sustainability

MDPI

Article

Hyperspectral Classification of Grasslands for Sustainable Management Using Feature Fusion GRCNet

Yuke Liu, Yilei Liu and Xin Pan *

College of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot 010010, China; 2023202100006@emails.imau.edu.cn (Y.L.); liuyilei@emails.imau.edu.cn (Y.L.)
* Correspondence: pxffyfx@126.com

Abstract: Grasslands play a crucial role in ecosystems, influencing key ecological functions such as biodiversity, climate regulation, and soil and water conservation. With the impacts of environmental changes and human activities, the functional status and health of grasslands are facing challenges. Therefore, efficient identification of grassland status is of great significance for the sustainable management, ecological protection, and restoration of grassland resources. To support the sustainable development of grasslands, the GRCNet network model is proposed. Grassland sweep photography was performed via a UAV-mounted hyperspectral imager to establish a 13-category grassland hyperspectral dataset. Then, Gaussian filter and principal component analysis (PCA) were used for noise reduction and dimensionality reduction in the hyperspectral images, and the GRCNet network model, which mainly consists of the GCA module, the RDC module, and the VIT-Base module, was established for the classification task. The experiments use the average accuracy, overall accuracy, F1 score, Kappa coefficient, and running time as the performance indexes, and the eight methods are compared with GRCNet. The results showed that the GRCNet network performed the best with AA reaching 92.84%, OA reaching 94.59%, F1 score reaching 97.22, and Kappa coefficient reaching 0.93. The GRCNet method is 10–20% more accurate than other methods. Comparative tests were also conducted using two public datasets, and GRCNet performed the best with a 2–20% improvement in accuracy. The results demonstrate the effectiveness of the GRCNet network model in hyperspectral grass classification tasks, which can be used as an efficient solution to the current problem of low hyperspectral accuracy and poor stability.

Keywords: sustainable development; hyperspectral image classification; grassland recognition; feature fusion; re-parameterized refocused convolution; attention mechanism; ViT

Academic Editor: Antonio Miguel Martínez-Graña

Received: 28 December 2024
Revised: 8 February 2025
Accepted: 19 February 2025
Published: 20 February 2025

Citation: Liu, Y.; Liu, Y.; Pan, X. Hyperspectral Classification of Grasslands for Sustainable Management Using Feature Fusion GRCNet. *Sustainability* **2025**, *17*, 1804. https://doi.org/10.3390/su17051804

1. Introduction

Grasslands are composite plant communities, accounting for 1/2 of the total land area in the world, are an essential part of ecology, and play an important role in maintaining soil and water, preventing wind, fixing sand, and regulating the climate [1–3] The health status and type identification of the grassland is of great significance to ecological environment monitoring, land use planning, and agricultural production. Grassland testing plays an important role in ecological sustainability. Hyperspectral imaging technology is based on acquiring spectral information of objects in narrow bands, providing rich spectral and spatial information, and has been widely used in feature identification, food safety, and medical diagnosis [4–6]. Hyperspectral images possessing rich spectral information of features can be used as a method to recognize grass species, and how to efficiently

recognize grass species categories becomes a key task. However, traditional deep learning feature classification methods cannot efficiently solve the problem of large data volume with many bands of hyperspectral data [7–10]. Wei-Qiang Pi [11] created a lightweight DGC-Fast-3DCNN and used the spatial information of UAV hyperspectral data to solve the problem of grassland degradation, and the overall accuracy reached 98.11%. Unile et al. [12] obtained three wavebands through image preprocessing and used Support Vector Machines (SVMs) and the Random Forest method for species identification, and the accuracies reached 96.92%. Jie Zhang et al. [13] proposed a dense residual convolutional network for classification research based on a residual network structure with accuracy, and Hao Li et al. [14] utilized a convolutional neural network (CNN), which combines CNN and LSTM, to effectively classify large-scale spectral data with an accuracy of 97.35% and excellent performance in stellar spectral data classification. Zhao Su [15] proposed an end-to-end, pixel-to-pixel semantic segmentation method for recognizing paddy fall on the basis of U-Net network by combining the advantages of dense blocks, DenseNet, attention mechanism, and jump connection for UAV (Unmanned Aerial Vehicle) remote sensing images. And the method [15] was able to process the input multi-band image. The accuracy of the model was 97.30% on fallen-rice images. The convolutional neural network, as a deep learning method, has been widely used in various image classification tasks.

However, the traditional convolutional neural network fails to fully utilize the spectral and spatial features of hyperspectral data when processing these data and cannot efficiently solve the problem of the semantic segmentation accuracy of hyperspectral images. To address the problem that hyperspectral data are too informative and complex in terms of bands to be recognized efficiently and accurately, this study innovatively proposes the GRCNet (Group Ref Conv Net) network model, which can efficiently process complex hyperspectral images in GRCNet, which is a new network architecture designed for accurate feature recognition and classification and is highly accurate and stable. The backbone of the network uses a linear structure and uses residual connections to enhance parameter transfer. The core of the network consists of the GCA (Group Conv A) module and the RDC (Ref Deformable Conv) module. The GCA module introduces grouped convolution and feature fusion to replace ordinary convolutional blocks with grouped convolution and performs feature fusion between the blocks, which improves the feature extraction and computation capability. The RDC module introduces the RefConv (Re-parameterized Refocusing Convolution) module and uses feature fusion to fuse two different types of attentional mechanisms, which enhances the local feature extraction capability in the convolution operation and increases the accuracy of the model. To further validate the advantages of the method, comparison experiments are conducted using two publicly available hyperspectral datasets (Indian Pines, Salinas). Finally, ablation experiments were conducted to demonstrate that the modified module improves accuracy compared to the original version. GRCNet provides an effective new method for the analysis and application of hyperspectral data, which can provide useful reference for future hyperspectral data processing and ecological monitoring research.

The main research directions of this study include the following:

(1) The problem of grassland species identification can be effectively solved using hyperspectral imagery and deep learning methods, providing new and effective solutions for the sustainable development of ecological environment.

(2) Currently available near-ground UAV hyperspectral datasets have low resolution, few species, and little hyperspectral differentiation between crops. To address these issues, a new hyperspectral dataset was created by using a UAV-mounted hyperspectral imager to take sweeping shots of the grass. The sample points of each species are rich,

and the image resolution is high. It can be used as an excellent hyperspectral image classification dataset.

(3) The traditional network model with low accuracy and poor stability is not applicable to the near-ground UAV hyperspectral classification task; in order to solve such problems, the GRCNet network is proposed. And the accuracy and stability of GRCNet is verified on a self-built hyperspectral dataset and three public hyperspectral datasets.

(4) Existing methods rarely fuse CNNs with ViT (Vision Transformer) models; in order to provide new deep learning ideas, a new method is used to fuse CNNs with ViT models and to fuse local features with global features.

2. Materials and Methods

2.1. Study Area

The data used in this experiment were taken from the Experimental Base of Agricultural and Animal Husbandry Crossing Area of Grassland Research Institute of Agricultural Sciences of China (111°47′ E, 40°35′ N) in Hohhot, Inner Mongolia, China, with an elevation of 1072 m, an average annual temperature of 6° C, and a frost-free period of 133 days, which has the typical continental climatic features of agricultural and animal husbandry crossings and is dominated by sandy typical grassland types, with sandy loamy soils. As shown in Figure 1.

