourennane. Improvement of target detection methods by multiway filtering. *IEEE Transactions on Geoscience and Remote Sensing*, 46(8):2407–2417, 2008.
- [55] Guangyi Chen and Shen-En Qian. Denoising and dimensionality reduction of hyperspectral imagery using wavelet packets, neighbour shrinking and principal component analysis. *International Journal of Remote Sensing*, 30(18):4889–4895, 2009.
- [56] Xianjie Zha, Rongshan Fu, Zhiyang Dai, and Bin Liu. Noise reduction in interferograms using the wavelet packet transform and wiener filtering. *IEEE Geoscience and Remote Sensing Letters*, 5(3):404–408, 2008.
- [57] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.
- [58] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. Deep learning on image denoising: An overview. *Neural Networks*, 2020.
- [59] Tao Lin, Julien Marot, and Salah Bourennane. Small target detection improvement in hyperspectral image. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 460–469. Springer, 2013.
- [60] Qiang Zhang, Qiangqiang Yuan, Jie Li, Fujun Sun, and Liangpei Zhang. Deep spatio-spectral bayesian posterior for hyperspectral image non-iid noise removal. *ISPRS Journal of Photogrammetry and Remote Sensing*, 164:125–137, 2020.
- [61] Lovedeep Gondara. Medical image denoising using convolutional denoising autoencoders. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 241–246. IEEE, 2016.
- [62] Kyunghyun Cho. Simple sparsification improves sparse denoising autoencoders in denoising highly corrupted images. In *International conference on machine learning*, pages 432–440. PMLR, 2013.



- [63] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.
- [64] Qiangqiang Yuan, Qiang Zhang, Jie Li, Huanfeng Shen, and Liangpei Zhang. Hyperspectral image denoising employing a spatial–spectral deep residual convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):1205–1218, 2018.
- [65] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2018.
- [66] Pedram Ghamisi, Bernhard Höfle, and Xiao Xiang Zhu. Hyperspectral and lidar data fusion using extinction profiles and deep convolutional neural network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(6):3011–3024, 2016.
- [67] Xichuan Zhou, Shengli Li, Fang Tang, Kai Qin, Shengdong Hu, and Shujun Liu. Deep learning with grouped features for spatial spectral classification of hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 14(1):97–101, 2016.
- [68] Moussa Amrani and Feng Jiang. Deep feature extraction and combination for synthetic aperture radar target classification. *Journal of Applied Remote Sensing*, 11(4):042616, 2017.
- [69] Pedram Ghamisi, Yushi Chen, and Xiao Xiang Zhu. A self-improving convolution neural network for the classification of hyperspectral data. *IEEE Geoscience and Remote Sensing Letters*, 13(10):1537–1541, 2016.
- [70] Lichao Mou, Pedram Ghamisi, and Xiao Xiang Zhu. Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3639–3655, 2017.
- [71] Wei Li, Guodong Wu, Fan Zhang, and Qian Du. Hyperspectral image classification using deep pixel-pair features. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):844–853, 2016.

- [72] Xuefeng Liu, Qiaoqiao Sun, Yue Meng, Min Fu, and Salah Bourennane. Hyperspectral image classification based on parameter-optimized 3d-cnns combined with transfer learning and virtual samples. *Remote Sensing*, 10(9):1425, 2018.
- [73] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.
- [74] Ce Zheng, Xue Jiang, and Xingzhao Liu. Generalized synthetic aperture radar automatic target recognition by convolutional neural network with joint use of two-dimensional principal component analysis and support vector machine. *Journal of Applied Remote Sensing*, 11(4):046007, 2017.
- [75] Abhishek Agarwal, Tarek El-Ghazawi, Hesham El-Askary, and Jacqueline Le-Moigne. Efficient hierarchical-pca dimension reduction for hyperspectral imagery. In *2007 IEEE International Symposium on Signal Processing and Information Technology*, pages 353–356. IEEE, 2007.
- [76] Xuefeng Liu, Salah Bourennane, and Caroline Fossati. Denoising of hyperspectral images using the parafac model and statistical performance analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 50(10):3717–3724, 2012.
- [77] Caglar Gulcehre, Marcin Moczulski, Misha Denil, and Yoshua Bengio. Noisy activation functions. In *International conference on machine learning*, pages 3059–3068. PMLR, 2016.
- [78] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.
- [79] Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2559–2566. IEEE, 2010.
- [80] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- [81] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

