Intelligent system for hyperspectral image processing based on generative adversarial networks*

Victor Sineglazov[1,†] and Oleksii Shcherban[1,*,†]

[1] State University "Kyiv Aviation Institute", 1 Liubomyra Huzara ave., 03058 Kyiv, Ukraine

Abstract

This paper addresses the challenge of hyperspectral image classification under conditions of limited labeled data and class imbalance. An improved method based on the AC-WGAN-GP architecture is proposed to enhance classification performance through dataset augmentation with synthetic samples generated via class-aware sampling and label embedding. The generator, discriminator, and classifier were modified accordingly, resulting in high classification accuracy on standard benchmark datasets. The proposed approach demonstrates strong potential for applications in remote sensing and precision agriculture.

Keywords

hyperspectral images, generative adversarial networks, AC-WGAN-GP, classification, synthetic samples, class-aware sampling, label embedding, precision agriculture

1. Introduction

The agricultural sector faces challenges from population growth, climate change, and limited resources. Precision agriculture tackles these issues using advanced technologies to improve yields and resource management [1].

Hyperspectral imaging (HSI) stands out for capturing detailed spectral profiles that reveal early signs of crop stress, such as nutrient deficiencies or disease [2, 3]. Combined with thermal imaging, it enhances monitoring during critical growth phases.

Traditional field-based approaches are labor-intensive and unsuitable for real-time analysis [4, 5]. In contrast, AI models efficiently process high-dimensional data from satellites, UAVs, and ground platforms—each with trade-offs in resolution, cost, and scalability [6–9].

Satellites offer broad coverage but lower resolution [10]; UAVs provide detailed data but have limited range and higher cost [11]; ground systems offer precision but poor scalability [5].

Despite these limitations, satellite sensing—when combined with AI and sensor fusion—offers a scalable solution for large-scale precision agriculture.

2. Hyperspectral and multispectral imaging

Hyperspectral imaging (HSI) captures hundreds of contiguous spectral bands across the visible to SWIR range, enabling fine-grained detection of crop conditions such as nutrient deficiency, disease, and water stress [2, 12]. In contrast, multispectral imaging (MSI) uses fewer (3–15) broader bands targeting key wavelengths, supporting vegetation indices like NDVI and EVI [5].

HSI is valuable in precision agriculture due to its ability to distinguish visually similar crops and detect early plant stress [13, 14]. However, its high dimensionality complicates storage and analysis, often requiring PCA, ICA, or autoencoders. Moreover, deep learning models demand large annotated datasets, which are costly to produce [15]. Spectral similarity among classes further complicates classification, often mitigated via spatial context or data augmentation [7].

* CSDP'2025: Cyber Security and Data Protection, July 31, 2025, Lviv, Ukraine

* Corresponding author.

† These authors contributed equally.

svm@kai.edu.ua (V. Sineglazov); oleksiishcherbanrw@gmail.com (O. Shcherban)

0000-0002-3297-9060 (V. Sineglazov); 0009-0004-8702-4917 (O. Shcherban)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

CEUR-WS.org/Vol-4042/paper13.pdf

HSI combines spectroscopy and imaging, producing a 3D cube with rich spectral-spatial data [16]. Pixel-wise classification supports tasks like target detection, change monitoring, and crop mapping [17, 18]. While early methods used SVM or KNN, modern approaches rely on CNNs, 3D-CNNs, and hybrid networks for improved performance [8, 9, 19].

Data collection is supported by platforms like satellites (e.g., Sentinel-2, Landsat), which offer wide coverage but lower resolution and weather limitations [6, 10]; UAVs, which provide high resolution but limited scalability [11, 20–24]; and open-access datasets (e.g., HISUI) that drive development of HSI classification methods. Combining platforms enables flexible monitoring tailored to agricultural needs.

3. Using generative adversarial networks for HSI classification tasks

Generative adversarial networks (GANs) are effective in generating structured data, including hyperspectral images, helping reduce dependence on large labeled datasets—a key benefit in data-scarce settings [15, 25]. This study utilizes the AC-WGAN-GP architecture, which combines class conditioning (AC-GAN), Wasserstein loss for stability, and gradient penalty regularization [20, 26].

The model generates realistic synthetic samples with specified labels, preserving diversity even with limited training data. Architectural enhancements further address class imbalance and spectral similarity, improving classification performance in both HSI and MSI contexts—relevant for precision agriculture [18, 27]. Such hybrid designs can be generalized across AI systems for improved adaptability and robustness [28–30].

3.1. AC-WGAN-GP architecture and its characteristics

The AC-WGAN-GP architecture includes a generator (G), discriminator (D), and auxiliary classifier (C). The generator receives Gaussian noise, spectral features (e.g., PCA), and class labels (one-hot or embedded) to produce synthetic spectral samples. Its structure comprises 1D transposed convolutions (Deconv1D) with ReLU activations and a final Tanh layer, using batch normalization for training stability [25, 31]. Related research on the detection of synthetic visual content highlights parallels between hyperspectral image generation and deepfake detection. For instance, recent studies have explored neural network-based systems for identifying biometric image manipulations [32, 33], demonstrating architectural strategies that can be adapted to improve the robustness of GAN-based HSI generation and classification.

Figure 1 shows the data flow and interaction among components, each optimized within a unified training framework.