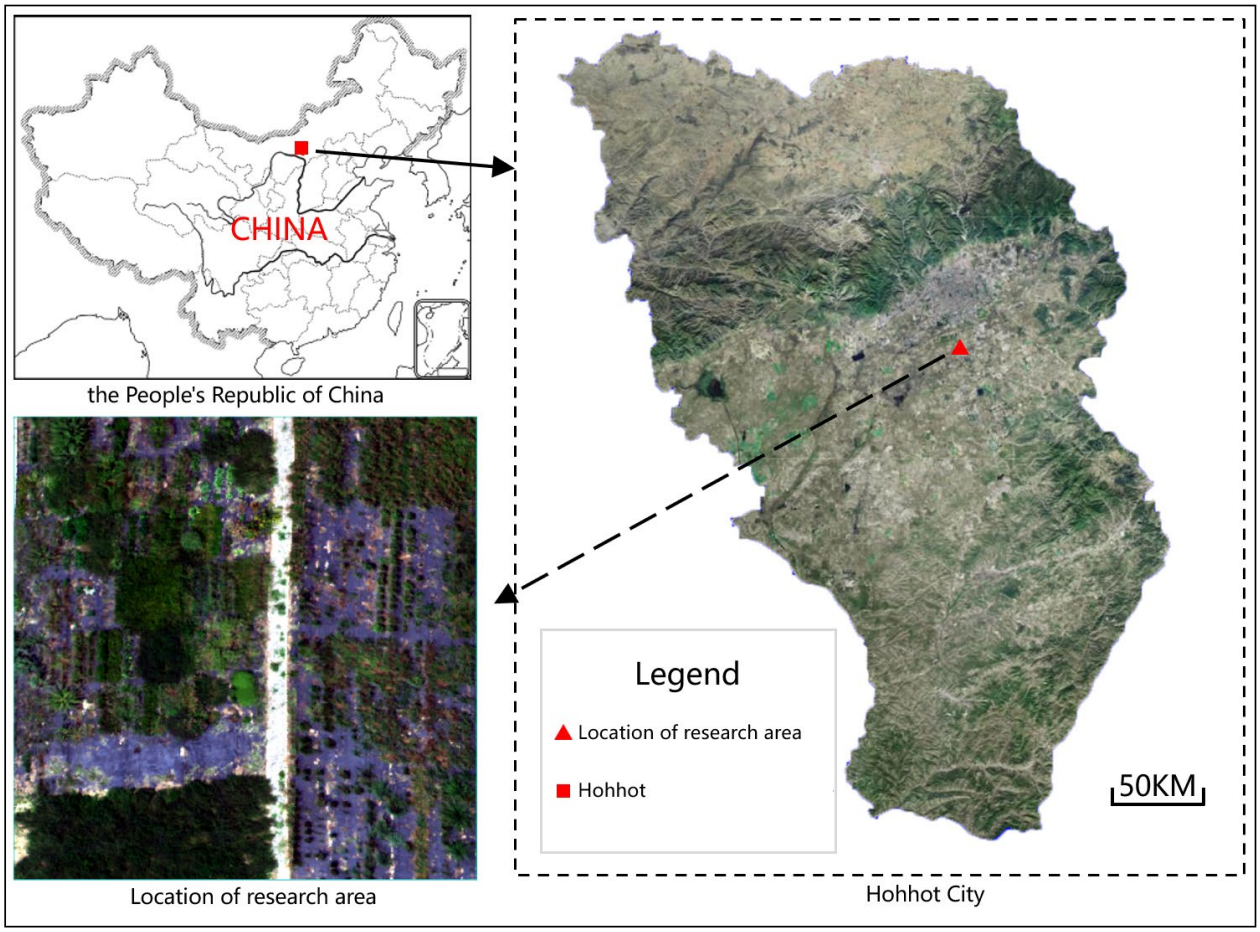

Figure 1. Research area.

2.2. Image Acquisition

The image acquisition equipment used this time was a DJI M300RTK rotor UAV equipped with a Nano-Hyperspec new miniature airborne hyperspectral imager, as shown in Figure 2. Grass-level sweeping was used to capture the images at the Experimental Base of the Agricultural and Animal Husbandry Intertwined Area of China Grassland Research Institute of Agricultural Sciences, Hohhot, Inner Mongolia (111°47′ E, 40°35′ N), with a spectral range of 400–1000 nm, 270 spectral channels, a spectral sampling rate of about 2.2 nm/pixel, 640 spatial channels, and a UAV flight altitude of 65 m. The hyperspectral camera was set to shoot at a focal length of 17 nm, an imaging resolution of 7.4 um, and an exposure time of 10 ms. The date of the shoot was May 2024, and the time of the shoot was chosen to be between 12:00 and 14:00 noon. The hyperspectral dataset Hohygrass-13 was created with thirteen categories (Ground, Virginal, Brome grass, Chinese wildrye, Lamewort, Sanguisorba, Alfalfa, Barley awn, short-awned barley grass, Pennisetum, Lyme grass, Sophora alopecuroides, Radix centaurae) with a dataset size of 700 × 700 pixels. The detailed categories of grass species are shown in Table 1, and the grass spectral information is shown in Figure 3.

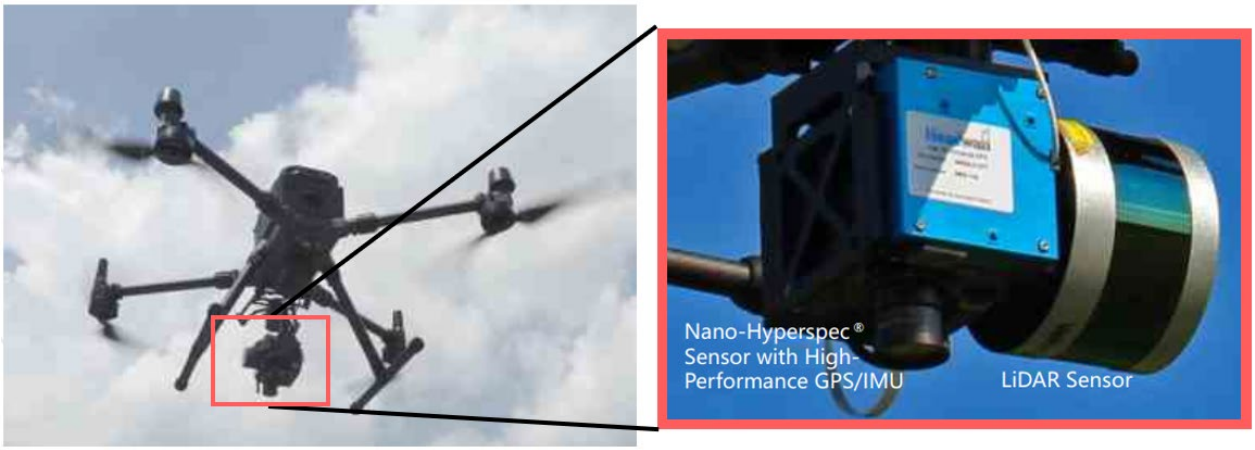

Figure 2. Research equipment.

Table 1. Grass species category.

ID	Category	Sample Count
C1	Ground	163,654
C2	Virginal	3502
C3	Brome grass	55,483
C4	Chinese wildrye	13,466
C5	Lamewort	17,634
C6	Sanguisorba	5742
C7	Alfalfa	49,630
C8	Barley awn	69,558
C9	Short-awned barley	33,620
C10	Pennisetum	9146
C11	Lyme grass	42,521
C12	Sophora alopecuroides	14,388
C13	Radix centaurae	11,656

Figure 3. Hohygrass-13 truth graph.

2.3. Data Pre-Processing

The hyperspectral images are characterized by dense bands, large amounts of bands, and a rich band information, etc. An excessive number of channels will affect the accuracy of hyperspectral classification and recognition, and excessive data will significantly increase the training time of the algorithm, so the preprocessing step is crucial in hyperspectral image processing. In this experiment, we mainly use the Gaussian filter and principal component analysis (PCA) [16] to preprocess hyperspectral images.

First of all, the Gaussian filter is used, which mainly uses a Gaussian function for smoothing the image; it achieves smoothing by matching the weighted average of each pixel value in the image with the pixel values in its neighborhood in order to reduce the noise and details. The use of Gaussian filter in this study reduces the error information that is subject to noise. Since hyperspectral images are characterized by a large number of channels and high density, which can significantly increase the computation time, filtering is required. Therefore, the principal component analysis (PCA) is used for dimensionality reduction. The PCA improves stability by linearly transforming and projecting high-dimensional data into a low-dimensional space while retaining the main information of the original data as much as possible. The core formula is as follows:

$$\sum Y = \frac{1}{2}\frac{1}{n-1}X^T X \tag{1}$$

$$Z = XW \tag{2}$$

where Y is the covariance matrix, X is the data matrix after centering, W is the eigenvector matrix of the covariance matrix, n is the number of selected eigenvectors, and Z is the data matrix after projection to the principal component space. The number of original data channels before use is 270, and the number of channels after dimensionality reduction is finally obtained to become 47 by PCA cumulative variance contribution selection method. These principal components preserve most of the information, with a cumulative contribution rate exceeding 95%. The first five principal components account for approximately 30%, 25%, 15%, 10%, and 8% of the variance, respectively. The cumulative variance contribution rate of the first ten principal components exceeds 85%.

2.4. Network Modeling

In this study, the GRCNet neural network is proposed to address the problems of a large amount of information and complex bands in hyperspectral data, which cannot be recognized efficiently and accurately. GRCNet is a new network architecture designed for accurate feature recognition and classification with high accuracy and stable results. The backbone of the network uses a linear structure, and the residual connections are used to enhance the parameter transfer and reduce the gradient loss. The core of the network consists of the GCA module and RDC module, and the ViT-Base module [17] is added at the end for global feature extraction. The GCA module introduces the grouped convolution and feature fusion method to replace the ordinary convolution blocks with grouped convolution, and the feature fusion operation is carried out between the blocks, which can improve the feature extraction and computation ability. The RDC module introduces the RefConv (Re-parameterized Refocusing Convolution) module and uses feature fusion to fuse two different types of attentional mechanisms, which can enhance the local feature extraction capability in convolutional operations to increase model accuracy.

The GRCNet network model first passes the preprocessed spectral image into the initial convolution layer (Initial Conv), and the initialization module uses the CBR module (Convolution–BatchNorm–ReLU) as the first computational processing of the image. The initialized parameters are passed into the GCA module, which is mainly used for feature extraction and hyperspectral image training, using the grouped convolution module (Group = 32) instead of the ordinary convolution block, and the parameters after the first grouping calculation are fused with the parameters after the second grouping calculation for the feature fusion, and finally, a 1×1 convolution block is used to implement the channel-mapping operation for the subsequent parameter transfer. After that, it passes into the GCB module, which also uses the grouped convolutional module and uses the inverse bottleneck structure to gradually increase the channel first and then reduce it using the mapping module to achieve high computational efficiency. After that, it passes into the RDC module, which passes through two RefConv modules, and then inputs into two attention mechanisms (Deformable and Coordinate) to combine the two results through feature fusion, so that the model output results are stable and the recognition accuracy rises, and then finally passes through the ViT-Base module, extracts the global features, and outputs the results. The structure of the GRCNet network is shown in Figure 4, and the specific parameters of each module are shown in Table 2.