- [82] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
- [83] Hagai Attias. Independent factor analysis. *Neural computation*, 11(4):803–851, 1999.
- [84] Yan Liu, Shusen Zhou, and Qingcai Chen. Discriminative deep belief networks for visual data classification. *Pattern Recognition*, 44(10-11):2287–2296, 2011.
- [85] Ahmed M Abdel-Zaher and Ayman M Eldeib. Breast cancer classification using deep belief networks. *Expert Systems with Applications*, 46:139–144, 2016.
- [86] Jiaojiao Li, Bobo Xi, Yunsong Li, Qian Du, and Keyan Wang. Hyperspectral classification based on texture feature enhancement and deep belief networks. *Remote Sensing*, 10(3):396, 2018.
- [87] Jingxiang Yang, Yongqiang Zhao, Jonathan Cheung-Wai Chan, and Chen Yi. Hyperspectral image classification using two-channel deep convolutional neural network. In *2016 IEEE international geoscience and remote sensing symposium (IGARSS)*, pages 5079–5082. IEEE, 2016.
- [88] Zilong Zhong, Jonathan Li, Zhiming Luo, and Michael Chapman. Spectral-spatial residual network for hyperspectral image classification: A 3-d deep learning framework. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2):847–858, 2017.
- [89] Pedram Ghamisi, Javier Plaza, Yushi Chen, Jun Li, and Antonio J Plaza. Advanced spectral classifiers for hyperspectral images: A review. *IEEE Geoscience and Remote Sensing Magazine*, 5(1):8–32, 2017.
- [90] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.
- [91] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

- [92] Uzma Sharif, Zahid Mehmood, Toqeer Mahmood, Muhammad Arshad Javid, Amjad Rehman, and Tanzila Saba. Scene analysis and search using local features and support vector machine for effective content-based image retrieval. *Artificial Intelligence Review*, 52(2):901–925, 2019.
- [93] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing*, 1(3):244–256, 1972.
- [94] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2):112–122, 1973.
- [95] Xiaorui Ma, Anyan Fu, Jie Wang, Hongyu Wang, and Baocai Yin. Hyperspectral image classification based on deep deconvolution network with skip architecture. *IEEE Transactions on Geoscience and Remote Sensing*, 56(8):4781–4791, 2018.
- [96] Chein-I Chang, Qian Du, Tzu-Lung Sun, and Mark LG Althouse. A joint band prioritization and band-decorrelation approach to band selection for hyperspectral image classification. *IEEE transactions on geoscience and remote sensing*, 37(6):2631–2641, 1999.
- [97] Chunhong Liu, Chunhui Zhao, and LY Zhang. A new method of hyperspectral remote sensing image dimensional reduction. *Journal of Image and Graphics*, 10(2):218–222, 2005.
- [98] PS Chavez, Graydon L Berlin, and Lynda B Sowers. Statistical method for selecting landsat mss. *J. Appl. Photogr. Eng*, 8(1):23–30, 1982.
- [99] Kang Sun, Xiurui Geng, Luyan Ji, and Yun Lu. A new band selection method for hyperspectral image based on data quality. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2697–2703, 2014.
- [100] Chi Wang, Maoguo Gong, Mingyang Zhang, and Yongqiang Chan. Unsupervised hyperspectral image band selection via column subset selection. *IEEE Geoscience and Remote Sensing Letters*, 12(7):1411–1415, 2015.
- [101] Kasthurirangan Gopalakrishnan, Siddhartha K Khaitan, Alok Choudhary, and Ankit Agrawal. Deep convolutional neural networks with transfer learning for

- computer vision-based data-driven pavement distress detection. *Construction and building materials*, 157:322–330, 2017.
- [102] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.
- [103] Julia R Fielding, Lee A Fox, Howard Heller, Steven E Seltzer, Clare M Tempany, Stuart G Silverman, and Graeme Steele. Spiral ct in the evaluation of flank pain: overall accuracy and feature analysis. *Journal of computer assisted tomography*, 21(4):635–638, 1997.
- [104] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.
- [105] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *stat*, 1050.
- [106] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.
- [107] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.
- [108] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.
- [109] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.
- [110] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.
- [111] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

- [112] Peicheng Zhou, Junwei Han, Gong Cheng, and Baochang Zhang. Learning compact and discriminative stacked autoencoder for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(7):4823–4833, 2019.
- [113] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.
- [114] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984.
- [115] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- [116] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.
- [117] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.
- [118] Zhen Zuo, Bing Shuai, Gang Wang, Xiao Liu, Xingxing Wang, Bing Wang, and Yushi Chen. Learning contextual dependence with convolutional hierarchical recurrent neural networks. *IEEE Transactions on Image Processing*, 25(7):2983–2996, 2016.
- [119] Lin Zhu, Yushi Chen, Pedram Ghamisi, and Jón Atli Benediktsson. Generative adversarial networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(9):5046–5063, 2018.
- [120] Miao Kang, Kefeng Ji, Xiangguang Leng, Xiangwei Xing, and Huanxin Zou. Synthetic aperture radar target recognition with feature fusion based on a stacked autoencoder. *Sensors*, 17(1):192, 2017.

