

A REFINED ENCODER–DECODER CAPSULE NETWORK FOR FINE-GRAINED CLASSIFICATION OF MULTI-CROP AGRICULTURAL SCENES

RÉSEAU DE CAPSULES ENCODEUR–DÉCODEUR AMÉLIORÉ POUR LA CLASSIFICATION FINE DE SCÈNES AGRICOLES MULTICULTURES

Wei Zhang¹, Yupu He^{2*}, Wei Qi³, Zijun Mai⁴, Renjing Ji⁵, Nan Gu⁶,
Wentong Bi⁷, and Yuanyuan Zha⁸

ABSTRACT

Precision agricultural monitoring depends on accurately distinguishing complex crop field patterns. Although the combination of hyperspectral imagery (HSI) and convolutional neural networks (CNNs) has become a popular approach for land cover classification, conventional CNNs still struggle to capture the joint spatial–spectral characteristics of intricate agricultural scenes due to their reliance on scalar neurons. To fill this gap, this paper presents the Agricultural Capsule Network (AgriCapsNet), an encoder–decoder capsule framework designed for fine-grained multi-crop classification with HSI. Within the encoder, a lightweight attention mechanism adaptively enhances discriminative features, while vector-based capsule layers model hierarchical part-whole relationships. The decoder is optimized through a joint formulation of margin loss and reconstruction loss, which strengthens the model's generalization ability. Experimental evaluation on a crop-rich HSI dataset demonstrates that AgriCapsNet surpasses nine existing methods, yielding an overall accuracy improvement ranging from 1.87% to 28.82%.

Keywords: Capsule network; attention mechanism; encoder–decoder; hyperspectral imagery; fine-grained crop mapping.

RÉSUMÉ

La surveillance agricole de précision repose sur la capacité à distinguer avec exactitude des configurations complexes de parcelles cultivées. Bien que l'association de l'imagerie hyperspectrale (HSI) et des réseaux neuronaux convolutifs (CNN) soit devenue une approche courante de la classification de l'occupation du sol, les CNN classiques peinent encore à saisir les caractéristiques spatiales et spectrales conjointes de scènes agricoles complexes, en raison de leur recours à des neurones scalaires. Pour combler cette lacune, cet article présente l'Agricultural Capsule Network (AgriCapsNet), un cadre capsulaire encodeur–décodeur conçu pour la classification fine de cultures multiples à partir d'HSI. Dans l'encodeur, un mécanisme d'attention léger renforce de manière adaptative les caractéristiques discriminantes, tandis que des couches de capsules vectorielles modélisent les relations hiérarchiques partie–tout. Le décodeur est optimisé au moyen d'une formulation conjointe de la perte de marge et de la perte de reconstruction, ce qui renforce la capacité de généralisation du modèle. Une évaluation expérimentale sur un jeu de données HSI riche en

1 Wei Zhang, senior engineer, Nanjing Hydraulic Research Institute, China, zhangwei@nhri.cn.

2 Yupu He, Professorate Senior Engineer, Nanjing Hydraulic Research Institute, China, heyupu28@163.com, corresponding author.

3 Wei Qi, senior engineer, Nanjing Hydraulic Research Institute, China, wqi@nhri.cn.

4 Zijun Mai, engineer, Nanjing Hydraulic Research Institute, China, zjmai@nhri.cn.

5 Renjing Ji, senior engineer, Nanjing Hydraulic Research Institute, China, E-mail: jirenjing@126.com.

6 Nan Gu, senior engineer, Nanjing Hydraulic Research Institute, China, nanguskr@163.com.

7 Wentong Bi, engineer, Nanjing Hydraulic Research Institute, China, wentongbi9867@163.com.

8 Yuanyuan Zha, Professor, Wuhan University, China, zhayuan87@whu.edu.cn.

cultures montre qu'AgriCapsNet surpasse neuf méthodes existantes, avec une amélioration de la précision globale comprise entre 1,87 % et 28,82 %.

Keywords: Réseau de capsules; Mécanisme d'attention; Encodeur-décodeur; Imagerie hyperspectrale; Cartographie des cultures à grain fin.

1. Introduction

Accurate and timely agricultural information is essential for food security^[1]. In this context, rapid and reliable classification of crop cultivation areas is critical for advancing sustainable development. However, complex agricultural landscapes present considerable challenges. Moreover, external disturbances such as soil background variations and weed infestation further obscure spectral signatures, elevating intra-class variability while diminishing inter-class separability.

