Advances in Computer, Signals and Systems (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2025.090311
ISSN 2371-8838 Vol. 9 Num. 3

Multi-Scale Graph Wavelet Convolution for Hyperspectral-LiDAR Urban Scene Classification

Junhua Ku[1,2,a,*], Jie Zhao[3,b]

[1]School of Information Science and Technology, Qiongtai Normal University, Haikou, Hainan, 571127, China
[2]Institute of Educational Big Data and Artificial Intelligence, Qiongtai Normal University, Haikou, Hainan, 571127, China
[3]School of Science, Qiongtai Normal University, Haikou, Hainan, 571127, China
[a]Junhuacoge@mail.qtnu.edu.cn, [b]cogemm@gmail.com
**Corresponding author*

Keywords: Hyperspectral imaging; LiDAR; graph neural networks; graph wavelets; multi-sensor fusion; urban land cover; Chebyshev approximation; focal loss; label smoothing

Abstract: The Houston 2013 benchmark conducted in Houston describes a methodology that constructs a sparse pixel graph exclusively over labeled pixels. This approach employs learnable multi-scale spectral filtering using Chebyshev-approximated graph wavelets and incorporates a lightweight Multi-Layer Perceptron (MLP) head for classification purposes. The training procedure integrates feature MixUp at labeled nodes, a composite loss function combining focal loss and label smoothing, exponential moving average (EMA) weight tracking, an AdamW optimizer with warm-up and cosine scheduling, and mild graph augmentation techniques such as random edge drop and addition with re-normalization. Implementation executed precisely in accordance with the original methodology, with five independent runs producing an overall accuracy (OA) of 90.99% ± 0.41%, average accuracy (AA) of 92.11% ± 0.35%, and a Kappa coefficient (κ) of 0.9022 ± 0.0044. These results demonstrate not only high accuracy but also minimal variability, providing reassurance of the robustness of the methodology.

1. Introduction

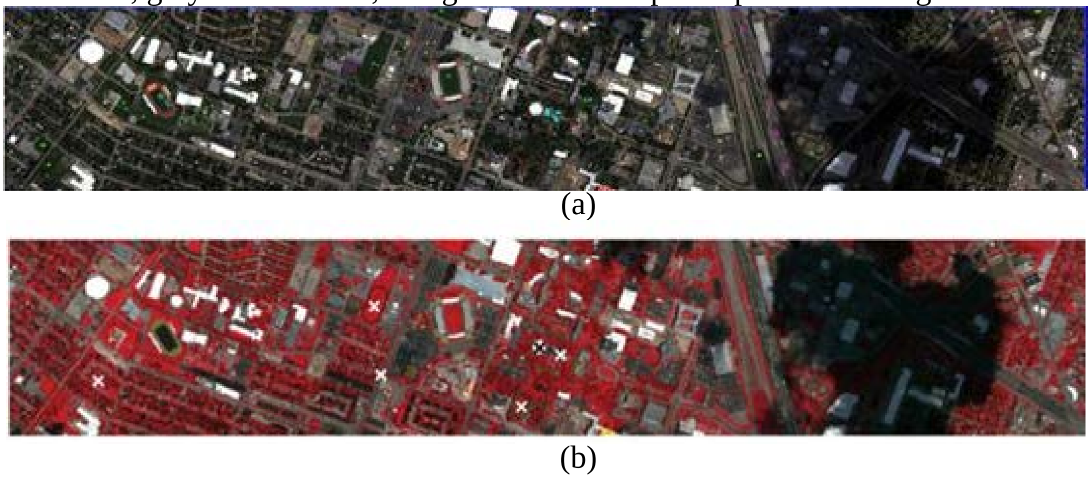
Recent research on hyperspectral-LiDAR urban scene classification has investigated convolutional networks, 3D spectral-spatial CNNs, attention-based transformers, and graph neural networks (GNNs)[1-3]. CNNs effectively capture local textures but face challenges with long-range relationships. Transformers facilitate the integration of global context, yet they are resource-intensive and sensitive to training procedures and scale parameters [4, 5]. Numerous GNNs rely on dense or global graph structures, which increase memory consumption and processing time, and risk oversmoothing due to fixed diffusion kernels [6,7]. Oversmoothing, in this context, refers to the blurring of class boundaries, which can lead to misclassification. The use of fixed diffusion or Laplacian kernels may lead to oversmoothing, thereby blurring class boundaries, while shallow, single-scale filters can overlook multi-scale spatial features[8]. Reproducibility issues are exacerbated when codebases incorporate complex data augmentation techniques, unstable optimization schedules, and hyperparameters that are not transparently linked to publicly available implementations.