- [121] Peng Liang, Wenzhong Shi, and Xiaokang Zhang. Remote sensing image classification based on stacked denoising autoencoder. *Remote Sensing*, 10(1):16, 2018.
- [122] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.
- [123] Su May Hsu and Hsiao-hua Burke. Multisensor fusion with hyperspectral imaging data: detection and classification. In *Handbook of Pattern Recognition and Computer Vision*, pages 347–364. World Scientific, 2005.
- [124] Anhar Risnumawan, Muhammad Ilham Perdana, Alif Habib Hidayatulloh, A Khoirul Rizal, and Indra Adji Sulistijono. Towards an automatic aircraft wreckage detection using a monocular camera of uav. In *2019 International Electronics Symposium (IES)*, pages 501–504. IEEE, 2019.
- [125] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [126] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [127] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.
- [128] Nicola Acito, Marco Diani, and Giovanni Corsini. Signal-dependent noise modeling and model parameter estimation in hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 49(8):2957–2971, 2011.
- [129] Mikhail L Uss, Benoit Vozel, Vladimir V Lukin, and Kacem Chehdi. Local signal-dependent noise variance estimation from hyperspectral textural images. *IEEE Journal of Selected Topics in Signal Processing*, 5(3):469–486, 2011.

- [130] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.
- [131] Le Sun, Zebin Wu, Jianjun Liu, Liang Xiao, and Zhihui Wei. Supervised spectral–spatial hyperspectral image classification with weighted markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, 53(3):1490–1503, 2014.
- [132] Vitor F Haertel and Yosio Edemir Shimabukuro. Spectral linear mixing model in low spatial resolution image data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(11):2555–2562, 2005.
- [133] Alessandro Maffei, Juan M Haut, Mercedes Eugenia Paoletti, Javier Plaza, Lorenzo Bruzzone, and Antonio Plaza. A single model cnn for hyperspectral image denoising. *IEEE Transactions on Geoscience and Remote Sensing*, 58(4):2516–2529, 2019.
- [134] Pengfei Shan. Image segmentation method based on k-mean algorithm. *EURASIP Journal on Image and Video Processing*, 2018(1):1–9, 2018.
- [135] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.
- [136] Asako Kanezaki. Unsupervised image segmentation by backpropagation. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1543–1547. IEEE, 2018.
- [137] Wonjik Kim, Asako Kanezaki, and Masayuki Tanaka. Unsupervised learning of image segmentation based on differentiable feature clustering. *IEEE Transactions on Image Processing*, 29:8055–8068, 2020.



## Résumé

Cette thèse est consacrée à l'analyse et au traitement d'images hyperspectrales principalement avec des modèles d'apprentissage en profondeur. Pour exploiter pleinement les informations spectrales et spatiales des données hyperspectrales, un réseau neuronal convolutif avec réglage des paramètres est proposé pour la classification hyperspectrale. En outre, pour résoudre le problème des échantillons étiquetés limités dans les images hyperspectrales, des méthodes d'extraction de caractéristiques non supervisées basées sur un réseau antagoniste génératif amélioré et un autoencodeur convolutif sont étudiées. De plus, un cadre d'autoencodeur de débruitage multi-échelle est conçu pour le débruitage et l'amélioration de la détection de cibles. Les résultats sur des données simulées et réelles montrent l'efficacité des méthodes proposées et leurs perspectives prometteuses pour les applications en imagerie hyperspectrale.

**Mots clés:** Classification, détection, extraction de caractéristiques, apprentissage non supervisé, apprentissage profond, imagerie hyperspectrale.

## Abstract

This thesis is devoted to analyzing and processing hyperspectral images mainly with deep learning methods. To fully exploit the spectral-spatial information of hyperspectral data, convolutional neural network with parameter tuning is proposed for hyperspectral classification. Besides, to solve the problem of limited labeled samples in hyperspectral images, unsupervised feature extraction methods based on improved generative adversarial network and convolutional autoencoder are investigated. In addition, a multi-scale denoising autoencoder framework is designed for denoising and improvements of target detection. The results on simulated and real-world data demonstrate that the effectiveness of the proposed methods and their promising prospects in hyperspectral imaging applications.

**Keywords:** Classification, detection, feature extraction, unsupervised learning, deep learning, hyperspectral image.