Hyperspectral imagery (HSI) holds considerable promise for agricultural classification owing to its fine spectral resolution^[2]. Nevertheless, practical HSI applications are often constrained by data redundancy and the Hughes phenomenon, which erode accuracy and limit scalability^[3]. To mitigate these challenges, prior research has concentrated on three fronts: data quality enhancement (e.g., preprocessing, sample libraries), feature optimization (e.g., spatial/frequency operators, sparse coding, deep features), and classifier refinement (e.g., improved machine learning, ensembles, deep networks)^[4,5].

Deep learning has recently propelled advances in HSI-based agricultural classification. Widely adopted architectures include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Deep Belief Networks (DBNs). Among them, CNN variants (1D, 2D, 3D) are particularly adept at spatial feature extraction^[6]. Nonetheless, RNNs and DBNs typically require vectorizing spatial inputs, which disrupts joint spatial-spectral information. Conventional CNNs built on scalar neurons also exhibit limited capacity for modeling complex, rotationally invariant features.

Capsule networks (CapsNet), pioneered by Hinton et al.^[7,8], employ vectorized neurons and dynamic routing to encode hierarchical spatial relationships and instantiation parameters. In HSI classification, CapsNet has demonstrated competitive results through adaptive designs and local optimization^[9]. However, existing capsule networks are not purpose-built for agricultural environments; they lack dedicated mechanisms to suppress background clutter and emphasize discriminative spectral-spatial patterns, leading to suboptimal feature extraction in the presence of mixed pixels and diverse crop types. Moreover, standard capsule layers often struggle to adequately model hierarchical spatial structures in fragmented fields, limiting class discriminability and cross-environment generalization.

To overcome these limitations, this paper proposes the Agricultural Capsule Network (AgriCapsNet), an encoder-decoder capsule architecture for fine-grained classification of cultivated areas using HSI. The encoder incorporates attention mechanisms and lightweight convolutions to strengthen spectral-spatial feature extraction, while the capsule layers enhance hierarchical representation. The decoder adopts a joint margin-reconstruction loss to improve classification generalization. Compared to baselines, AgriCapsNet achieves superior accuracy in multi-crop agricultural landscapes.

2. Methodology

2.1 Overview of the Proposed Model

As shown in Figure 1, AgriCapsNet adopts an encoder-decoder framework. The encoder integrates a feature-focused former with capsule layers (PrimaryCaps and DigitCaps) to strengthen spatial-spectral feature extraction, thereby boosting the network's attention and representation capacity. The decoder addresses cultivated land cover identification by simultaneously performing reconstruction and classification. Joint optimization of these two loss functions effectively improves the model's generalization ability.

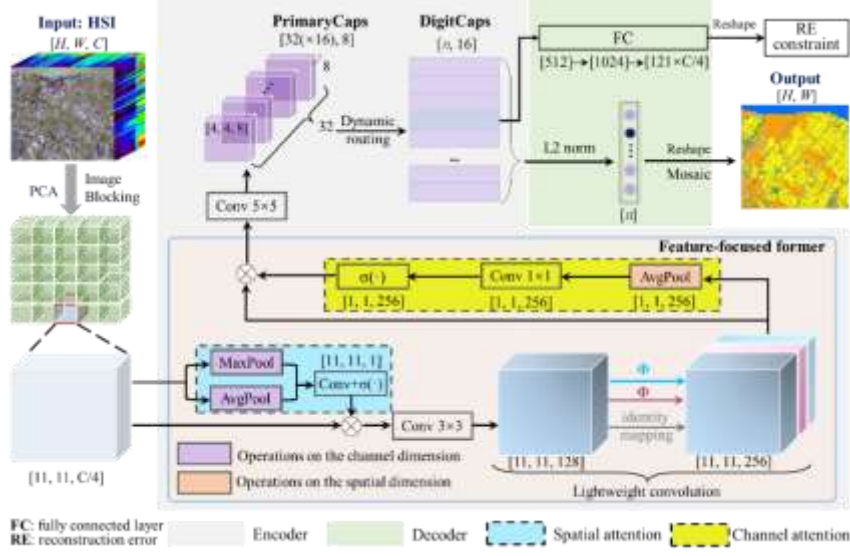


Figure 1. Schematic illustration of the AgriCapsNet architecture.