Table 2. Network parameters.

Module Name	Type	Input Data Size	Output Data Size	Parameter Quantity	Kernel_Size
Initial Conv	StandardConv	(num_of_bands, H, W)	(64, H-2, W-2)	(9 × num_of_bands + 1) × 64	3 × 3
GCA block	GroupConv-A	(64, H-2, W-2)	(128, H-2, W-2)	(64 × 128/32) × 3 × 3 + 128	3 × 3
	GroupConv-B	(128, H-2, W-2)	(128, H-2, W-2)	(128 × 128/32) × 3 × 3 + 128	3 × 3
	FusionConv	(256, H-2, W-2)	(64, H-2, W-2)	256 × 64×1×1 + 64	1 × 1
GCB block	GroupConv-A	(64, H-2, W-2)	(128, H-2, W-2)	(64 × 128/32) × 3 × 3 + 128	3 × 3
	GroupConv-B	(128, H-2, W-2)	(128, H-2, W-2)	(128 × 128/32) × 3 × 3 + 128	3 × 3
	FusionConv	(128, H-2, W-2)	(64, H-2, W-2)	128 × 64 × 1 × 1 + 64	1 × 1
RDC block	RefConv	(64, H-2, W-2)	(64, H-2, W-2)	(3 × 3 × 64 + 1) × 64	3 × 3
	Deformable	(64, H-2, W-2)	(64, H-2, W-2)	10,953	
	Coordinate	(64, H-2, W-2)	(64, H-2, W-2)	2050	
	Fusion Conv	(128, H-2, W-2)	(64, H-2, W-2)	(1 × 1 × 128 + 1) × 64	1 × 1
ViT-Base		(64, H-2, W-2)	(batch_size, 768)	86,730,538	3 × 3 (patch)

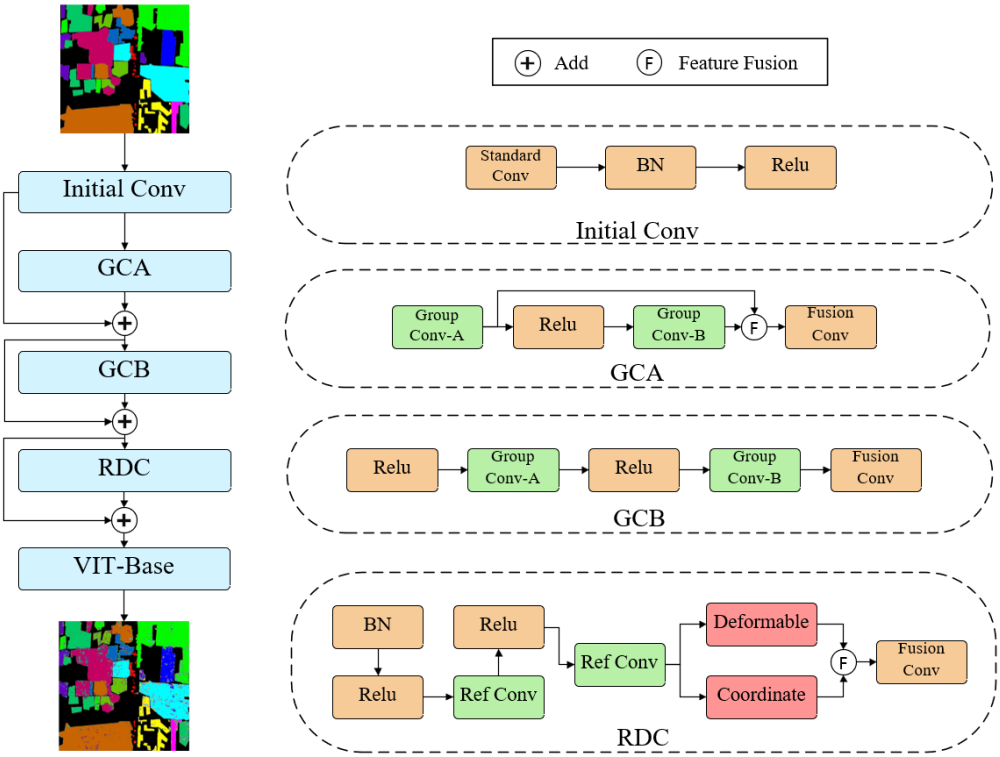

Figure 4. Network structure.

2.4.1. RefConv Module

The RDC module is designed to solve the problem that the traditional convolutional module has a large computational capacity, poor feature performance, and is more prone to overfitting. Instead of the ordinary convolutional block, Re-parameterized Refocusing Convolution is added into the network to effectively improve the data computation accuracy. It is a new convolutional layer design, which is based on the idea of structural reparameterization and aims to improve the performance of convolutional neural networks by refocusing transformations of the convolutional kernel without increasing the data computation time. The RefConv technique equips each convolution with an additional refocusing transformation operation (new learnable parameters), and the weights of these parameters are trained during training and mixed with the original weights during computation without adding additional weights. In the paper presented by Z Cai [18] and others, experiments were conducted and verified that RefConv can significantly improve the performance of various backbone models without additional computation time. In

addition, the paper shows that RefConv can improve the performance of convolutional neural networks for target detection and semantic segmentation.

RefConv sets the convolutional kernel size to K × K by replacing the regular convolutional layers and has the same number of input and output channels. After passing the parameters into the network, the weights W_r are initialized, and these weights are combined with the original weights W_b to generate new weights W_t. This weight is then passed to the group convolution for computation, and finally, the result is regenerated to exactly the same structure as the original convolution structure, keeping only the newly generated W_t, which is the output parameter. This process computes the output model exactly the same as the original model, and does not increase the computation time outside the module. The specific form of the RefConv network is shown in Figure 5.

$$w_b, w_t \in \mathbb{R}^{(C_{out} \frac{C_{in}}{g} \times K \times K)} \tag{3}$$

where C_{in} is the number of input channels, C_{out} is the number of output channels, the group number g is initially set to $C = C_{in} = C_{out} = g$, and the convolution kernel size is K. Typically, the initial base weights and transform weights of the model are shaped in such a way that $K = 3$ by default, and padding = 1 to ensure that w_b, w_t have the same shape. In order to reduce the number of parameters, this paper proposes a formula to determine the array G of hyperparameters:

$$G = \frac{C_{out} \times C_{in}}{g^2} \tag{4}$$

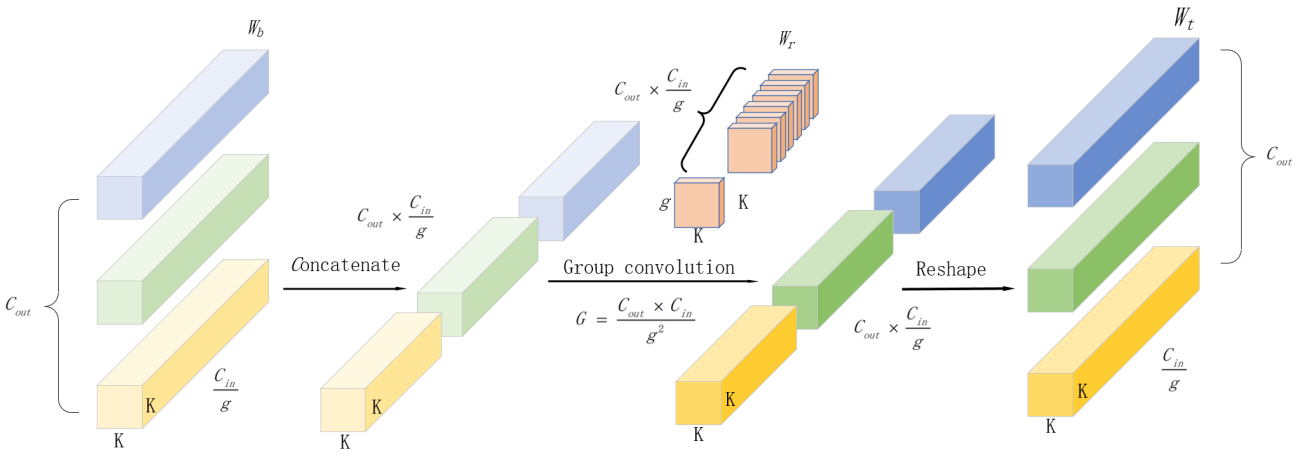

Figure 5. Re-parameterized Refocusing Convolution.