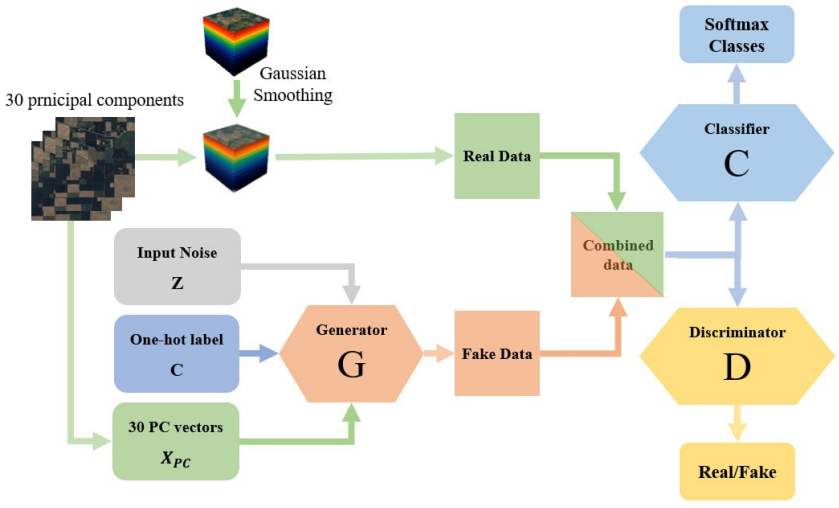

Figure 1: Architecture AC-WGAN-GP

The discriminator, built from 1D Conv layers with Leaky-ReLU activations, evaluates sample authenticity. It outputs a linear value and uses Wasserstein loss with gradient penalty, ensuring stable training under distribution shifts [26].

The auxiliary classifier performs multi-class classification on both real and generated samples. It consists of a Conv layer, flattening, and a fully connected Softmax output. Batch normalization is excluded to preserve spectral sensitivity. It also guides the generator to produce correctly labeled data [20, 34].

Together, G, D, and C form a closed feedback loop: the generator is informed by both discriminator and classifier, promoting realism and class accuracy—crucial for imbalanced or overlapping spectral classes [15, 18, 35].

3.2. Training process and loss functions

The AC-WGAN-GP model trains the generator (G), discriminator (D), and auxiliary classifier (C) in an alternating fashion to ensure stable and controlled generation. Batch Normalization (BN) accelerates convergence, while Gradient Penalty (GP) enforces Lipschitz continuity for stable training [25, 26].

The discriminator uses the Wasserstein loss with GP:

$$L_D = \mathbb{E}_{\tilde{x}\sim p_g} D(\tilde{x}) - \mathbb{E}_{x\sim p(x)} D(x) + \lambda \mathbb{E}_{\hat{x}\sim p(\hat{x})}\left[\left(\left\|\nabla_{\hat{x}} D(\hat{x})\right\|_2 - 1\right)^2\right]. \quad (1)$$

This balances the Wasserstein distance and gradient regularization to prevent mode collapse.

The generator minimizes:

$$L_G = -\mathbb{E}_{\tilde{x}\sim p_g}[D(\tilde{x})] + \mathbb{E}[\log p(C = c\,|\,\tilde{x})] \quad (2)$$

combining realism (discriminator) and class consistency (classifier).

The classifier is trained on both real and synthetic data:

$$L_C = \mathbb{E}[\log p(C = c\,|\,x)] + \mathbb{E}[\log p(C = c\,|\,\tilde{x})] \quad (3)$$

ensuring inter-class discrimination even under spectral overlap and imbalance.

4. Hyperspectral image processing using AC-WGAN-GP

HSI processing faces the “curse of dimensionality”: hundreds of spectral channels increase training complexity and risk overfitting, especially with limited labeled data [15, 25].

AC-WGAN-GP addresses this by generating synthetic samples that preserve spectral and semantic class properties. Conditional generation via class labels and classifier guidance improves data diversity and reduces overfitting [18, 20, 26]. Labeled HSI data is scarce due to costly expert annotation, leading to class imbalance. AC-WGAN-GP augments datasets—particularly rare classe —improving balance and training efficacy [15, 27].

Spectral overlap between classes causes classification ambiguity. The model maintains semantic consistency through classifier feedback, enhancing discrimination [14, 25]. The improved AC-WGAN-GP shows gains in average accuracy (AA) and Cohen’s kappa (κ) on benchmarks like Indian Pines, Salinas, and Pavia University [18]. Mode collapse is mitigated through gradient penalty and auxiliary classification, ensuring diversity and training stability [26, 36].

In summary, AC-WGAN-GP effectively processes hyperspectral data under limited-label conditions, enhancing classification through synthetic, spectrally valid augmentation.

5. Problem formulation of hyperspectral image classification using AC-WGAN-GP

In hyperspectral image classification tasks involving generative models, evaluation metrics play a crucial role in objectively comparing model performance and quantifying improvements resulting from architectural modifications. In this study, we employ three widely adopted metrics: Overall Accuracy (OA), Average Accuracy (AA), and the Cohen's Kappa coefficient (κ), which are standard in hyperspectral classification research [18, 20, 25].

While global metrics like OA, AA, and κ assess overall performance, per-class metrics—Precision, Recall, and F1-score—reveal how well individual classes are classified.

Overall Accuracy (OA) is a standard metric in HSI classification that measures the proportion of correctly predicted samples among all test samples:

$$OA = \frac{1}{N}\sum_{i=1}^{\bar{C}} h_{ii}, \tag{4}$$

where N is the total number of test samples, $\bar{C}$ is the number of classes, and h_{ii} represents correctly classified samples of class i (confusion matrix diagonal).