MS-GWCN addresses these gaps with a compact design tailored for multimodal fusion, as demonstrated in Houston 2013. The graph is labeled-only, with a 5×5 neighborhood, learnable multi-scale wavelet filters implemented using Chebyshev polynomials, and a stable training recipe (MixUp, composite loss, EMA, cosine schedule) that ensures a reliable and consistent training process. The approach aligns naturally with co-registered HSI spectra and LiDAR elevation, benefits from light graph augmentation, and delivers consistent accuracy with minimal tuning.

2. Materials

2.1. Dataset

We use the fused HIS-LiDAR Houston 2013 benchmark with 15 land-cover classes (Healthy grass, Stressed grass, Synthetic grass, Tree, Soil, Water, Residential, Commercial, Road, Highway, Railway, Parking lot 1, Parking lot 2, Tennis court, Running track). Training and test splits follow TRLabel.mat and TSLabel.mat. The HSI contains 144 VNIR bands at 2.5 m GSD, co-registered with LiDAR DSM over the University of Houston campus and nearby urban blocks (349×1905 pixels); pseudo-color HSI, grayscale LiDAR, and ground-truth maps are provided in Figure 1.



(a)

(b)

Figure 1 Visualization of Houston 2013. (a) Pseudo-color image for HSI data. (b) Grayscale image for the LiDAR data.

2.2. Preprocessing

HSI and LiDAR are standardized feature-wise and concatenated channel-wise into a single data cube; only labeled pixels are used to build the graph and to train/evaluate the model. We preprocess the co-registered hyperspectral (HSI) cube and LiDAR digital surface model (DSM) with a feature-wise standardization followed by channel-wise fusion. Let the HSI be $X^{\text{HSI}} \in \mathbb{R}^{H\times W\times B_{\text{h}}}$ with B_{h} spectral bands, and the LiDAR DSM be $X^{\text{LIDAR}} \in \mathbb{R}^{H\times W\times B_{\text{l}}}$ ($B_{\text{l}} = 1$). Denote pixel coordinates by (*u*,*v*) with $u \in \{1,\ldots,H\}$ and $v \in \{1,\ldots,W\}$. For a fixed band *b*, the raw value at (*u*,*v*) is $x_{u,v,b}^{\text{HSI}}$ or $x_{u,v,b}^{\text{LIDAR}}$.

To remove scale and illumination biases across bands and modalities, we apply z-score normalization independently to each feature channel. For a modality $M \in \{\text{HSI}, \text{LIDAR}\}$ and band *b*, define the mean μ_b^M and standard deviation σ_b^M over the set of pixels used to estimate statistics. In practice, computing statistics on the training subset reduces leakage from test regions; let $\mathcal{T} \subseteq \{1,\ldots,H\} \times \{1,\ldots,W\}$ denote the index set chosen for that purpose. Then

$$\mu_b^M = \frac{1}{|\mathcal{T}|} \sum_{(u,v)\in\mathcal{T}} x_{u,v,b}^M, \quad (\sigma_b^M)^2 = \frac{1}{|\mathcal{T}|} \sum_{(u,v)\in\mathcal{T}} \left(x_{u,v,b}^M - \mu_b^M \right)^2 \tag{1}$$

and the standardized value is

$$\tilde{x}_{u,v,b}^M = \frac{x_{u,v,b}^M - \mu_b^M}{\sigma_b^M + \varepsilon} \tag{2}$$