This design would make RefConv complementary to the original convolution; a larger g implies a sparser original convolution that requires more cross-channel connections, while a larger g results in a smaller G, which fulfills this requirement. Suppose the size of the feature map is $B \times C_{in} \times H \times W$, B is the batch, C_{in} is the number of input channels, H is the height of the image, W is the width of the image, K is the size of the convolution kernel, and small k is the scaling factor. Then, the FLOPs of the original convolution are as follows:

$$BHW \cdot \frac{C_{out} \times C_{in} \times K^2}{g} \tag{5}$$

And refocusing transformation's FLOPs are only the following:

$$\frac{C_{out} \times C_{in} \times K^2 \times k^2}{G} \tag{6}$$

This formula does not depend on the batch size B. For example, assuming B = 256, $H = W$ = 28, $C_{in} = C_{out} = g$ = 512, and K = 3, the FLOPs for the original convolution are 925 M, while the FLOPs for the refocusing transformation is only 21 M.

2.4.2. Deformable Attention and Coordinate Attention

Usually, attention mechanisms significantly increase the computational complexity of a model and introduce a large number of parameters that pose a challenge to the model's memory usage, and different attention mechanisms capture different information content that cannot be used directly. In this study, we experimentally fused two different types of attention mechanisms and effectively combined their properties to solve the integration problem. Deformable Attention [19–22] is an enhanced attention mechanism that aims to address the limitations of traditional attention mechanisms when dealing with complex visual tasks. It improves the performance of the model by introducing deformable sampling points that enable the attention mechanism to flexibly focus on different regions of the input feature map. The structure of Deformable Attention is shown in Figure 6.

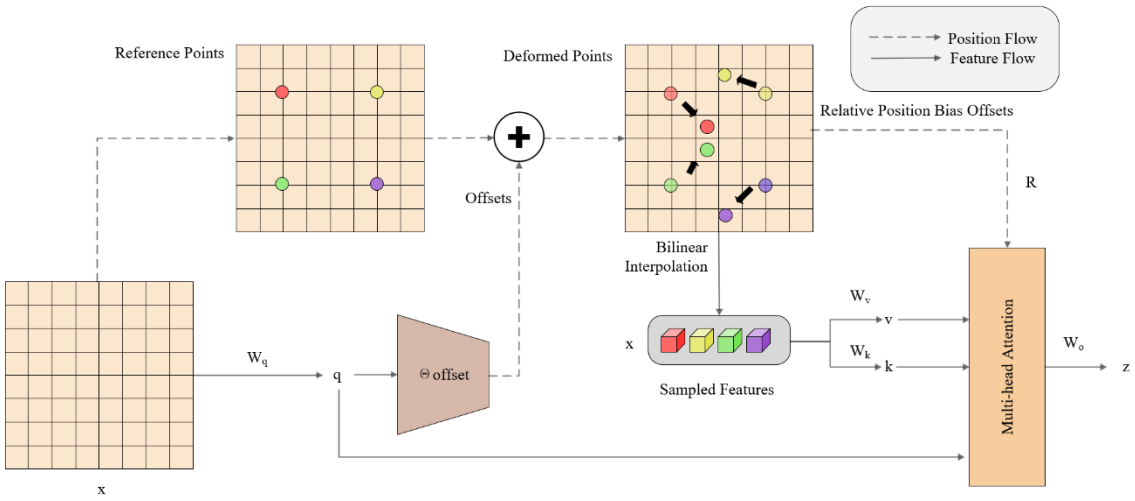

Figure 6. Deformable Attention structure.

The Deformable Attention module first queries each location by offset computation and generates a set of offsets Δp_i to adjust the position of the sampling point through an offset network. Next, specific sample point locations are determined by summing the query points q and the offsets $pi = q + \Delta p_i$. Then, the feature map is sampled at these sample point locations to obtain the corresponding feature values X (pi) = X ($q + \Delta p_i$). After obtaining these base eigenvalues, the module performs a weighted summation of the eigenvalues of each sampling point with its attention weight to obtain the final output as in Equation (7):

$$Y(q) = \sum_{i=1}^{n} A(q, p_i) \cdot X(p_i) \tag{7}$$

where Y (q) is the output, A (q, pi) is the computed attentional weight of the sampled points, and X (p_i) is the feature value. After generating the reference point and completing the offset calculation, after feature sampling and projection at the computed offset point location, the calculation is performed using the multi-head attention approach, where the attention scores are computed and weighted and summed, and the results are output.

$$z^{(m)} = \sigma\left[\frac{q^{(m)}\widetilde{k}^{(m)}}{\sqrt{d}} + \phi(\hat{B}; R)\right]\widetilde{v}^{(m)} \tag{8}$$

where q is the query, k is the key, v is the value vector, d is the dimension of each header, and ϕ $(B; R)$ is the bias value extracted from the relative position bias table B using the relative position information R. Finally, the output of each header is concatenated and linearly projected to obtain the final output.

$$z = Concat(z^{(1)}, \ldots, z^{(m)})W_o \tag{9}$$

where *Concat* is the output of all the heads spliced in the last dimension to form a new representation. W_o is the output projection matrix.

Coordinate attention [23,24] captures long-range dependencies in one spatial direction while retaining precise positional information in the other by decomposing channel attention into a one-dimensional feature encoding process that aggregates features along two spatial directions. This design retains the advantages of channel attention in capturing global information while introducing location information, thus improving the ability to generate spatially selective attention maps, and the coordinate attention structure is shown in Figure 7.

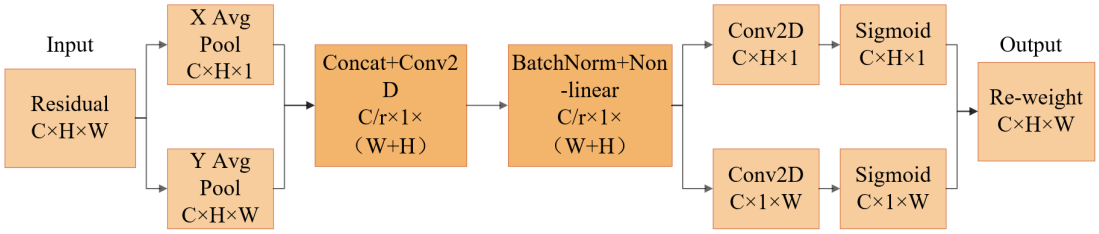

Figure 7. Coordinate attention structure.

The coordinate attention mechanism first aggregates the features of each channel through global average pooling, captures inter-channel dependencies, and generates weights through two fully connected layers to be applied to the input feature map. In order to better capture the precise location information, horizontal coordinate encoding and vertical coordinate encoding are set up, respectively:

$$z_h^c(h) = \frac{1}{W}\sum_{i=1}^{W-1}\binom{n}{k}x_c(h,i) \tag{10}$$

$$z_h^c(w) = \frac{1}{H}\sum_{i=1}^{H-1}\binom{n}{k}x_c(j,w) \tag{11}$$

where z is the output result, W is the width, H is the height, and x (h, i) is the encoded position of the feature map. The above encoded feature maps are concatenated and fed into a shared one-dimensional convolutional transformation function F1 to generate an intermediate feature map f = δ(F1 [z_h,z_w]) that contains orientation information. They are then converted into 1D convolutional transformations separately to generate two attention maps g_h = σ(F_h(f_h)) with g_w = σ(F_w(f_w)), which are finally applied to the input feature maps by multiplication:

$$y_c(i,j) = x_c(i,j) \times g_h^c(i) \times z_w^c(j) \tag{12}$$

2.4.3. ViT-Base Module

To better capture global features, the model uses the ViT-Base module. ViT-Base (Vision Transformer Base) is a vision model based on the Transformer architecture, specialized for computer vision tasks such as image classification, target detection, etc. The Base model

is a common configuration of ViT and is a standard ViT model that contains 12 layers of Transformer blocks, a 768-dimensional hidden layer, and 12 self-attentive heads.

As shown in Figure 8, first, the incoming image is sliced into fixed-size image blocks (Patches); then, these patches are converted into one-dimensional vectors, each patch is treated as a Token, all the Token are used as inputs to the model, and Positional Encoding is added to each Token. Pass the location of the original image for each patch. After that, add the CLS Token, which is a token that can represent the global features of the entire image that will eventually be used for classification. Input these into the Transformer Encoder; this part mainly consists of multiple Encoder layers, each containing Self-Attention and Feed-Forward Neural Networks. Eventually, extracting the image features carried by the CLS Token using MLP Head. Output the classification result.