Average Accuracy (AA) evaluates classification performance across all classes equally, regardless of class size. It is calculated as the mean of per-class accuracies:

$$AA = \frac{1}{\bar{C}}\sum_{i=1}^{\bar{C}} \frac{h_{ii}}{N_i}, \tag{5}$$

where $\bar{C}$ is the number of classes, h_{ii} the correctly classified samples for class i, and N_i the total test samples in class i.

The Kappa coefficient (κ) measures agreement between predicted and true labels while accounting for chance. Unlike OA, it reflects class distribution, making it suitable for imbalanced datasets [20, 25]. It is computed as:

$$\kappa = \frac{N\sum_{i=1}^{\bar{C}} h_{ii} - \sum_{i=1}^{\bar{C}} (h_{i+} \cdot h_{+i})}{N^2 - \sum_{i=1}^{\bar{C}} (h_{i+} \cdot h_{+i})} \tag{6}$$

where N is the total number of test samples, $\bar{C}$ the number of classes, h_{ii} correct predictions, h_{i+} actual counts, and h_{+i} predicted counts per class.

6. Proposed method

The improved AC-WGAN-GP retains the classical conditional GAN structure comprising a generator (G), a discriminator (D), and an auxiliary classifier (C), but introduces targeted modifications to address class imbalance, spectral overlap, mode collapse, and training instability (Figure 2).

A revised training strategy complements the architecture. Conditional samples are generated using class embeddings and PCA vectors, followed by clustering in spectral space—applied only to real training data. Synthetic samples closest to the cluster centers (measured by cosine similarity) are selected and merged with real data for classifier training.

Crucially, all processing is confined to training data, avoiding test set leakage. This improves evaluation rigor and reproducibility, ensuring fair performance assessment under realistic constraints.

6.1. Improved generator architecture G

In the improved AC-WGAN-GP, the generator synthesizes conditional hyperspectral samples by combining class and spectral information. The baseline design with Deconv1D layers, ReLU, and batch normalization suffered from mode collapse and weak control via one-hot labels (Figure 3).

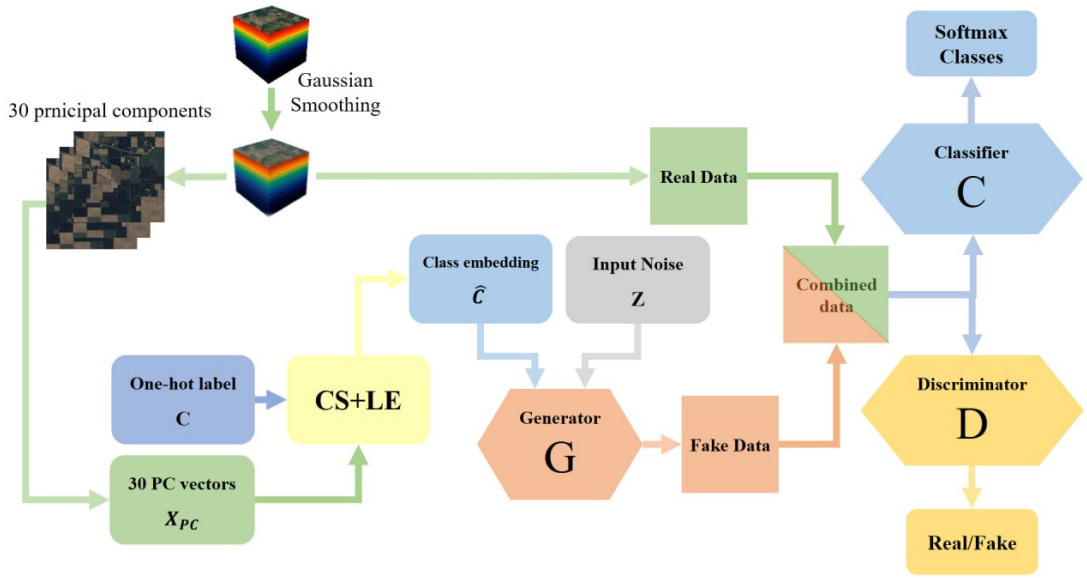

Figure 2: Enhanced architecture of the AC-WGAN-GP

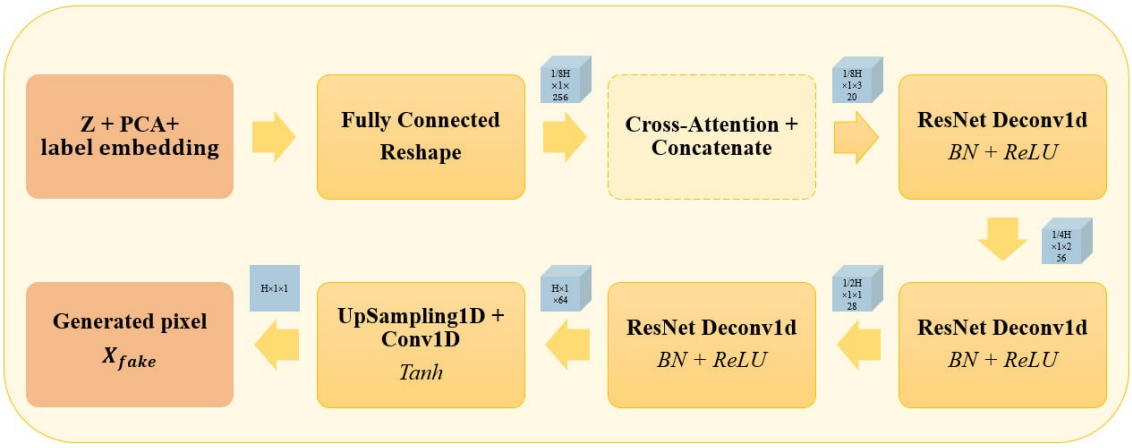

Figure 3: Improved architecture of the generator G

The updated architecture incorporates a Class-aware Sampling and Label Embedding (CS+LE) module, encoding labels into dense vectors and concatenating them with PCA features and Gaussian noise. This richer input better captures class identity and spectral variation.