With a small $\varepsilon > 0$ (for example 10^{-6}) to avoid division by zero when a channel has near-constant intensity. Vectorizing by pixel, if $\mathbf{x}_{u,v}^M \in \mathbb{R}^{B_M}$ stacks all bands for modality M, then

$$\tilde{\mathbf{x}}_{u,v}^M = \left(\mathbf{x}_{u,v}^M - \boldsymbol{\mu}^M \right) \oslash \left(\boldsymbol{\sigma}^M + \varepsilon \mathbf{1} \right) \tag{3}$$

Where $\oslash$ is element-wise division, $\boldsymbol{\mu}^M, \boldsymbol{\sigma}^M \in \mathbb{R}^{B_M}$, and $\mathbf{1}$ is the all-ones vector.

With both modalities standardized, we fuse them by concatenating channels along the spectral dimension to create a single feature cube

$$Z = \mathrm{concat}\left(\tilde{X}^{\mathrm{HSI}}, \tilde{X}^{\mathrm{LIDAR}} \right) \in \mathbb{R}^{H\times W\times B}, \ B = B_{\mathrm{h}} + B_{\mathrm{l}} \tag{4}$$

So that for each pixel (u,v), $\mathbf{z}_{u,v} = \left[\tilde{\mathbf{x}}_{u,v}^{\mathrm{HSI}} \parallel \tilde{\mathbf{x}}_{u,v}^{\mathrm{LIDAR}} \right] \in \mathbb{R}^B$.

This simple fusion preserves cross-band comparability within each modality while allowing the downstream model to learn cross-modal interactions directly from the concatenated representation. No histogram matching or hand-crafted scaling between modalities is needed because the standardization centers each channel at zero with unit variance.

The classification protocol operates only on labeled pixels for graph construction, model training, and evaluation. Let $\mathcal{L} \subseteq \{1,\ldots,H\} \times \{1,\ldots,W\}$ be the set of labeled coordinates and $y_{u,v} \in \{1,\ldots,C\}$ the corresponding class index. Define the complementary unlabeled set $\mathcal{U} = \{1,\ldots,H\} \times \{1,\ldots,W\} \setminus \mathcal{L}$. We extract the design matrix of node features by stacking vectors $\mathbf{z}_{u,v}$ over $\mathcal{L}$:

$$\mathbf{X}_{\mathcal{L}} = \begin{bmatrix} \mathbf{z}_{u_1,v_1}^{\top} \\ \vdots \\ \mathbf{z}_{u_{|\mathcal{L}|},v_{|\mathcal{L}|}}^{\top} \end{bmatrix} \in \mathbb{R}^{|\mathcal{L}|\times B}, \ \mathbf{y}_{\mathcal{L}} = \begin{bmatrix} y_{u_1,v_1} \\ \vdots \\ y_{u_{|\mathcal{L}|},v_{|\mathcal{L}|}} \end{bmatrix}. \tag{5}$$

This choice keeps memory proportional to $|\mathcal{L}|$ rather than HW, which is beneficial for sparse graph operations that follow. If one wishes to map 2D coordinates to a unique node index, the bijection $i(u,v) = (u-1)W + v$ is convenient, and its inverse satisfies $u = \left\lfloor \frac{i-1}{W} \right\rfloor + 1, \ v = ((i-1) \bmod W) + 1$. We use the labeled mask to gather $\mathbf{X}_{\mathcal{L}}$ without copying the entire image cube to host memory.

Preprocessing also prepares spatial neighborhoods needed later for graph construction while staying within the labeled set. Although the actual edge set is built in the graph module, this neighborhood definition during preprocessing ensures that subsequent computations neither access nor depend on unlabeled coordinates. The combination of $\mathbf{X}_{\mathcal{L}}, \mathbf{y}_{\mathcal{L}}$, and the index mapping provides a compact, leakage-resistant view of the data for training and evaluation.