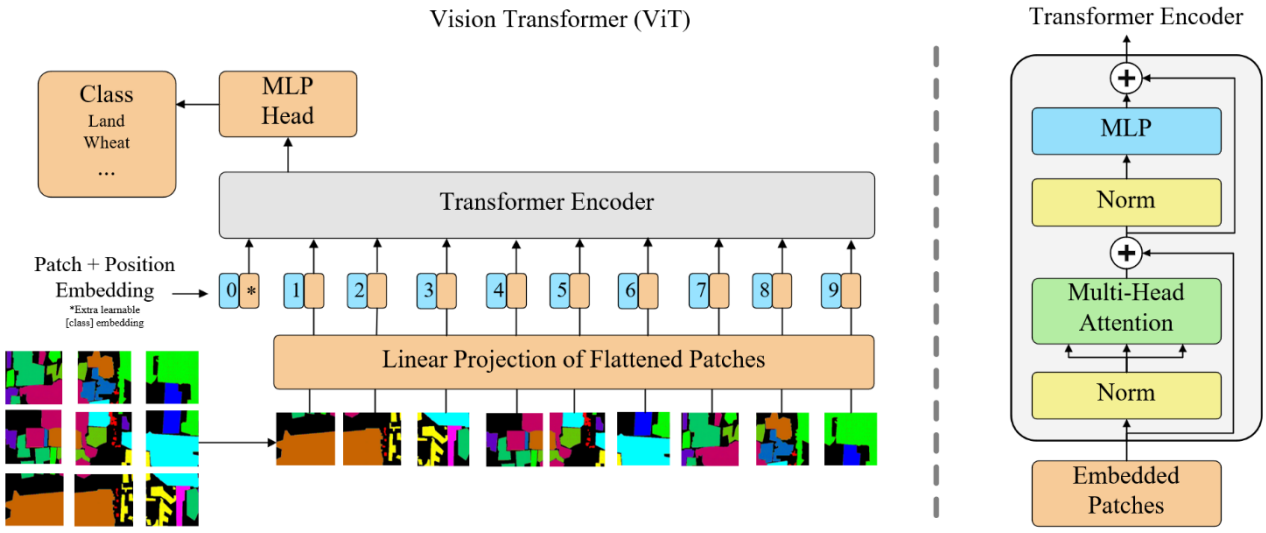

Figure 8. Vision transformer structure.

2.5. Experimental Environment and Configuration

In the experimental environment, the operating system is Window11, GPU is NVIDIA GeForce RTX4070 with 12 G of video memory, 32 G of RAM, CUDA version 12.1, PyTorch version 2.2.1, python version 3.11.8.

The specific values of the network training parameters are shown in Table 3. Selecting 20% of the sample points of each category as training can firstly significantly improve the network training efficiency and ensure that the network will not receive too much information to appear overfitting state. Patch width is set to 9 × 9 to ensure that it is consistent with the width of ViT. The initial learning rate is set to 0.1 to accelerate the training speed, and the final learning rate is set to 0.01 to ensure the training accuracy.

Table 3. Environment configuration.

Parameter	Value	Paraphrase
Samples per class	20%	Selection ratio of each type of sample
Patch	9	The side length of the patch
Lr	0.1	Initial learning rate
Lrf	0.01	Final learning rate
Learning epochs A	170	Initial learning epochs
Learning epochs B	30	Final learning epochs
Batch size	64	The number of data samples used in each training iteration

2.6. Evaluation Indicators and Experimental Setup

The evaluation metrics for the experimental results are average accuracy (AA), overall accuracy (OA), F1 Score, Kappa coefficient, and running time (Times):

$$AA = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

$$OA = \frac{1}{N} \sum_{i=1}^{n} Accuracy_i \tag{14}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{15}$$

$$Kappa = \frac{PA - PE}{1 - PE} \tag{16}$$

where TP (True Positive) is the number of samples correctly categorized as positive, TN (True Negative) is the number of samples correctly categorized as negative, FP (False Positive) is the number of samples incorrectly categorized as positive, FN (False Negative) is the number of samples incorrectly categorized as negative, and N is the total number of categories. Accuracy is the classification accuracy of the ith category. Kappa coefficient is a parameter used to measure the classification consistency by means of a confusion matrix, where PA is the ratio of the number of correctly categorized samples to the total number of samples; and PE is the expected probability of consistency.

This was followed by a comparison test using the self-constructed hyperspectral dataset, Hohygrass-13, and three publicly available hyperspectral datasets, Indian Pines, Pavia University, and Salinas. To ensure the breadth of the control classification methods, nine models were selected as controls, which cover a wide range of methods applicable to different data types and tasks, including U-Net, GCN [25], KNN [26], Fast-3DCNN, Fast-2DCNN, ResNet-18, ResNet-50 [27,28], DCGAN [29], and MRViT [30]. U-Net performs well in medical segmentation and is also suitable for remote sensing image classification; GCN performs feature extraction on data nodes and performs well for graph data classification; KNN is a nonparametric regression algorithm used to contrast deep learning against machine learning; Fast-3DCNN is a 3D convolutional network based on the improvement of C3D, and Fast- 2DCNN is an improved 2D convolutional network based on AlexNet; ResNet network is a residual network based on the residual structure, which is used to control the residual structure in this experiment; DCGAN network is an adversarial network, which can be used for image classification tasks by competing with the generator and the discriminator to improve the classification accuracy. MRViT is a novel network model specifically designed for hyperspectral images.

3. Results

3.1. Experimental Results

By using nine models on the self-constructed Hohygrass-13 hyperspectral dataset for comparative experiments, the models were used to classify and recognize each of the four feature classes in the figure. The results in Figure 9b show that GRCNet performs the best among all the models, with AA reaching 92.84%, OA reaching 94.59%, F1 score reaching 97.22, and Kappa coefficient reaching 0.93. Compared to the other eight algorithms, AA improved by 5%~70%, OA improved by 5%~60%, and F1 score improved by 3~50%. Figure 9a is taken from the well captured hyperspectral image truth map. ResNet-50 Figure 9d Compared to ResNet-18 (Figure 9c) has an excellent performance accuracy of about 89% in three performance metrics, which is the best performance, so the residual connection is chosen as the backbone network to transmit the data. The AA metrics of

Fast-3DCNN Figure 9f are significantly better than that of Fast-2DCNN (Figure 9e), but its computation ResNet network outperforms CNN, and the AA, OA, and F1 of ResNet network outperform the CNN parameters. The DCGAN network (Figure 9g) performs poorly in the comparison experiments, and the comparison with other datasets shows that DCGAN is not suitable for the complex datasets. U-Net (Figure 9h) and GCN (Figure 9i) performs well compared to the other. The model has a lead in some of the parameters, but it is not outstanding. The KNN algorithm (Figure 9j) is a machine learning method without parameter amount computation and is not applicable to hyperspectral semantic segmentation task with only 26.93% accuracy. According to the results of comparison experiments, GRCNet is significantly better than other classification models in grass spectral recognition and classification, with the highest accuracy and the best stability. The model recognition results are shown in Figure 9, and the confusion matrix of GRCNet is shown in Figure 10, and the specific data are shown in Table 4.

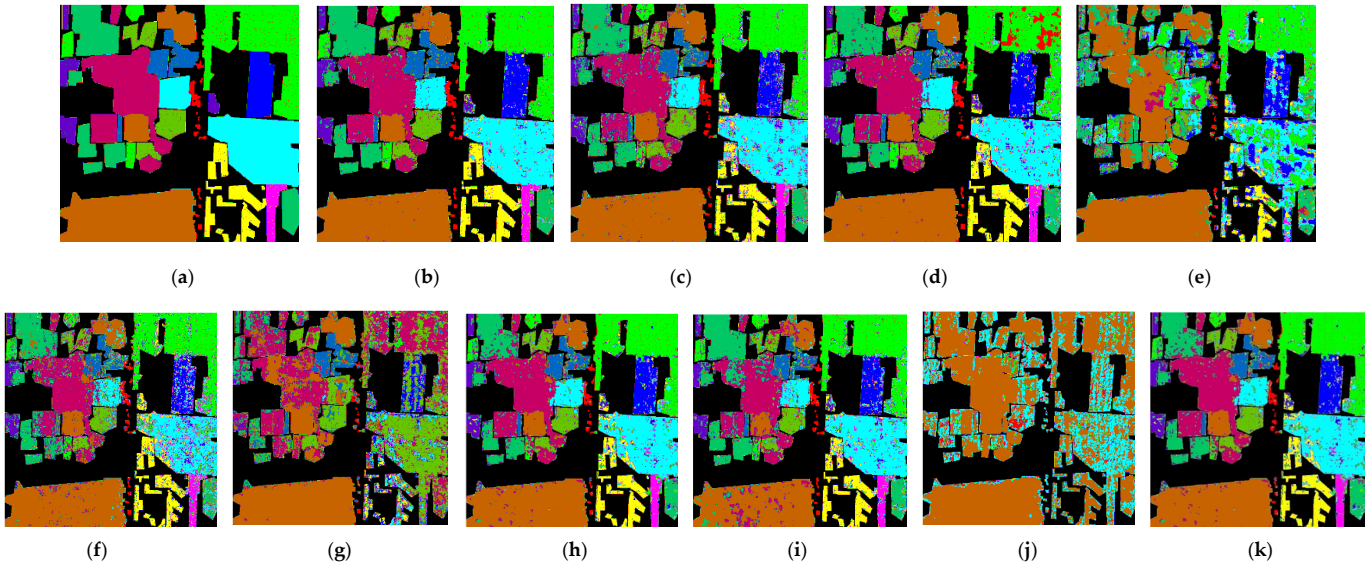

Figure 9. Experimental results of Hohygrass-13: (**a**) truth plot; (**b**) GRCNet; (**c**) ResNet-18; (**d**) ResNet-50; (**e**) Fast-2DCNN; (**f**) Fast-3DCNN; (**g**) DCGAN; (**h**) U-Net; (**i**) GCN; (**j**) KNN; (**k**) MRVIT.