To improve training stability and representation of minority classes, ResNet-style Deconv1D blocks with skip connections were added. A cross-attention mechanism aligns label and spectral embeddings with generator features, enhancing semantic coherence. Spectral Dropout is applied in intermediate layers to zero out entire spectral bands, improving robustness. The output is generated via UpSampling1D and a Conv1D layer with Tanh activation, ensuring normalized spectra. This architecture produces more diverse, class-consistent, and spectrally realistic samples.

6.2. Improved discriminator architecture D

In AC-WGAN-GP, the discriminator assesses how closely generated spectra resemble real hyperspectral data and provides feedback to the generator. The initial version used Conv1D layers with LeakyReLU and batch normalization, but the latter conflicts withWGAN-GP's requirement for sample independence, causing instability (Figure 4).

To address this, batch normalization was replaced with LayerNorm, which operates per sample and ensures stable training with gradient penalty. Each Conv1D layer is followed by LeakyReLU and LayerNorm for consistent processing. To mitigate mode collapse, a Minibatch Discrimination layer was added to detect similarity across samples, encouraging output diversity. The final output is a scalar critic score from Flatten and Dense(1), as required by the WGAN formulation. Overall, these changes improve training stability, prevent mode collapse, and enhance the model's ability to distinguish synthetic from real spectra.

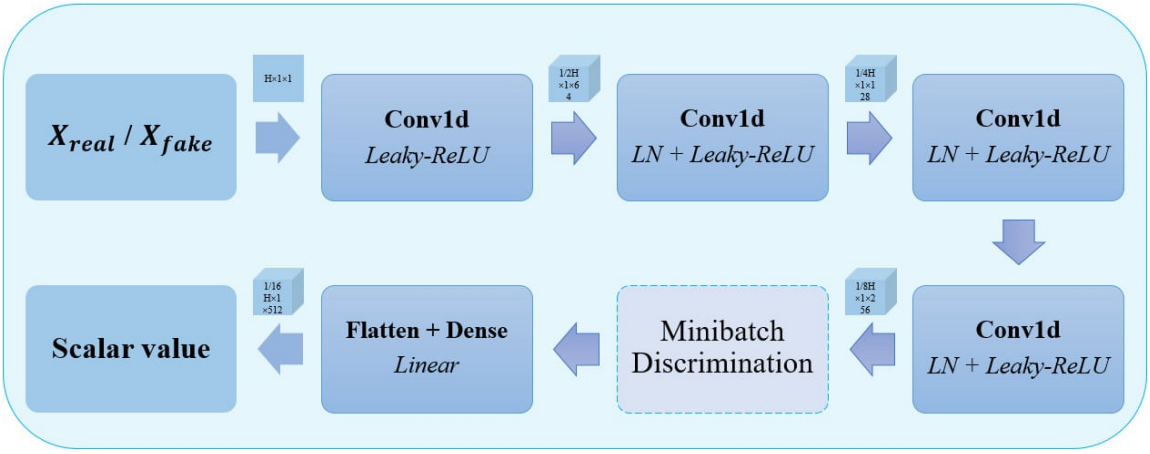

Figure 4: Improved architecture of the discriminator D

6.3. Improved architecture of the classifier C

The auxiliary classifier C in AC-WGAN-GP predicts class labels for real and synthetic samples and guides the generator. The initial design with a single Conv1D layer and Softmax output was too shallow to handle spectral similarity and rare classes (Figure 5).

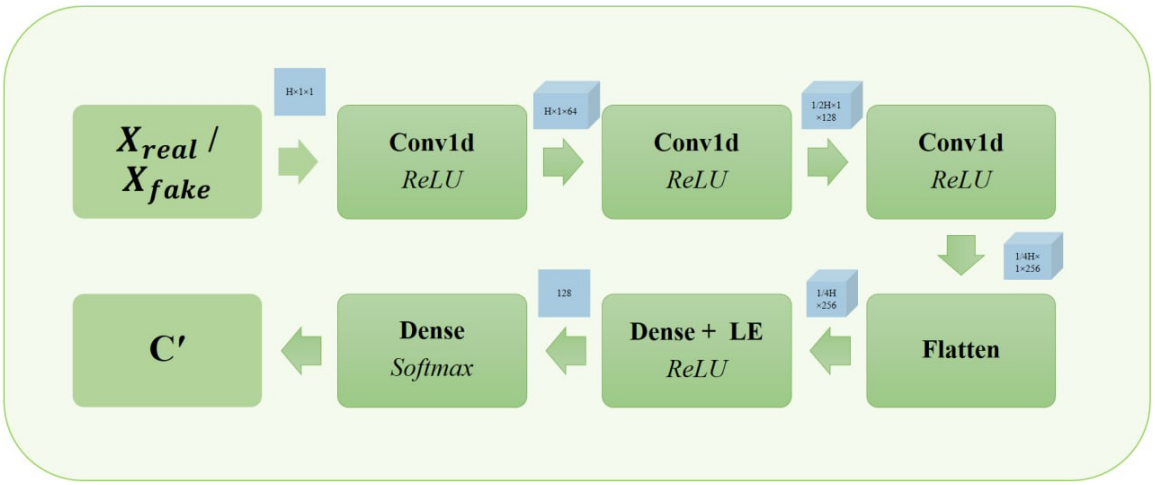

Figure 5: Improved architecture of the classifier C

To improve performance, the classifier was deepened with three Conv1D layers followed by ReLU and batch normalization. Class labels are passed as dense embeddings, encoding semantic relationships. Extracted features are flattened and concatenated with the label embedding, then processed by a dense layer (256 units) used for both classification and contrastive loss. This enhances class separation while maintaining intra-class compactness.