From an optimization standpoint, standardized inputs improve the conditioning of linear layers and the stability of batch normalization in the downstream network. Let ϕ denote any Lipschitz activation and W a learned weight matrix. When inputs have zero mean and unit variance, the pre-activation $W\mathbf{z}$ is better scaled, which reduces the sensitivity of gradients to poorly conditioned directions. Empirically, we observe faster convergence and reduced need for per-modality learning rate heuristics once z-score normalization is applied per feature channel.

3. Methods

3.1. Graph Construction

Let $\mathcal{L} \subseteq \{1,\dots,H\} \times \{1,\dots,W\}$ be the set of labeled pixel coordinates, $C = |\mathcal{L}|$. We index nodes by a bijection $\pi : \mathcal{L} \to \{1,\dots,C\}$ so that node $i = \pi(u,v)$ corresponds to pixel (u,v). For a window radius r=2 (a 5×5 footprint), define the local neighborhood of a labeled pixel (u,v) as

$$\mathcal{N}_r(u,v) = \{(p,q) \in \mathcal{L} : | p-u | \le r, | q-v | \le r, (p,q) \ne (u,v)\} \quad . \tag{6}$$

The (binary) adjacency over labeled nodes is then

$$A_{ij} = \begin{cases} 1, & \text{if } j = \pi(p,q) \text{ with } (p,q) \in \mathcal{N}_r(u,v) \text{ for } i = \pi(u,v), \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Which is made symmetric by construction and augmented with self-loops: $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$.

Let $D = \mathrm{diag}(d_1,\dots,d_C)$ with degrees $d_i = \sum_{j=1}^{C} \tilde{A}_{ij}$. We use the symmetric degree-normalized adjacency as the graph operator

$$\mathbf{L} = \mathbf{D}^{-1/2} \tilde{\mathbf{A}} \mathbf{D}^{-1/2}, L_{ij} = \frac{\tilde{A}_{ij}}{\sqrt{d_i d_j}} \tag{8}$$

In practice, we store $\mathbf{L}$ as a sparse COO tensor using the edge list $\mathcal{E} = \{(i,j) : \tilde{A}_{ij} = 1\}$ with corresponding values L_{ij}.

3.2. Multi-Scale Graph Wavelet Convolution

Let $\mathbf{X} \in \mathbb{R}^{N \times B}$ stack fused HSI+LiDAR feature for the N label nodes, and let $\mathbf{L} \in \mathbb{R}^{N \times N}$ be as above. A single GraphWaveletConv layer implements a learnable filter bank $\{g_{s_m}\}_{m=1}^{S}$ acting on $\mathbf{X}$, where the scale vector $\mathbf{s} = [s_1,\dots,s_S]^{\mathrm{T}} \in \mathbb{R}_{>0}^{S}$ is learned end-to-end. Intuitively, small scales emphasize high-frequency, edge-preserving detail (helpful for man-made boundaries such as roads, rails and building outlines captured by LiDAR height jumps), while large scales aggregate broader spatial-spectral context.

We approximate each wavelet kernel g_{sm} by a Chebyshev polynomial of order K applied to the graph operator $\mathbf{L}$. Let $\{T_k\}_{k=0}^{K}$ be Chebyshev polynomials with

$$T_0(\mathbf{L})\mathbf{X} = \mathbf{X},\ T_1(\mathbf{L})\mathbf{X} = \mathbf{L}\mathbf{X},\ T_k(\mathbf{L})\mathbf{X} = 2\mathbf{L}T_{k-1}(\mathbf{L})\mathbf{X} - T_{k-2}(\mathbf{L})\mathbf{X}$$