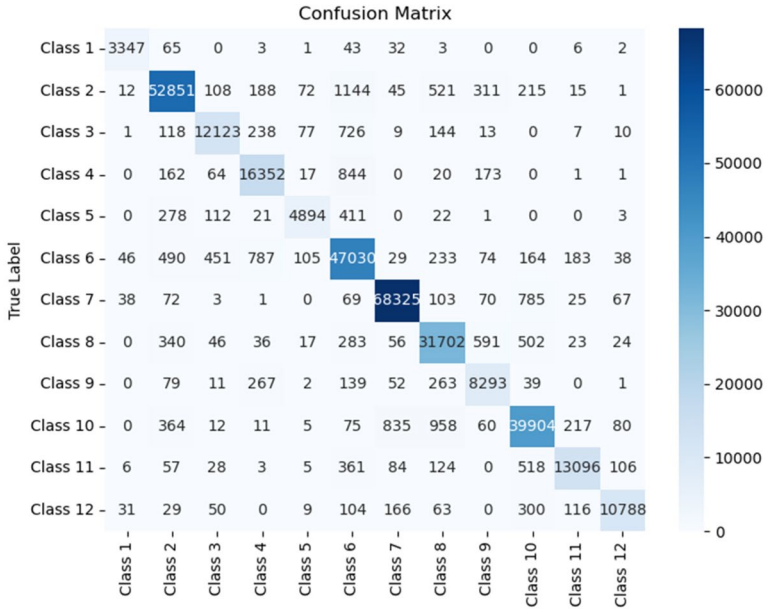

Figure 10. GRCNet confusion matrix.

Table 4. Experimental results of Hohygrass-13.

Mod	AA (%)	OA (%)	F1	Kappa	Times (Second)	Number of Parameters
GRCNet	92.84	94.59	97.22	0.93	3140	82.89 M
ResNet-18	74.38	83.84	91.21	0.81	936	11.73 M
ResNet-50	87.89	89.43	94.42	0.88	1335	25.67 M
Fast-2 DCNN	38.63	51.04	67.58	0.42	303	2.32 M
Fast-3 DCNN	75.90	79.31	88.46	0.77	525	7.52 M
DCGAN	41.81	43.28	60.41	0.32	413	2.66 M
U-Net	83.31	87.63	91.37	0.82	391	7.13 M
GCN	83.97	87.00	93.05	0.83	231	0.06 M
KNN	13.05	26.93	42.44	0.08	181	None
MRVIT	83.73	87.45	93.05	0.85	2441	57.22 M

3.2. Comparative Tests

To further demonstrate the effectiveness of the GRCNet method, three public dataset datasets, Indian Pines (Figure 11), Pavia University (Figure 12), and Salinas (Figure 13), were selected for comparative experiments, and eight models were selected as control experiments for GRCNet. The Indian Pines (Figure 11) public dataset contains 220 spectral bands covering the spectral range from 400 nm to 2500 nm. There are 16 feature classes, e.g., soybean, corn, grass, and woods. It is characterized by a large number of spectral bands and an uneven number of distributions, which makes it suitable for farmland classification studies. The Pavia University (Figure 12) dataset packages 103 spectral bands covering the spectral range from 430 nm to 860 nm. There are nine feature classes, such as roads, buildings, grass, and trees. It is characterized by a moderate amount of spectra, mainly concentrated in the visible and near-infrared ranges, with a high resolution, which is suitable for man-made buildings and urban landscapes. The Salinas (Figure 13) dataset packages 224 spectral bands, covering the spectral range from 400 nm to 2500 nm. There are sixteen feature classes, mainly for different crops, such as lettuce, tomatoes, and vineyards. It is characterized by a wide spectral range, which helps to distinguish crop species, and a more homogeneous data classification, which is suitable for agricultural applications.

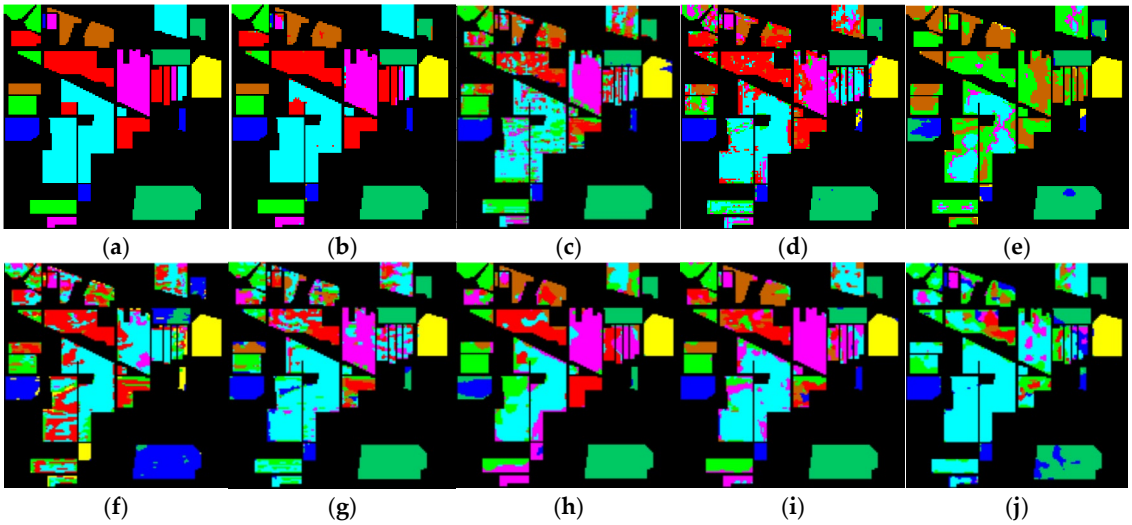

Figure 11. Indian Pines: (**a**) truth plot; (**b**) GRCNet; (**c**) ResNet-18; (**d**) ResNet-50; (**e**) Fast-2DCNN; (**f**) Fast-3DCNN; (**g**) DCGAN; (**h**) U-Net; (**i**) GCN; (**j**) KNN.

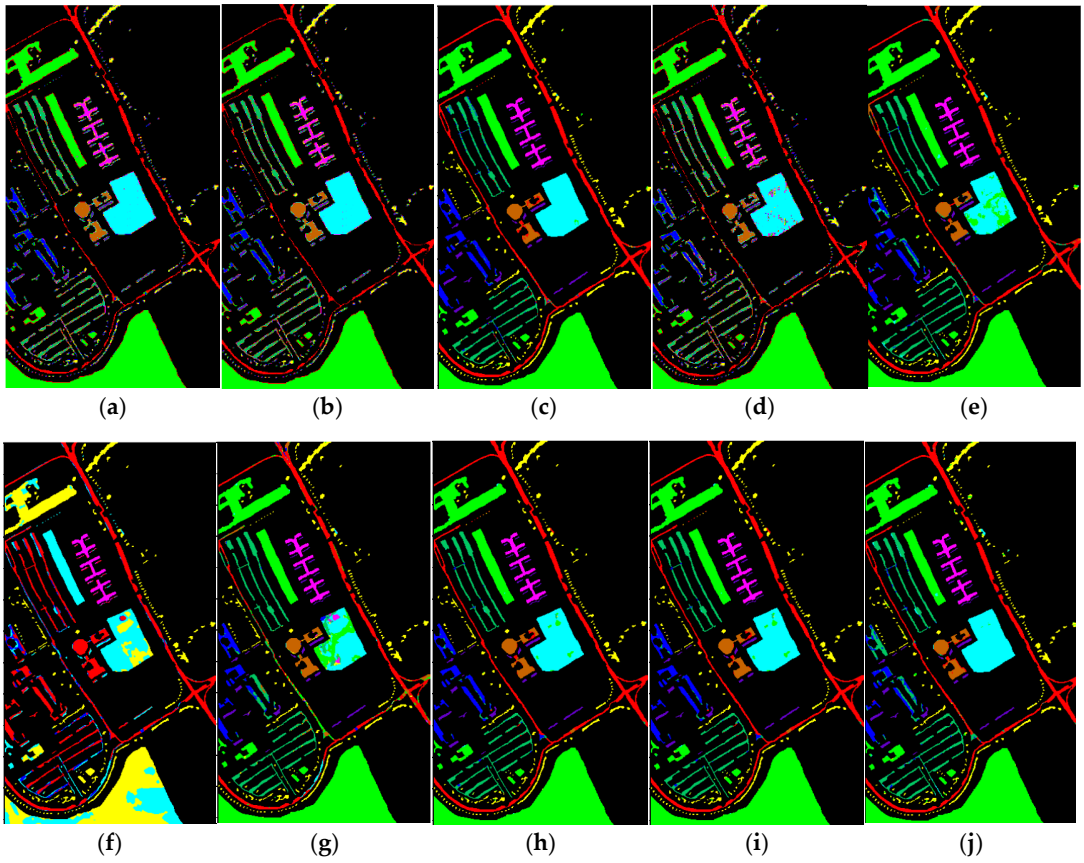

Figure 12. Pavia University: (**a**) truth plot; (**b**) GRCNet; (**c**) ResNet-18; (**d**) ResNet-50; (**e**) Fast-2DCNN; (**f**) Fast-3DCNN; (**g**) DCGAN; (**h**) U-Net; (**i**) GCN; (**j**) KNN.