6.4. Loss functions of the improved AC-WGAN-GP

The improved AC-WGAN-GP architecture employs a multi-component loss formulation to enable efficient and stable training across all network modules. Each component of the loss not only incorporates core adversarial objectives common to classical GANs but also introduces domain-specific terms tailored to the challenges of hyperspectral classification.

6.4.1. Generator loss

Unlike in traditional GANs, the generator in AC-WGAN-GP is optimized not only through adversarial feedback from the discriminator but also by enforcing alignment with class conditions and spectral context.

The basic Wasserstein loss component for the generator is given by:

$$L_{WGAN} = -\mathbb{E}_{z,c}[D(G(z,c))], \tag{7}$$

where: z denotes the latent noise vector; c is the conditional class label; $G(z, c)$ is the generated spectral sample; $D(G(z, c))$ is the "realism" score assigned by the discriminator.

1. Cosine Similarity with PCA Vectors (Cosine PCA Loss)—ensures that the generated spectrum aligns with the average PCA vector of its target class:

$$L_{PCA} = \mathbb{E}[1 - \cos(\hat{x}, x_{PCA})]. \tag{8}$$

2. Cosine Alignment Loss—enforces the classifier's internal feature representation f to align with the class embedding vector e:

$$L_{align} = \mathbb{E}[1 - \cos(f, e)]. \tag{9}$$

3. Categorical Cross-Entropy—penalizes the generator if the classifier fails to recognize the correct class of a generated sample:

$$L_{ce} = \mathbb{E}_{z,c}[-\log P_{cls}(c \mid G(z,c))]. \tag{10}$$

The full generator loss is then defined as:

$$L_G = L_{WGAN} + \lambda_{PCA} \cdot L_{PCA} + \lambda_{align} \cdot L_{align} + \lambda_{ce} \cdot L_{ce}, \tag{11}$$

where λ_{PCA}, λ_{align}, and λ_{ce} are weighting coefficients that control the contribution of each loss component. These are tuned empirically based on data characteristics, class imbalance, and desired classification performance.

6.4.2. Discriminator loss

In the AC-WGAN-GP framework, the discriminator functions as a critic that estimates the divergence between real and generated spectral samples. Unlike in classical GANs, where the discriminator performs binary classification, the WGAN formulation approximates the Wasserstein distance between real and synthetic distributions.

The discriminator loss is defined as:

$$L_D = \mathbb{E}_{\tilde{x}\sim p_g} D(\tilde{x}) - \mathbb{E}_{x\sim p(x)} D(x) + \lambda \cdot L_{gp}, \tag{12}$$

$$L_{gp} = \mathbb{E}_{\hat{x}\sim p(\hat{x})}\left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2\right], \tag{13}$$

where: p_g is the distribution of generated samples (from the generator); p_{data} is the distribution of real training samples; $\tilde{x}$ is a generated spectrum $G(z, c)$; x is a real spectral sample; $\hat{x}$ is a linear interpolation between x and $\tilde{x}$; λ is a hyperparameter controlling the weight of the gradient penalty term L_{gp} that ensures 1-Lipschitz continuity.

6.4.3. Classifier loss

The auxiliary classifier in AC-WGAN-GP is responsible for both class prediction and learning discriminative features for regularization. Its loss function comprises several components aimed at maximizing classification accuracy while structuring the feature space.

1. Categorical Cross-Entropy (Class-Weighted) This standard classification loss is weighted to compensate for class imbalance:

$$L_{ce} = -\mathbb{E}_{x,y}[\omega_y \log P_{cls}(y|x)], \tag{14}$$

where x is the spectral sample, y is the true class label, $P_{\text{cls}}(y \mid x)$ is the predicted probability, and ω_y is the inverse class frequency weight.

2. Contrastive Loss This term promotes closeness of features from the same class and separation between features from different classes:

$$L_{contrast} = \mathbb{E}_{i\,j}\begin{cases} \|f_i - f_j\|^2 & , if\ y_i = y_j \\ max(0, \|f_i - f_j\| - \delta)^2 & , if\ y_i \neq y_j \end{cases}, \tag{15}$$

where f_i, f_j are feature vectors and δ is a margin parameter.

3. Cosine Alignment Loss Aligns the feature vector with the corresponding class embedding:

$$L_{align} = \mathbb{E}_{x,y}[1 - \cos(f(x), e_y)], \tag{16}$$

where $f(x)$ is the feature vector from the classifier and e_y is the embedding of class y.

4. Embedding Divergence Loss Regularizes class embeddings to prevent their collapse in latent space:

$$L_{\text{div}} = \sum_{i\neq j}\left(\frac{1}{\|e_i - e_j\|^2 + \varepsilon}\right), \tag{17}$$

where e_i, e_j are embeddings of different classes, and ε is a small positive constant to avoid division by zero.