2.2 Feature-focused Former

This encoder component combines convolutional operators, a lightweight convolution module, and attention mechanisms to strengthen feature extraction.

2.2.1 Convolutional Operator

Standard convolution relies on learnable kernels to transform inputs and extract multi-scale patterns. For a given layer, the per-channel pre-activation is obtained by summing the convolution results of kernels over selected input feature maps and adding a bias term (i.e., kernel-wise summation with bias), which is then passed through a nonlinear activation function^[10] to produce the final activation output (see Eq. (1)).

$$\begin{cases} u_{\beta}^l = f(v_{\beta}^l) \\ v_{\beta}^l = \sum_{\alpha \in A_{\beta}} u_{\alpha}^{l-1} * \kappa_{\alpha\beta}^l + b_{\beta}^l \end{cases} \quad (1)$$

Here, u_{β}^l is the activation output of the β -th channel in the l -th convolutional layer, and $f(\cdot)$ is an activation function (e.g., Tanh, Sigmoid, or ReLU). The pre-activation feature v_{β}^l is computed by the sum of convolution operations between the previous layer's feature maps u_{α}^{l-1} and the corresponding convolution kernel $\kappa_{\alpha\beta}^l$, plus a bias b_{β}^l . The subset A_{β} indicates which input channels contribute to the β -th output, and $*$ denotes the convolution operator.

2.2.2 Lightweight Convolution Module

To cut computational cost while retaining representational diversity, a Ghost-module-inspired lightweight design is adopted^[11]. It first applies a standard convolution to produce intrinsic features (Eq. (2)), then expands them via efficient linear operations (Eq. (3)), yielding rich outputs with notably fewer resources.

$$Y = u * \kappa \quad (2)$$

$$y_{ij} = \Phi_{i,j}(y_i), \forall i = 1, \dots, m, j = 1, \dots, s \quad (3)$$

Here, Y is the intrinsic feature map from the standard convolution, u denotes the input feature, and the convolution kernel $k \in \mathbb{R}^{m \times k \times k \times c}$, $\Phi_{i,j}$ denotes the j -th linear operation on the i -th intrinsic feature y_i of Y (with $\Phi_{i,s}$ as the identity mapping). The final feature set is $\{y_{11}, y_{12}, \dots, y_{ms}\}$, where the total output channels $n=m \cdot s$ ($m \leq n$, m is the number of intrinsic features, s is the number of linear operations). Excluding the last identity mapping, the relationship becomes $m(s-1) = n/s \cdot (s-1)$.

2.2.3 Attention Mechanisms

Spatial attention is computed by pooling and convolving the concatenated channel-wise statistics (Eq. (4)):

$$\mathbf{O}_s = \sigma(\text{Conv}([\text{AvgPool}(\mathbf{F}); \text{MaxPool}(\mathbf{F})])) \quad (4)$$

Where $\mathbf{F}$ is the input, $\text{AvgPool}(\cdot)$ and $\text{MaxPool}(\cdot)$ are average/max pooling. The convolution operation $\text{Conv}(\cdot)$ uses a 7×7 kernel, $\sigma(\cdot)$ is Sigmoid activation function, and $\mathbf{O}_s$ denotes spatial attention weights.

Channel attention adopts an adaptive 1-D convolution to model local cross-channel interactions without dimensionality reduction (Eqs. (5)–(6)):

$$\boldsymbol{\omega} = \sigma(\text{C1D}_k(\mathbf{y})) \quad (5)$$

Here, y is the input channel feature, $\text{C1D}(\cdot)$ denotes 1-D convolution, $\boldsymbol{\omega}$ are channel attention weights, and the kernel size k is adaptively determined from the channel number C :

$$k = \psi(C) = \left\lfloor \frac{\log(C)}{\gamma} + \frac{b}{\gamma} \right\rfloor_{\text{odd}} \quad (6)$$

Here, the parameters are set as $b=1$, $\gamma=2$.

2.3 Capsule Layer

The capsule layer comprises PrimaryCaps, which convert convolutional features into vector-form capsules, and DigitCaps, which employ dynamic routing to produce high-level capsules that encode per-class semantic attributes. Information flows between capsules through three operations: affine transformation, weighted summation, and a squash nonlinearity. The routing mechanism adaptively updates coupling coefficients to selectively pass informative features from low-level capsules to higher-level ones^[8].