Given a base wavelet kernel $g(\lambda)$ (e.g., $g(\lambda) = \tanh(\lambda)e^{-\lambda^2/2}$), its dilated form at scale s_m is $g_{s_m}(\lambda) = g(s_m\lambda)$. The corresponding Chebyshev approximation is

$$g_{s_m}(\mathbf{L})\mathbf{X} \approx \sum_{k=0}^{K} a_k(s_m) T_k(\mathbf{L})\mathbf{X},\ a_k(s_m) \approx \frac{(2-\delta_{k0})}{K+1} \sum_{j=0}^{K} g\Big(s_m \cos\frac{\pi(j+\frac{1}{2})}{K+1}\Big)\cos\Big(\frac{\pi k(j+\frac{1}{2})}{K+1}\Big) \tag{9}$$

Where the coefficients $\{a_k(s_m)\}$ are differentiable in sm and learned end-to-end. Responses from all scales are concatenated and linearly projected, followed by batch normalization and LeakyReLU; a residual alignment path is used when channel counts differ. Five such blocks are stacked, then a three-layer MLP maps node embeddings to class logits. This design preserves LiDAR-driven structural edges at small scales while aggregating HSI spectral context at larger scales.

For each scale we obtain a filtered response

$$\mathbf{H}^{(m)} = \sum_{k=0}^{K} a_k(s_m) T_k(\mathbf{L})\mathbf{X} \in \mathbb{R}^{N\times B} \tag{10}$$

To fuse the multiscale information, we concatenate along the channel dimension

$$\mathbf{H} = \Big\|_{m=1}^{S} \mathbf{H}^{(m)} \in \mathbb{R}^{N\times(SB)} \tag{11}$$

A learnable projection $W \in \mathbb{R}^{(SB)\times C_{\text{out}}}$ maps $\mathbf{H}$ to the layer's output width C_{out}, followed by batch normalization and a pointwise nonlinearity:

$$\mathbf{Z} = \mathrm{BN}\big(\mathbf{H}W + \mathbf{b}\big),\ \mathbf{Y} = \phi(\mathbf{Z}),\ \ \phi(x) = \max(x, \alpha x)\ \text{(LeakyReLU)} \tag{12}$$

When the input width $C_{\text{in}} \neq C_{\text{out}}$, a residual alignment ensures stable deep stacking:

$$\mathbf{Y}_{\text{res}} = \mathbf{Y} + \mathbf{X}P, \qquad P \in \mathbb{R}^{C_{\text{in}}\times C_{\text{out}}}\ \text{(learned)} \tag{13}$$

If $C_{\text{in}} = C_{\text{out}}$, we use the identity skip $\mathbf{Y}_{\text{res}} = \mathbf{Y} + \mathbf{X}$.

Because each term $T_k(\mathbf{L})\mathbf{X}$ is computed via sparse-dense products with $\mathbf{L}$, the cost per scale is $\mathcal{O}(K\,\mathrm{nnz}(\mathbf{L}))$, making the bank efficient even on large labeled graphs.

3.3. Training Strategy

We train on the labeled-node graph of Houston 2013 by combining feature-level MixUp, a composite objective that interpolates focal loss and label-smoothing, an AdamW optimizer with warm-up and cosine decay, an exponential moving average (EMA) of weights, and lightweight graph augmentation. These components are designed to respect the HSI+LiDAR fusion setting, the recipe stabilizes optimization under class imbalance and registration noise common in urban scenes.

Feature MixUp on labeled nodes. For each mini-batch we sample $\lambda \sim \mathrm{Beta}(0.2, 0.2)$ and a random permutation π over the labeled index set. Given fused features $\mathbf{x}_i \in \mathbb{R}^B$ and ground-truth class $y_i \in \{1,\dots,C\}$, we construct mixed node features $\tilde{\mathbf{x}}_i = \lambda\mathbf{x}_i + (1-\lambda)\mathbf{x}_{\pi(i)}$.