Total Classifier Loss:

$$L_C = L_{ce} + \lambda_{\text{contrast}} \cdot L_{\text{contrast}} + \lambda_{align} \cdot L_{align} + \lambda_{\text{div}} \cdot L_{\text{div}}, \tag{18}$$

where $\lambda_{\text{contrast}}$, λ_{align}, λ_{div} are hyperparameters controlling the contribution of each regularization component. These are selected empirically based on task complexity and class imbalance.

7. Results

7.1. Experimental setup and execution specifics

All experiments were conducted using PyCharm Community Edition 2024.3.4 with Python 3.9 and the TensorFlow 2.19.0 framework. The development environment ran on Windows 10 and local machine specifications were as follows. Synthetic samples were generated in online mode without being saved to disk, reducing memory usage and preventing data duplication. Spectral vectors were reduced to $H = 30$ components using Principal Component Analysis (PCA). The training and test splits were performed with strict class separation, eliminating potential data leakage.

7.2. Analysis of incremental improvements in AC-WGAN-GP

Table 1 summarizes the stepwise impact of architectural enhancements in AC-WGAN-GP on hyperspectral image classification. Each modification (from the baseline to the inclusion of cosine alignment and embedding divergence losses) leads to progressive gains in OA, AA, and κ across all datasets.

The most notable improvement is observed on the challenging *Indian Pines* dataset, where introducing the CSLE module (Step 2) increases OA to 67.74%. Adding ResNet-style deconvolutions with Spectral Dropout (Step 3), followed by Minibatch Discrimination and Layer Normalization (Step 4), improves training stability—especially on *KSC*, where accuracy reaches 90.12%.

Incorporating Contrastive Loss (Step 5) and the final alignment-based loss terms (Step 6) yields the highest performance, notably OA = 90.55% on Salinas and κ = 63.63 on Indian Pines. These results confirm the effectiveness of the proposed architectural and loss-based enhancements.

Table 1
Impact of Incremental Architectural Improvements on Classification Performance (Training Ratio: 5%)

Step	Improvement	Salinas			Indian Pines			KSC		
		OA	AA	κ	OA	AA	κ	OA	AA	κ
1	Baseline AC-WGAN-GP	89.54	94.46	88.49	66.30	54.48	61.11	89.76	84.61	88.50
2	CSLE	89.73	94.78	88.46	67.74	56.61	63.10	89.81	84.26	88.64
3	ResNet + Spectral Dropout	89.76	94.83	88.54	67.89	56.82	63.28	90.05	85.53	89.18
4	Minibatch Disc. + LayerNorm	90.09	95.01	89.05	68.08	56.25	63.36	90.12	85.57	89.27
5	Contrastive Loss	90.44	95.13	89.45	68.10	56.31	63.56	90.16	86.18	89.42
6	Cosine Align. + Divergence	**90.55**	**95.20**	**89.50**	**68.60**	**58.09**	**63.63**	**90.62**	**86.30**	**89.55**

7.3. Analysis of results at different training set sizes

The classification performance was evaluated on three benchmark HSI datasets—Salinas, Indian Pines, and KSC—at varying training ratios. As shown in Table 2, all datasets demonstrate clear improvement in OA, AA, and κ with increased training size. Prediction maps for studied datasets are shown in Figures 6–8.

Per-class F1 analysis reveals that classes with stable and well-separated spectral signatures (e.g., *Stubble*, *Woods*, *Water*) achieve F1 scores above 90–95%. In contrast, classes with limited samples or high spectral overlap (e.g., *Oats*, *Corn*, *Oak Forest*) show lower F1 scores (40–70%).

The improved model achieves reliable classification even under extreme label scarcity, achieving over 90% OA on the Salinas and KSC datasets using only 5% of labeled data, and over 68% on Indian Pines—one of the most challenging datasets.

Table 2
Overall Classification Results Across Datasets

Dataset	Train Ratio (%)	OA (%)	AA (%)	κ
Salinas	1	88.70	92.90	87.41
	5	90.55	95.20	89.50
	10	91.27	95.36	90.07
Indian Pines	1	54.15	43.24	46.78
	5	68.60	58.09	63.63
	10	74.07	63.82	70.24
KSC	1	80.71	70.73	78.50
	5	90.62	86.30	89.55
	10	93.06	89.46	92.27