The affine transformation captures spatial relationships between capsules across layers by multiplying each lower-level capsule $\mathbf{u}_i$ with a weight matrix $\mathbf{W}_{ij}$ ($i=1, 2, 3, \dots$), yielding higher-level features $\mathbf{U}_{ji}$. These features are then aggregated via a weighted sum with coupling coefficients c_{ij} ($i=1, 2, 3, \dots$), producing the input vector $\hat{\mathbf{v}}_j$ for the next-level capsule:

$$\begin{cases} \mathbf{U}_{ji} = \mathbf{W}_{ij} \mathbf{u}_i \\ \hat{\mathbf{v}}_j = \sum_i c_{ij} \mathbf{U}_{ji} \end{cases} \quad (7)$$

A squash function is applied to $\hat{\mathbf{v}}_j$ to compress its magnitude into (0, 1) without altering its orientation. The resulting capsule $\mathbf{v}_j$ thus has a length that can be interpreted as the existence probability of the corresponding entity:

$$\mathbf{v}_j = \frac{\|\hat{\mathbf{v}}_j\|^2}{1 + \|\hat{\mathbf{v}}_j\|^2} \frac{\hat{\mathbf{v}}_j}{\|\hat{\mathbf{v}}_j\|} \quad (8)$$

Here, $\|\cdot\|$ denotes the L2 norm, i.e., the magnitude of the vector, and $\|\cdot\|^2$ represents the square of the L2 norm. The first part of the formula performs the squashing of the vector magnitude, while the latter part preserves the direction of the vector. Overall, the operation normalizes the vector while maintaining its original direction.

Dynamic routing iteratively refines the coupling coefficients c_{ij} to control how much information flows from lower-level capsules to higher-level ones. Starting from an initial logit b set to zero, each iteration performs softmax normalization, computes a weighted sum of the transformed features, applies the squash activation, and updates b by adding the agreement (inner product) between the output capsule and the prediction $\mathbf{U}_{ji}$. A larger agreement increases the corresponding c_{ij} , so informative lower-level features are routed more strongly to the appropriate higher-level capsule.

2.4 Decoder and Loss Function

The decoder is built with fully connected layers whose dimensions transition as $[512] \rightarrow [1024] \rightarrow [\text{patch size} \times \text{patch size} \times C/2]$. It reconstructs data through a linear mapping followed by a reshape operation:

$$\mathbf{O}_r = \text{Reshape}(\mathbf{W}\mathbf{x}) \quad (9)$$

Where $\mathbf{x}$ is the input vector, $\mathbf{W}$ is the weight matrix, $\text{Reshape}(\cdot)$ rearranges the result into the target output shape. The training objective consists of a margin loss combined with a reconstruction regularizer. The reconstruction loss is simply the sum of squared errors between the original input $\mathbf{I}$ and the reconstructed output $\mathbf{O}_r$. The complete loss function is given by:

$$\begin{cases} L_c = T_c \max(0, m^+ - \|\mathbf{v}_c\|)^2 + \lambda(1 - T_c) \max(0, \|\mathbf{v}_c\| - m^-)^2 \\ L_{\text{total}} = \sum_c L_c + \delta \|\mathbf{I} - \mathbf{O}_r\|^2 \end{cases} \quad (10)$$

Here, L_c denotes the margin loss for class c , with the indicator T_c being 1 for the true class and 0 otherwise. The upper and lower margin thresholds are set to $m^+=0.9$ and $m^-=0.1$, which respectively penalize incorrect positive predictions and missed detections. The sparsity coefficient λ equals 0.5, and the reconstruction weight δ is fixed at 2×10^{-5} , L_{total} denotes the total loss.

3. Experiments and Analysis

3.1 Dataset

This study employs the WHU-Hi-HongHu (WHH) hyperspectral dataset^[12], which is publicly available within the remote sensing community. Collected by a Headwall Nano-Hyperspec sensor over Honghu City, Hubei Province, in December 2017, the image consists of 940×475 pixels at a spatial resolution of 0.043 m and covers 270 spectral bands. It includes 22 land cover classes, 17 of which are cash crops. For training, 100 samples per class were randomly chosen. The class names and sample details are listed in Table 2, with a corresponding illustration in Figure 2.

Table 2. Land cover categories and sample information for the WHH dataset.