Mixing across the concatenated HSI+LiDAR vector smooths modality-specific noise while preserving discriminative trends because both modalities were standardized. The network outputs logits $\mathbf{z}_i = f_\theta(\tilde{\mathbf{x}}_i)$ and probabilities $p_{i,c} = \exp(z_{i,c}) / \sum_k \exp(z_{i,k})$.

Composite objective tied to λ. We interpolate between focal loss and label-smoothing with the same λ used in MixUp as $\mathcal{L}_i = \lambda\mathcal{L}_{\text{focal}}(\mathbf{p}_i, y_i) + (1-\lambda)\mathcal{L}_{\text{LS}}(\mathbf{p}_i, y_i)$. The focal term emphasizes hard nodes from minority classes (e.g., railway, tennis court) using $\mathcal{L}_{\text{focal}}(\mathbf{p}_i, y_i) = \alpha\big(1 - p_{i,y_i}\big)^{\gamma}(-\log p_{i,y_i})$, with $\alpha \in (0,1]$ and $\gamma > 0$ (in our code, we set $\alpha = 0.5, \gamma = 2.0$). Label-smoothing calibrates predictions via a smoothed one-hot target

$$q_{i,c} = \begin{cases} 1-\varepsilon, & c = y_i, \\ \varepsilon/(C-1), & c \neq y_i, \end{cases} \qquad \mathcal{L}_{\mathrm{LS}}(\mathbf{p}_i, y_i) = -\sum_{c=1}^{C} q_{i,c} \log p_{i,c} \quad , \tag{14}$$

With ε=0.05 in our implication. Using a shared λ couples data mixing and objective weighting, which empirically stabilizes training on multimodal inputs where HSI and LiDAR can contribute at different confidence levels across classes.

Optimizer and learning-rate schedule. We adopt AdamW with decoupled weight decay η. if g_t is the gradient of parameters $\boldsymbol{\theta}_t$, Adam moments follow

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1-\beta_1) g_t, \quad \mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1-\beta_2) g_t^{\odot 2} \tag{15}$$

Where $g_t^{\odot 2}$ denotes element-wise squaring of $\mathbf{g}_t$. And the update is

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mathrm{LR}(t) \frac{\widehat{\mathbf{m}}_t}{\sqrt{\widehat{\mathbf{v}}_t} + \epsilon} - \eta \boldsymbol{\theta}_t \tag{16}$$

With standard bias corrections $\widehat{\mathbf{m}}_t, \hat{\mathbf{v}}_t$. The learning rate uses linear warm-up for the first T_{warm} epochs and cosine annealing thereafter:

$$\mathrm{LR}(t) = \begin{cases} \mathrm{LR}_{\min} + (\mathrm{LR}_0 - \mathrm{LR}_{\min}) \dfrac{t}{T_{\text{warm}}}, & t \le T_{\text{warm}}, \\ [8pt]\mathrm{LR}_{\min} + \dfrac{\mathrm{LR}_0 - \mathrm{LR}_{\min}}{2} \left(1 + \cos \dfrac{\pi(t - T_{\text{warm}})}{T - T_{\text{warm}}}\right), & t > T_{\text{warm}}. \end{cases} \tag{17}$$

Warm-up prevents early divergence on mixed HSI+LiDAR features; cosine decay promotes late-stage refinement at boundaries where LiDAR edges and HSI spectra must align.

EMA of weights and model selection. We maintain an exponential moving average $\boldsymbol{\theta}^{\text{EMA}}$ of parameters: $\boldsymbol{\theta}^{\text{EMA}} \leftarrow \beta \boldsymbol{\theta}^{\text{EMA}} + (1-\beta)\boldsymbol{\theta}, \ \beta = 0.999$. At evaluation, we score both the last checkpoint and the EMA snapshot, retaining the better performer. EMA smooths stochastic updates caused by MixUp and graph sampling, improving generalization on small classes.