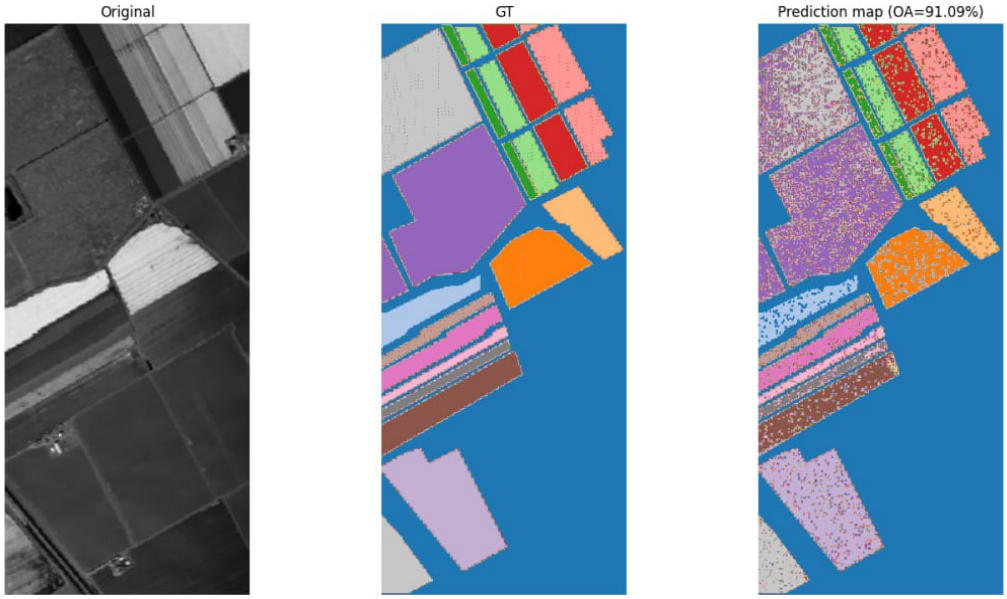

Figure 6: Prediction maps for Salinas dataset (5% training ratio). Figure shows spectral band, ground truth, and classification result

Figure 7: Prediction maps for Indian Pines dataset (5% training ratio). Figure shows spectral band, ground truth, and classification result

Figure 8: Prediction maps for KSC dataset (5% training ratio). Figure shows spectral band, ground truth, and classification result

Conclusions

This study presented an enhanced AC-WGAN-GP architecture for conditional hyperspectral sample generation, addressing class imbalance, spectral overlap, and training instability. Improvements include class-aware sampling, label embeddings, PCA-based conditioning, ResNet deconvolutions, crossattention, and spectral dropout in the generator; layer normalization and minibatch discrimination in the discriminator; and weighted categorical, contrastive, and embedding divergence losses in the classifier.

The method demonstrated strong performance under limited data and imbalanced class distributions, with all evaluations performed without test data leakage. Future research will extend this approach to 3D hyperspectral data, semantic segmentation tasks.

Declaration on Generative AI

While preparing this work, the authors used the AI programs Grammarly Pro to correct text grammar and Strike Plagiarism to search for possible plagiarism. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, J. Chanussot, Hyperspectral Remote Sensing Data Analysis and future Challenges, IEEE Geoscience and Remote Sensing Magazine 1 (2013). doi:10.1109/MGRS.2013.2244672

[2] P. Ghamisi, J. Plaza, Y. Chen, J. Li, A. Plaza, Advanced Spectral Classifiers for Hyperspectral Images: A Review, IEEE Geoscience and Remote Sensing Magazine 5 (2017).

[3] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, F. Fraundorfer, Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources, IEEE Geoscience and Remote Sensing Magazine 5 (2017). doi:10.1109/MGRS.2017.2762307

[4] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, J. A. Benediktsson, Deep Learning for Hyperspectral Image Classification: An Overview, IEEE Transactions on Geoscience and Remote Sensing 57 (2019) 6690–6709. doi:10.1109/TGRS.2019.2907932

[5] B. Lu, P. D. Dao, J. Liu, Y. He, J. Shang, Recent Advances of Hyperspectral Imaging Technology and Applications in Agriculture, Remote Sensing 12 (2020) 2659. doi:10.3390/rs12162659

[6] B. G. Ram, P. Oduor, C. Igathinathane, K. Howatt, X. Sun, A Systematic Review of Hyperspectral Imaging in Precision Agriculture: Analysis of Its Current State and Future Prospects, Comput. Electron. Agriculture 222 (2024). doi:10.1016/j.compag.2024.109037

[7] H. Yu, B. Kong, Y. Hou, X. Xu, T. Chen, X. Liu, A Critical Review on Applications of Hyperspectral Remote Sensing in Crop Monitoring, Experimental Agriculture 58 (2022). doi:10.1017/S0014479722000278

[8] V. Sineglazov, A. Kot, Design of Hybrid Neural Networks of the Ensemble Structure, Eastern-European J. Enterprise Technol. 1 (2021) 31–45. doi:10.15587/1729-4061.2021.225301

[9] M. Zgurovsky, V. Sineglazov, E. Chumachenko, Classification and Analysis Topologies Known Artificial Neurons and Neural Networks, in: Artificial Intelligence Systems Based on Hybrid Neural Networks, vol. 904, Springer, Cham, 2021. doi:10.1007/978-3-030-48453-8_1

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Nets, in: Advances in Neural Information Processing Systems, 27, 2014. doi:10.48550/arXiv.1406.2661

[11] A. Odena, C. Olah, J. Shlens, Conditional Image Synthesis with Auxiliary Classifier Gans, arXiv, 2016. doi:10.48550/arXiv.1610.09585

[12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. C. Courville, Improved Training of Wasserstein Gans, in: Advances in Neural Information Processing Systems, 30, 2017, 5767–5777. doi:10.48550/arXiv.1704.00028

[13] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, arXiv, 2017. doi:10.48550/arXiv.1701.07875

[14] A. Abbas, S. Jain, M. Gour, S. Vankudothu, Tomato Plant Disease Detection using Transfer Learnin with C-GAN Synthetic Images, Comput. Electron. Agriculture 187 (2021). doi:10.1016/j.compag.2021.106279

[15] C. Sun, X. Zhang, H. Meng, X. Cao, J. Zhang, AC-WGAN-GP: Generating Labeled Samples for Improving Hyperspectral Image Classification with Small-Samples, Remote Sensing 14 (2022) 4910. doi:10.3390/rs14194910

[16] A. Radford, L. Metz, S. Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, arXiv, 2016. doi:10.48550/arXiv. 1511.06434