Dataset	Code	Class	Train	Test	Total
WHH	1	Red roof	100	13941	14041
	2	Road	100	3412	3512
	3	Bare soil	100	21721	21821
	4	Cotton	100	163185	163285
	5	Cotton firewood	100	6118	6218
	6	Rape	100	44457	44557
	7	Chinese cabbage	100	24003	24103
	8	Pakchoi	100	3954	4054

Dataset	Code	Class	Train	Test	Total
	9	Cabbage	100	10719	10819
	10	Tuber mustard	100	12294	12394
	11	Brassica parachinensis	100	10915	11015
	12	Brassica chinensis	100	8854	8954
	13	Small brassica chinensis	100	22407	22507
	14	Lactuca sativa	100	7256	7356
	15	Celtuce	100	902	1002
	16	Film covered lettuce	100	7162	7262
	17	Romaine lettuce	100	2910	3010
	18	Carrot	100	3117	3217
	19	White radish	100	8612	8712
	20	Garlic sprout	100	3386	3486
	21	Broad bean	100	1228	1328
	22	Tree	100	3940	4040
Total			2200	384493	386693

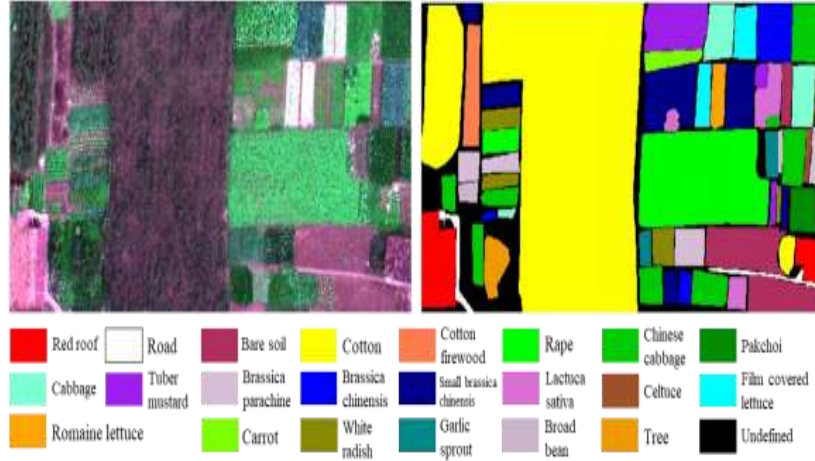


Figure 2. Visualization of the WHH hyperspectral image.

3.2 Experimental Configuration

All experiments ran on a workstation with 128 GB RAM, a 10-core Intel Core i9-9900X CPU, and an NVIDIA GeForce RTX 2080Ti GPU. The reported results (mean $\pm$ standard deviation) are based on 10 independent runs per dataset. Evaluation metrics include per-class accuracy, Overall Accuracy (OA), Average Accuracy (AA), and the Kappa coefficient (κ).

3.3 Parameter Settings and Comparative Models

The configuration of AgriCapsNet covers training hyperparameters and network structure. Training uses the Adam optimizer with an initial learning rate of 0.001 and an exponential decay schedule (decay factor 0.8 per epoch). The total number of epochs is 20, the batch size is 64, and the default patch size is 11 $\times$ 11. In the encoder, the lightweight convolution module first produces 128 intrinsic features and finally outputs 256-dimensional feature maps. Following the common guideline that DigitCaps capsule dimension should be twice that of PrimaryCaps, PrimaryCaps employs 32 convolutional units with capsule dimension 8, while DigitCaps capsules have dimension 16.

The convolution kernel size of the lightweight convolution module was determined through a comparative study. Three kernel sizes were tested: 3×3 (Scheme A), 5×5 (Scheme B), and 7×7 (Scheme C). The optimal scheme for the WHH dataset was found to be B. The models used for comparison include SVM^[13], RF^[14], LSTM^[15], 1D-CNN^[16], Vision Transformer (ViT)^[17], MSCNN^[15], 3D-CNN^[18], SSUN^[15], and CapsNet-Base^[8].

3.4 Comparison with the State of the Arts

The classification performance of all models was evaluated on the WHH dataset, and the quantitative results are summarized in Table 3. AgriCapsNet achieves the best overall performance, with an OA of 95.83%, outperforming the other models by 1.87%–28.82%. For AA and κ , the improvements range from 2.07% to 31.98% and from 0.0235 to 0.3405, respectively. The model also obtains the highest per-class accuracy in most categories, including a perfect 100% for broad bean. The classification maps in Figure 3 further indicate that AgriCapsNet significantly suppresses speckle noise and delineates crop boundaries more clearly, especially for challenging classes such as rape. MSCNN, SSUN, and CapsNet-Base also show competitive results (OA > 92%), benefiting from their multi-scale feature extraction and hybrid network designs.