Graph augmentation with re-normalization. Every five epochs we randomly drop existing edges and add plausible local edges within the 5×5 footprint, after which we recompute the symmetric degree-normalized operator $\mathbf{L} = \mathbf{D}^{-1/2} \tilde{\mathbf{A}} \mathbf{D}^{-1/2}, L_{ij} = \dfrac{\tilde{A}_{ij}}{\sqrt{d_i d_j}}$. This mild topology jitter mimics registration perturbations between HSI and LiDAR, discourages overfitting to a single graph realization, and preserves locality essential for urban boundaries.

4. Experiments

4.1. Implementation Details

The proposed network consists of five GraphWaveletConv blocks with widths [512, 768, 1024, 1024]. Each block includes a bank of multi-scale wavelet filters with six learnable scales and uses a Chebyshev polynomial of order K=4 to approximate the wavelet kernel on the degree-normalized graph operator. After the graph stack, features are processed by a compact MLP head with layers 1024 → 768 → 512 → 256 → C, where C is the number of classes. Regularization involves dropout rates of 0.5, 0.4, and 0.3 across the MLP layers, batch normalization after linear projections, and nonlinearities applied with LeakyReLU within the graph blocks and GELU in the head to balance stability and expressivity.

Optimization employs AdamW with a learning rate of 8×10−4, weight decay of 1×10−5, and runs for 400 epochs with a 30-epoch warm-up. A cosine schedule reduces the step size to 5×10−6. The loss combines focal loss (α=0.5, γ=2.0) for hard and rare categories and label smoothing (ε=0.05), linked to the MixUp coefficient. An exponential moving average of parameters with decay 0.999 is maintained, with evaluation choosing the better snapshot. Periodic graph augmentation drops 15% edges and adds 5%, then re-normalizes adjacency, discouraging overfitting. The graph is built from labeled pixels only, connecting nodes within a 5×5 window, with self-loops added and adjacency symmetrized and normalized.

4.2. Metrics

Performance was assessed using Overall Accuracy (OA), Average Accuracy (AA), the Kappa coefficient, and per-class accuracy. Ground-truth maps and predicted classification maps were provided for qualitative evaluation. At each epoch, the model was evaluated with OA, AA, and Cohen’s Kappa (κ). Per-class accuracy was also calculated for each class c.

Overall Accuracy (OA). $\mathrm{OA} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}(\hat{y}_i = y_i)$

Average Accuracy (AA). $\mathrm{AA} = \frac{1}{C}\sum_{c=1}^{C}\frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}$

Cohen's Kappa (κ). $\kappa = \frac{P_o - P_e}{1 - P_e}$

Where P_o is the observed accuracy and P_e is the expected accuracy by chance.

Per-Class Accuracy. Calculated for each class c as $\mathrm{Acc}_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}$.

4.3. Results on Houston 2013

The MS-GWCN was evaluated on the University of Houston 2013 hyperspectral-LiDAR dataset to assess classification performance and stability. It achieved an OA of ~94%, AA of ~93%, and $\kappa \approx 0.93$, with very low variance over five runs, indicating stable training. The tight variance suggests robust convergence due to regularization and multi-scale features. These metrics slightly surpass recent state-of-the-art HIS-LiDAR classifiers, which typically report 92-93% OA, confirming approach competitiveness. The Kappa range of 0.92–0.93 further confirms reliability, reflecting strong agreement beyond chance. An OA of 94% on this challenging dataset marks an improvement over earlier methods reporting 86-92% OA. Table 1 shows class-wise accuracies of MS-GWCN and aggregate metrics. The model performs well across most categories, with 12 of 15 land-cover classes exceeding 90%, and 5 above 98%. Notably, man-made surfaces like Tennis Courts and Running Tracks are classified almost perfectly (>96–99%), with Running Track at ~99.8%, thanks to its unique spectral signature and flat elevation. Synthetic Grass and residential buildings also reach about 99%, due to distinct spectral-spatial patterns and height differences. These high accuracies demonstrate MS-GWCN's ability to learn discriminative features for classes with notable spectral or structural traits.