[17] Y. Zhan, D. Hu, Y. Wang, X. Yu, Semisupervised Hyperspectral Image Classification based on Generative Adversarial Networks, IEEE Geoscience and Remote Sensing Letters 15 (2018) 212–216. doi:10.1109/LGRS.2017.2780890

[18] Z. He, K. Xia, P. Ghamisi, Y. Hu, S. Fan, B. Zu, Hypervitgan: Semisupervised GAN with Transformer for HSI Classification, IEEE J. Selected Topics Appl. Earth Observations Remote Sensing 15 (2022). doi:10.1109/JSTARS.2022.3192127

[19] M. Zgurovsky, V. Sineglazov, E. Chumachenko, Classification and Analysis of Multicriteria Optimization Methods, in: Artificial Intelligence Systems Based on Hybrid Neural Networks, 904, 2021. doi:10.1007/978-3-030-48453-8_2

[20] Y. Huang, Z. Chen, J. Liu, Limited Agricultural Spectral Dataset Expansion based on Generative Adversarial Networks, Comput. Electron. Agriculture 215 (2023). doi:10.1016/j.Compag.2023.108385

[21] V. Sokolov, P. Skladannyi, A. Platonenko, Video Channel Suppression Method of Unmanned Aerial Vehicles, in: IEEE 41st Int. Conf. on Electronics and Nanotechnology (2022) 473–477. doi: 10.1109/ELNANO54667.2022.9927105

[22] P. Skladannyi, et al., Adaptive Methods for Embedding Digital Watermarks to Protect Audio and Video Images in Information and Communication Systems, in: Classic, Quantum, and Post-Quantum Cryptography, vol. 4016, 2025, 13-31.

[23] Y. Kostiuk, et al., Application of Statistical and Neural Network Algorithms in Steganographic Synthesis and Analysis of Hidden Information in Audio and Graphic Files, in: Classic, Quantum, and Post-Quantum Cryptography, vol. 4016, 2025, 45-65.

[24] N. Dovzhenko, et al., Research of UAV and Sensor Network Integration Features for Routing Optimization and Energy Consumption Reduction, in: Cybersecurity Providing in Information and Telecommunication Systems II, vol. 3826 (2024) 236-241.

[25] L. Zhu, Y. Chen, P. Ghamisi, J. A. Benediktsson, Generative Adversarial Networks for Hyperspectral Image Classification, IEEE Transactions on Geoscience and Remote Sensing, 56 (2018) 5046–5063. doi:10.1109/TGRS.2018.2805286

[26] J. Feng, N. Zhao, R. Shang, X. Zhang, L. Jiao, Self-Supervised Divide-and-Conquer Generative Adversarial Network for Classification of Hyperspectral Images, IEEE Transactions on Geoscience and Remote Sensing 60 (2022) 1–17. doi:10.1109/TGRS.2022.3202908

[27] Y. Lu, D. Chen, E. Olaniyi, Y. Huang, Gans for Image Augmentation in Agriculture: A Systematic Review, Comput. Electron. Agriculture 200 (2022). doi:10.1016/j.compag.2022.107208

[28] M. Zgurovsky, V. Sineglazov, E. Chumachenko, Formation of Hybrid artificial Neural Networks Topologies, in: Artificial Intelligence Systems Based on Hybrid Neural Networks, volume 904 of Studies in Computational Intelligence, Springer, Cham, 2021. doi:10.1007/978-3-030-48453-8_3

[29] V. M. Sineglazov, K. D. Riazanovskiy, O. I. Chumachenko, Multicriteria Conditional Optimization based on Genetic Algorithms, System Research and Information Technologies (2020). doi:10.20535/SRIT.2308-8893.2020.3.07

[30] A. V. Iatsyshyn, et al. Application of Augmented Reality Technologies for Education Projects Preparation, in: Cloud Technologies in Education, 2643, 2020, 134–160.

[31] M. Zaliskyi, R. Odarchenko, S. Gnatyuk, Y. Petrova, A. Chaplits, Method of Traffic Monitoring for DDoS Attacks Detection in e-Health Systems and Networks, in: Informatics & Data-Driven Medicine, 2255, 2018, 193–204.

[32] V. Dudykevych, H. Mykytyn, K. Ruda, The Concept of a Deepfake Detection System of Biometric Image Modifications based on Neural Networks, in: 2022 IEEE 3rd KhPI Week on Advanced technology: conference proceedings, 2022, 585–588.

[33] V. Dudykevych, S. Yevseiev, G. Mykytyn, K. Ruda, H. Hulak, Detecting Deepfake Modifications of Biometric Images using Neural Networks, in: Cybersecurity Providing in Information and Telecommunication Systems, 3654, 2024, 391–397.

[34] V. Kharchenko, I. Chyrka, Detection of Airplanes on the Ground using YOLO Neural Network, in: Int. Conf. on Mathematical Methods in Electromagnetic Theory, 2018, 294–297. doi:10.1109/MMET.2018.8460392

[35] O. Zaporozhets, V. Isaienko, K. Synylo, Trends on Current and Forecasted Aircraft Hybrid Electric Architectures and Their Impact on Environment, Energy 211 (2020). doi:10.1016/j.energy.2020.118814

[36] Z. Hu, Y. Khokhlachova, V. Sydorenko, I. Opirskyy, Method for Optimization of Information Security Systems Behavior under Conditions of Influences, Int. J. Intell. Syst. Appl. 9 (12) (2017) 46–58. doi:10.5815/ijisa.2017.12.05