Table 3. Quantitative classification performance of the compared methods on the WHH

Class	RF	SVM	LSTM	1D-CNN	ViT	MSCNN	3D-CNN	SSUN	CapsNet-Base	AgriCapsNet
Red roof	80.54±1.74	90.24±0.47	89.02±1.41	89.86±1.19	95.95±0.62	95.69±1.08	93.52±0.88	96.77±1.49	98.11±0.92	98.14±0.90
Road	81.17±2.15	81.49±0.94	84.14±1.79	86.94±1.75	93.42±0.55	95.19±1.64	94.77±1.50	94.87±1.74	98.10±0.80	98.37±0.64
Bare soil	68.43±1.49	70.79±1.40	70.13±4.08	71.39±2.01	90.90±0.72	93.13±2.43	87.77±2.07	92.58±1.59	90.34±2.74	93.22±2.03
Cotton	72.11±1.55	80.09±0.75	79.71±1.93	80.81±2.01	94.78±0.72	97.67±0.58	94.07±1.36	98.28±0.64	96.65±0.89	96.99±0.62
Cotton firewood	67.72±2.20	74.93±1.28	73.07±3.35	73.18±2.15	95.88±1.22	95.51±2.30	88.81±1.73	98.36±0.76	97.10±1.11	98.01±0.71
Rape	74.58±2.22	79.67±0.62	81.33±2.02	82.27±0.91	86.52±1.23	91.39±1.05	93.66±0.76	92.09±1.41	93.97±0.90	96.65±1.12
Chinese cabbage	46.77±3.18	59.10±1.93	59.65±2.84	59.49±2.68	64.39±1.83	69.54±2.40	70.72±3.42	75.37±3.16	85.24±3.60	91.81±2.12
Pakchoi	26.81±2.41	41.34±1.60	45.10±3.44	46.04±1.39	69.12±2.75	89.87±3.03	75.22±5.45	91.27±2.98	95.01±2.69	98.47±1.09
Cabbage	87.38±1.46	91.09±0.55	92.83±0.87	92.51±0.72	93.33±2.40	94.43±1.27	94.49±1.02	96.45±1.01	98.39±0.84	98.71±0.69
Tuber mustard	34.91±2.63	54.76±3.09	57.96±1.56	55.21±0.89	77.65±4.11	88.95±2.17	84.61±7.37	90.12±2.74	91.89±2.36	93.63±1.61
Brassica parachinensis	31.32±3.81	42.03±1.51	47.27±2.88	46.12±1.97	62.96±3.20	79.87±1.96	77.53±3.48	79.03±4.61	89.68±2.87	93.95±1.63
Brassica chinensis	58.61±2.49	55.18±3.61	57.53±2.25	57.42±1.94	80.86±2.56	84.92±3.22	78.59±2.58	85.98±2.84	79.48±7.18	93.37±2.29
Small brassica chinensis	54.49±1.76	50.73±1.58	52.93±3.01	49.47±1.75	70.02±1.93	78.54±2.07	74.91±3.35	81.85±2.25	83.23±1.80	86.95±1.15
Lactuca sativa	55.01±2.14	69.02±1.08	71.25±2.33	69.04±2.78	81.13±2.18	92.90±1.94	90.44±1.70	94.30±1.27	96.14±1.53	97.80±0.65
Celtuce	74.07±1.74	86.71±0.98	87.90±1.82	87.05±1.75	92.53±2.10	98.57±0.63	96.71±1.08	98.17±0.90	99.40±0.72	99.78±0.47
Film covered lettuce	77.10±2.32	78.12±1.18	81.31±2.80	78.72±2.27	95.04±1.26	93.30±2.19	96.04±0.75	93.93±1.39	98.49±0.59	98.78±0.49
Romaine lettuce	65.87±2.60	76.81±0.78	77.92±2.45	77.08±2.33	96.23±1.34	97.00±1.79	94.44±3.51	97.21±1.91	98.97±0.72	99.70±0.25
Carrot	70.06±2.61	74.37±1.30	72.99±2.11	73.26±1.77	89.44±2.64	97.03±1.02	93.70±1.81	96.39±1.32	99.02±0.43	99.40±0.61
White radish	73.59±1.90	77.09±1.68	78.42±2.02	77.61±1.92	84.77±3.07	86.32±2.16	88.15±2.04	91.94±0.84	95.94±1.12	96.60±0.63
Garlic sprout	76.08±1.66	83.47±1.33	81.12±1.28	80.14±1.66	94.45±1.32	95.88±1.96	94.71±0.35	96.05±0.87	99.07±0.53	99.29±0.46
Broad bean	74.58±2.63	68.84±2.01	66.64±4.98	63.59±2.96	97.50±1.11	99.15±0.43	97.01±1.78	98.16±1.30	100.00±0.00	100.00±0.00
Tree	74.15±2.80	73.75±1.93	73.43±3.28	72.43±3.26	90.59±1.63	98.82±0.68	94.97±2.56	98.18±0.49	99.11±1.08	99.34±0.67
OA(%)	67.01±0.88	73.80±0.41	74.40±0.79	74.62±0.99	87.69±0.60	92.01±0.38	89.55±0.59	93.21±0.44	93.96±0.58	95.83±0.37
AA(%)	64.79±0.37	70.89±0.18	71.89±0.43	71.35±0.36	86.25±0.89	91.53±0.41	88.86±0.37	92.61±0.47	94.70±0.41	96.77±0.24
$\kappa \times 100$	60.70±0.92	68.28±0.44	69.04±0.83	69.20±1.07	84.62±0.74	89.95±0.47	86.95±0.70	91.45±0.54	92.40±0.71	94.75±0.46