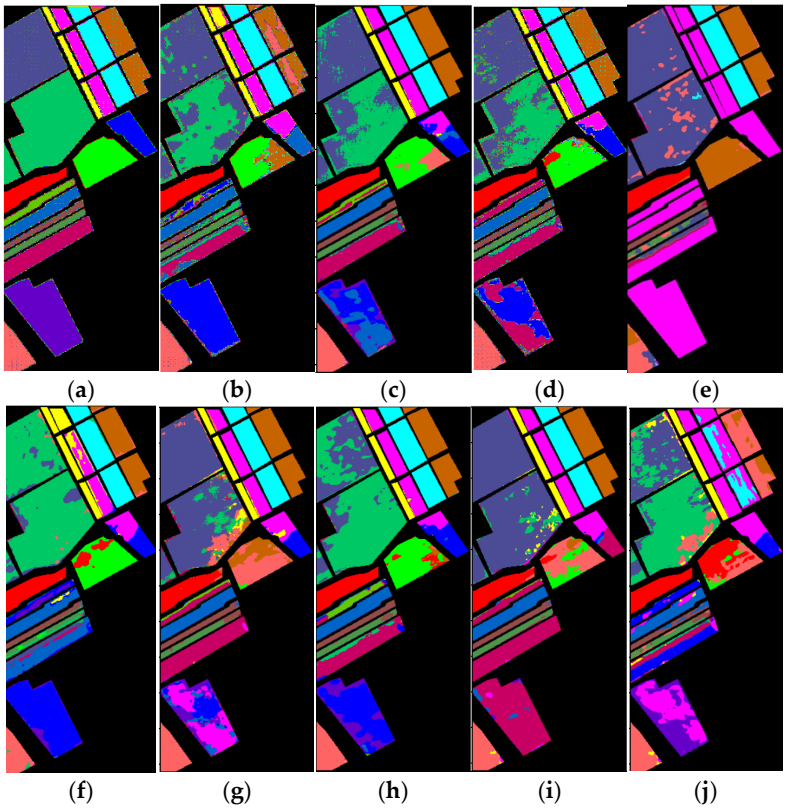

Figure 13. Salinas: (**a**) truth plot; (**b**) GRCNet; (**c**) ResNet-18; (**d**) ResNet-50; (**e**) Fast-2DCNN; (**f**) Fast-3DCNN; (**g**) DCGAN; (**h**) U-Net; (**i**) GCN; (**j**) KNN.

GRCNet performs best in the Indian (Figure 11) dataset with AA of 98.46%, OA of 98.51%, and F1 score of 99.25. Due to more spectral band information in these data, Fast-2DCNN (Figure 11e) and KNN (Figure 11j) are not able to use the complex data efficiently, and Fast-2DCNN (Figure 11e) has low accuracy due to insufficient layers to learn enough features. The remaining six classes of algorithms are defective in different degrees, such as low computational accuracy and poor model stability. The specific parameters are shown in Table 5.

Table 5. Experimental results of Indian Pines.

Mod	AA (%)	OA (%)	F1	Times (s)
GRCNet	98.46	98.51	99.25	621
ResNet-18	73.32	70.19	82.49	105
ResNet-50	71.29	71.66	83.49	123
Fast-2DCNN	51.18	42.31	59.46	96
Fast-3DCNN	46.51	45.14	62.20	163
DCGAN	71.96	71.61	83.45	51
U-Net	76.50	75.13	85.80	119
GCN	86.19	80.90	89.44	45
KNN	58.58	60.23	75.19	39

GRCNet performs best in the PaviaU (Figure 12) dataset with 99.71% AA, 99.81% OA, and 99.90 F1 scores. This dataset has a high resolution (1.3 M), and the spectral bands are mainly concentrated in the visible and near-infrared, with a clear distinction between the features, which contributes to the model's classification ability, so that the distinction between different types of methods on this dataset is not obvious. However, Fast-2DCNN has low accuracy because it does not learn the grass data (green color). The specific parameters are shown in Table 6.

Table 6. Experimental results of Pavia University.

Mod	AA (%)	OA (%)	F1	Times (s)
GRCNet	99.71	99.81	99.90	2439
ResNet-18	95.56	97.52	98.74	178
ResNet-50	95.02	98.81	98.38	475
Fast-2DCNN	41.31	32.90	49.51	44
Fast-3DCNN	91.07	93.41	96.59	536
DCGAN	88.33	89.56	94.49	229
U-Net	96.84	98.02	99.00	411
GCN	97.81	98.37	99.18	169
KNN	91.33	95.50	97.69	174

GRCNet had the best performance in the PaviaU (Figure 13) dataset with 88.12% AA, 78.43% OA, and 88.64 F1 scores. The dataset had a large number of crop species, but some of the vegetation spectral data were similar, and the accuracy was generally low. Compared with the other eight algorithms, GRCNet (Figure 13b) has more favorable segmentation results, but the running time is longer due to the large amount of computing power of the ViT module. The specific parameters are shown in Table 7.

Table 7. Experimental results of Salinas.

Mod	AA (%)	OA (%)	F1	Times (s)
GRCNet	88.12	78.43	88.64	3391
ResNet-18	83.90	76.98	86.97	323
ResNet-50	81.21	74.68	85.49	594
Fast-2DCNN	43.86	40.08	57.22	29
Fast-3DCNN	63.17	59.73	74.79	621
DCGAN	72.72	61.08	75.83	216
U-Net	83.70	77.58	87.50	533
GCN	70.57	58.53	73.84	163
KNN	59.02	58.95	74.17	158

3.3. Ablation Experiments

Ablation experiments are used in deep learning to evaluate the impact of different modules on the network. Net is the base model without the addition of GroupConv, RefConv, and the attention mechanism with ViT network, and no feature fusion is added, only linear convolutional operations. After adding the sub-GroupConv, the accuracy rises slightly by 2% on average, but the running time decreases, and the oscillation interval at $\pm$5% is obvious during the experiment, with poor stability. RefConv and the attention mechanism after feature fusion are added, and the accuracy improves significantly, with the AA rising by 6%, the OA rising by 5%, the F1 score rising by 2%, and the Kappa coefficient rising by 6%. by 6%. After adding ViT-Base, the accuracy is improved again, AA increased by 3%, OA increased by 3%, F1 score increased by 2%, Kappa coefficient increased by 4%, and it performs well in multiple experiments with high stability and the error of multiple experiments does not exceed $\pm$ 1%. The specific ablation experiment data are shown in Table 8.

Table 8. Results of ablation test.

Net	Group	Ref	Attention	ViT	AA	OA	F1	Kappa	Times (s)
√	×	×	×	×	83.48	84.77	91.76	0.81	648
√	√	×	×	×	84.62	86.73	93.18	0.83	631
√	√	√	×	×	86.44	88.56	93.64	0.86	638
√	√	√	√	×	90.03	91.94	95.01	0.89	979
√	√	√	√	√	92.84	94.59	97.22	0.93	3140

√ means used, × means not used.

Experiments were conducted using modules with different parameters. Firstly, experiments were performed with two different types of groups. When the group size was set to 64, the computation time increased significantly. With a group size of 16, the model's accuracy did not improve. The results indicate that the best performance is achieved when the group size is 32, as it improves model accuracy while controlling computational costs. For the Ref module, which allows adjustment of the kernel size, the best performance was observed with a 3×3 kernel when both the input and output channels were 64. A 5×5 kernel performed better for larger channel sizes, but switching to a larger kernel significantly increased computation time. The ViT module has five variants, ranging from smallest to largest. In this experiment, ViT-Base was used, which is a medium-sized model and serves as a standard version more suitable for UAV hyperspectral data. This is because ViT-Large has 307 M parameters, requiring immense computational resources, while ViT-Small has too few layers to effectively handle hyperspectral image recognition tasks.

4. Discussion

In recent years, near-surface grass hyperspectral identification and classification tasks are mainly captured by UAVs with hyperspectral imagers, but near-surface grass hyperspectral identification and classification tasks face the problems of low accuracy and high cost, and in addition, the existing research on near-surface hyperspectral identification by UAVs is still relatively scarce. Therefore, in this study, the UAV grass hyperspectral dataset Hohygrass-13 is created, and a GRCNet network model is proposed, which improves the grouped convolution module using the feature fusion method, improves the ordinary convolution block with the attention mechanism using the feature fusion method, and adds the ViT-Base module to enhance the global feature extraction. Finally, the effectiveness of the GRCNet network is verified using several comparison methods.