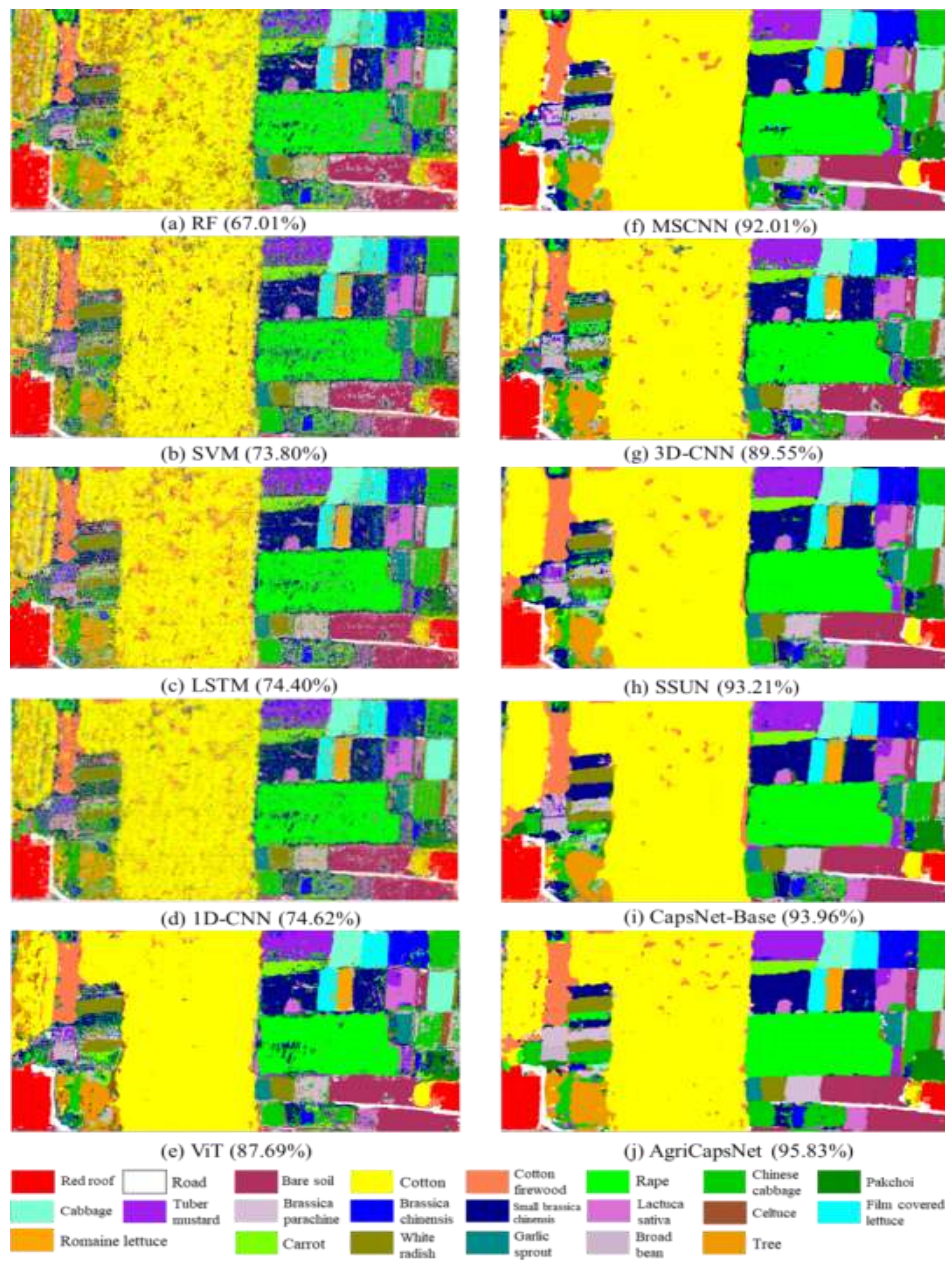


Figure 3. Classification maps of the WHH dataset obtained by different approaches

Conclusion

To enable fine-scale, intelligent classification of complex agricultural landscapes using hyperspectral imagery, this work introduces AgriCapsNet, a network specifically designed to amplify spatial-spectral feature representation. The model captures both spatial and spectral characteristics of diverse land cover categories through capsule-based vector neurons, which comprehensively encode instantiation parameters of object features, thereby enhancing feature discrimination and representation power. AgriCapsNet was evaluated on the crop-dense WHH dataset and compared against nine benchmark methods. The experimental results demonstrate that AgriCapsNet consistently outperforms its competitors in both qualitative and quantitative terms, yielding an OA improvement ranging from 1.87% to 28.82% on this dataset.

References

- Rasmussen, L.V. et al. 2024. Joint environmental and social benefits from diversified agriculture. *Science*. 384(6691): 87-93.
- Tong, Q., Xue, Y., and Zhang, L. 2014. Progress in hyperspectral remote sensing science and technology in China over the past three decades. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 7(1): 70-91.
- Ghamisi, P. et al. 2018. New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning. *IEEE Geoscience and Remote Sensing Magazine*. 6(3): 10-43.
- Du, P. et al. 2020. Advances of four machine learning methods for spatial data handling: A review. *Journal of Geovisualization and Spatial Analysis*. 4(1): 1-25.
- Ghamisi, P., Plaza, J., Chen, Y., Li, J., and Plaza, A.J. 2017. Advanced spectral classifiers for hyperspectral images: A review. *IEEE Geoscience and Remote Sensing Magazine*. 5(1): 8-32.
- Audebert, N., Le Saux, B., and Lefevre, S. 2019. Deep learning for classification of hyperspectral data: A comparative review. *IEEE Geoscience and Remote Sensing Magazine*. 7(2): 159-173.
- Hinton, G.E., Krizhevsky, A., and Wang, S.D. 2011. Transforming auto-encoders. In *Proceedings of the International Conference on Artificial Neural Networks*. 44-51.
- Sabour, S., Frosst, N., and Hinton, G.E. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*. Vol. 30.
- Li, H.C., Wang, W.Y., Pan, L., Li, W., Du, Q., and Tao, R. 2020. Robust capsule network based on maximum correntropy criterion for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 13: 738-751.
- Lecun, Y., Bengio, Y., and Hinton, G. 2015. Deep learning. *Nature*. 521(7553): 436-444.
- Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., and Xu, C. 2020. GhostNet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1580-1589.
- Zhong, Y., Hu, X., Luo, C., Wang, X., Zhao, J., and Zhang, L. 2020. WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF. *Remote Sensing of Environment*. 250: 112012.
- Mountrakis, G., Im, J., and Ogole, C. 2011. Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*. 66(3): 247-259.
- Belgiu, M. and Drăguț, L. 2016. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*. 114: 24-31.
- Xu, Y., Zhang, L., Du, B., and Zhang, F. 2018. Spectral-spatial unified networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*. 56(10): 5893-5909.
- Hu, W., Huang, Y., Wei, L., Zhang, F., and Li, H. 2015. Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*. 2015: 1-12.
- Dosovitskiy, A. et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv:2010.11929*.
- Chen, Y., Jiang, H., Li, C., Jia, X., and Ghamisi, P. 2016. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*. 54(10): 6232-6251.