In this study, a hyperspectral imager carried by a UAV was used to take a sweeping photo of the grassland, collect hyperspectral images, and establish a hyperspectral dataset of grassland containing four categories, Hohygrass. In order to minimize the influence of factors such as ambient light and machine noise, the hyperspectral images were preprocessed, and a Gaussian filter was used for the noise reduction process to reduce the noise and make the images smoother and clearer. Due to the high number of hyperspectral data channels, which will increase the difficulty and training time of model training, the dimensionality is reduced using principal component analysis (PCA), which retains most of the important information and reduces the dimensionality of the data by extracting the main components. Subsequently, a neural network model called GRCNet was constructed, which mainly consists of the GC module and the RDC module, which uses the feature fusion method to improve the neural network and introduces the Refconv module and the attention mechanism.

Using GRCNet in the self-built hyperspectral dataset Hohygrass-13, the average accuracy (AA) reaches 92.84%, the overall accuracy (OA) reaches 94.59%, the F1 score reaches 97.22, and the Kappa coefficient reaches 0.93; the four evaluation indexes have been improved by more than 10% compared with the other eight control methods; in order to further demonstrate the GRCNet algorithm, three different kinds of public hyperspectral datasets are used, and eight methods are used for comparison experiments to prove the robustness of GRCNet. The experimental results in the three datasets show that the GRCNet network model has the highest experimental results, with the highest prediction accuracy reaching 99.81%. Finally, ablation experiments are used to demonstrate the effectiveness of each module. The results show that GRCNet has significant advantages in processing hyperspectral data and can provide high classification accuracy and stability. Through experimental verification, GRCNet performs well in the hyperspectral grass recognition classification task, solves the problems of low training accuracy and high computation time, and provides an efficient and accurate solution for this field.

Author Contributions: Conceptualization, Y.L. (Yuke Liu), Y.L. (Yilei Liu), and X.P.; methodology, Y.L. (Yuke Liu); software, Y.L. (Yuke Liu); validation, Y.L. (Yilei Liu); data curation, Y.L. (Yilei Liu); writing—original draft preparation, Y.L. (Yuke Liu); writing—review and editing, Y.L. (Yilei Liu); project administration, Y.L. (Yuke Liu) and Y.L. (Yilei Liu); funding acquisition, X.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Inner Mongolia Natural Science Foundation Project, "Multi modal grass image feature extraction and recognition" (grant number: 2023LHMS06014).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhang, Y.; Wang, T.; You, Y.; Wang, D.; Lu, M.; Wang, H. A Novel Estimation Method of Grassland Fractional Vegetation Cover Based on Multi-Sensor Data Fusion. *Comput. Electron. Agric.* **2024**, *225*, 109310. [CrossRef]
2. Shuai, L.; Li, Z.; Chen, Z.; Luo, D.; Mu, J. A Research Review on Deep Learning Combined with Hyperspectral Imaging in Multiscale Agricultural Sensing. *Comput. Electron. Agric.* **2024**, *217*, 108577. [CrossRef]
3. Liu, Y.; Fan, Y.; Feng, H.; Chen, R.; Bian, M.; Ma, Y.; Yue, J.; Yang, G. Estimating Potato Above-Ground Biomass Based on Vegetation Indices and Texture Features Constructed from Sensitive Bands of UAV Hyperspectral Imagery. *Comput. Electron. Agric.* **2024**, *220*, 108918. [CrossRef]
4. Ram, B.G.; Oduor, P.; Igathinathane, C.; Howatt, K.; Sun, X. A Systematic Review of Hyperspectral Imaging in Precision Agriculture: Analysis of Its Current State and Future Prospects. *Comput. Electron. Agric.* **2024**, *222*, 109037. [CrossRef]
5. Zhang, Y.; Gao, J.; Cen, H.; Lu, Y.; Yu, X.; He, Y.; Pieters, J.G. Automated Spectral Feature Extraction from Hyperspectral Images to Differentiate Weedy Rice and Barnyard Grass from a Rice Crop. *Comput. Electron. Agric.* **2019**, *159*, 42–49. [CrossRef]
6. Barbedo, J.G.A. A Review on the Combination of Deep Learning Techniques with Proximal Hyperspectral Images in Agriculture. *Comput. Electron. Agric.* **2023**, *210*, 107920. [CrossRef]
7. Zhao, X.H. Research on Hyperspectral Image Recognition Algorithm for Grassland Pasture Based on Machine Learning. Master's Thesis, Inner Mongolia Agricultural University, Hohhot, China, 2021. [CrossRef]
8. Ma, J. Identification and Retrieval of Desert Grassland Plants in Yili Based on Multi-Source Remote Sensing. Master's Thesis, Xinjiang Agricultural University, Urumqi, China, 2022. [CrossRef]
9. He, W.Q.; Li, Z.K.; Fang, Z.Q. Hyperspectral Image Domain Adaptation Classification Based on Transformer. *Laser J.* **2024**, 1–12. Available online: http://kns.cnki.net/kcms/detail/50.1085.TN.20240620.0843.002.html (accessed on 1 July 2024).
10. Lü, H.H.; Bai, S.; Zhang, H. Hyperspectral Image Classification Based on 3D Dilated Convolution and Graph Convolution. *Optoelectron. Laser* **2024**, *35*, 971–980. [CrossRef]
11. Pi, W.Q. Research on the Identification and Classification of Grassland Degradation Indicators Based on UAV Hyperspectral Remote Sensing. Ph.D. Thesis, Inner Mongolia Agricultural University, Hohhot, China, 2021. [CrossRef]
12. Wu, N.; Bao, Y.; Bu, R.; Tu, B.; Tao, S.; Bao, Y.; Jin, E. Identification of Typical Grassland Degradation Indicator Species Based on UAV Hyperspectral Remote Sensing. Remote Sens. *Technol. Appl.* **2024**, *39*, 248–258.
13. Zhang, J.; Lu, M.X.; Li, J.K. A Denoising Method for High-Noise Images Based on Residual Dense Convolutional Autoencoder. *Comput. Sci.* **2024**, *51* (Suppl. 1), 567–573.
14. Li, H.; Zhao, Q.; Cui, C.Z. Star Spectrum Classification Algorithm Based on CNN and LSTM Hybrid Deep Model. *Spectrosc. Spectral Anal.* **2024**, *44*, 1668–1675.
15. Su, Z.; Wang, Y.; Xu, Q.; Gao, R.; Kong, Q. LodgeNet: Improved rice lodging recognition using semantic segmentation of UAV high-resolution remote sensing images. *Comput. Electron. Agric.* **2022**, *196*, 106873. [CrossRef]
16. Liu, Z.Y.; Wu, H.F.; Huang, J.F. Application of neural networks to discriminate fungal infection levels in rice panicles using hyperspectral reflectance and principal components analysis. *Comput. Electron. Agric.* **2010**, *72*, 99–106. [CrossRef]
17. Xia, Z.; Pan, X.; Song, S.; Li, L.E.; Huang, G. Vision transformer with deformable attention. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 4794–4803. [CrossRef]
18. Cai, Z.; Ding, X.; Shen, Q.; Cao, X. Refconv: Re-parameterized refocusing convolution for powerful convnets. *arXiv* **2023**, arXiv:2310.10563.
19. Yu, Y.; Xiong, Y.; Huang, W.; Scott, M.R. Deformable siamese attention networks for visual object tracking. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 6728–6737. [CrossRef]
20. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
21. Wang, W.; Dai, J.; Chen, Z.; Huang, Z.; Li, Z.; Zhu, X.; Hu, X.; Lu, T.; Lu, L.; Li, H.; et al. InternImage: Exploring large-scale vision foundation models with deformable convolutions. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 14408–14419. [CrossRef]
22. Liu, N.; Long, Y.; Zou, C.; Niu, Q.; Pan, L.; Wu, H. Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 3225–3234. [CrossRef]

23. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722. [CrossRef]
24. Xie, C.; Zhu, H.; Fei, Y. Deep coordinate attention network for single image super-resolution. *IET Image Process.* **2022**, *16*, 273–284. [CrossRef]
25. Mohammadizadeh, M.; Mozhdehi, A.; Ioannou, Y.; Wang, X. Meta-GCN: A dynamically weighted loss minimization method for dealing with the data imbalance in graph neural networks. *arXiv* **2023**, arXiv:2406.17073. [CrossRef]
26. Bo, C.; Lu, H.; Wang, D. Spectral-spatial K-nearest neighbor approach for hyperspectral image classification. *Multimed. Tools Appl.* **2018**, *77*, 10419–10436. [CrossRef]
27. Bello, I.; Fedus, W.; Du, X.; Cubuk, E.D.; Srinivas, A.; Lin, T.-Y.; Shelens, J.; Zoph, B. Revisiting ResNets: Improved training and scaling strategies. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 22614–22627.
28. Theckedath, D.; Sedamkar, R.R. Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks. *SN Comput. Sci.* **2020**, *1*, 79. [CrossRef]
29. Wu, Q.; Chen, Y.; Meng, J. DCGAN-based data augmentation for tomato leaf disease identification. *IEEE Access* **2020**, *8*, 98716–98728. [CrossRef]
30. Yu, L.; Yang, X.; Chen, H.; Qin, J.; Heng, P.A. Volumetric ConvNets with Mixed Residual Connections for Automated Prostate Segmentation from 3D MR Images. *Proc. AAAI Conf. Artif. Intell.* **2017**, *31*. [CrossRef]