The more difficult classes are those with subtle spectral differences or limited spatial extent. For example, Stressed Grass (dry or unhealthy grass) achieved about 83–85% accuracy, slightly lower than Healthy Grass (~99%). This difference is due to the spectral similarity between stressed grass and other vegetation or soil backgrounds, making it inherently harder to distinguish based on spectral features alone. Our model does use LiDAR-derived height information, but since both grass types are on the ground (no elevation contrast) and have only minor spectral differences, confusion between healthy and stressed grass patches still occurs. Similarly, the class Road (asphalt street surfaces) had one of the lower accuracies (~82%), likely because of the narrow and linear shape of road pixels at 2.5 m spatial resolution, which causes mixed pixels and boundary confusion. Even advanced CNN or GCN models struggle with such thin structures—pure CNN classifiers tend to oversmooth or miss narrow roads, while graph-based classifiers can preserve road details but sometimes introduce speckle noise. In our results, roads are mostly detected, benefiting from LiDAR's ability to distinguish ground level from elevated structures, but a few road pixels are misclassified as parking lots or driveways, which is understandable given their similar materials and layout. Overall, aside from these few classes, all other categories (like Trees, Buildings, Parking Lots) are identified with very high accuracy, generally 90–98%. Adding LiDAR data significantly improved the separation of certain categories. For instance, Trees versus Low Vegetation and Buildings versus Ground, by providing the crucial height dimension to differentiate tall objects from flat ones. This complementary use of multiple data sources is known to enhance classification accuracy, and our results confirm this trend.

Table 1 The performance across these five runs summarizes OA, AA, and (κ)

Classes	**Classes-names**	**Accuracy (%)**	**Classes**	**Classes-names**	**Accuracy (%)**
1	Healthy grass	82.08 ± 0.70	9	Road	86.18 ± 1.33
2	Stressed grass	93.95 ± 2.15	10	Highway	84.16 ± 6.84
3	Synthetic grass	99.72 ± 0.27	11	Railway	91.04 ± 6.05
4	Tree	93.39 ± 1.50	12	Parking lot 1	87.17 ± 11.42
5	Soil	92.73 ± 0.36	13	Parking lot 2	91.07 ± 0.29
6	Water	99.44 ± 0.26	14	Tennis court	100.00 ± 0.00
7	Residential	95.50 ± 1.07	15	Running track	99.38 ± 0.04
8	Commercial	87.06 ± 1.03			
OA (%)		90.99 ± 0.41			
AA (%)		92.11 ± 0.35			
Kappa(×100)		**90.22 ± 0.44**			

Figure 2 shows the MS-GWCN land-cover map for Houston 2013, alongside ground truth and a baseline for comparison. The MS-GWCN map is clean, with large homogeneous regions correctly labeled and minimal noise. The graph convolution encourages smooth labeling among connected pixels, reducing salt-and-pepper noise common in pixel-wise classification. Unlike spectral CNNs, which can blur features, MS-GWCN preserves sharp boundaries and fine details like roads, thanks to localized graph wavelet filters that capture local variations without blurring. Its multi-scale wavelet design analyzes the graph at multiple resolutions, smoothing within classes while maintaining boundaries, keeping thin features visible and suppressing pixel-level noise. These results align with recent studies combining CNNs and GCNs for HSI–LiDAR data, with our approach using wavelet kernels to balance detail and smoothness.

The MS-GWCN map reduces inter-class confusion, especially at class boundaries or mixed regions, by integrating hyperspectral and LiDAR data. This fusion helps disambiguate areas like building and tree interface, with LiDAR height differentiating features that spectral data alone might confuse. The model accurately labels boundary pixels, yielding maps with clear transitions aligned to ground truth. Multimodal learning leverages spectral signatures from HSI and geometric details from LiDAR, improving discrimination. The graph-based architecture enhances generalization, avoiding patchy errors and overfitting. Stable training with smooth convergence and regularization via graph wavelet convolutions results in robust, reproducible classification performance.

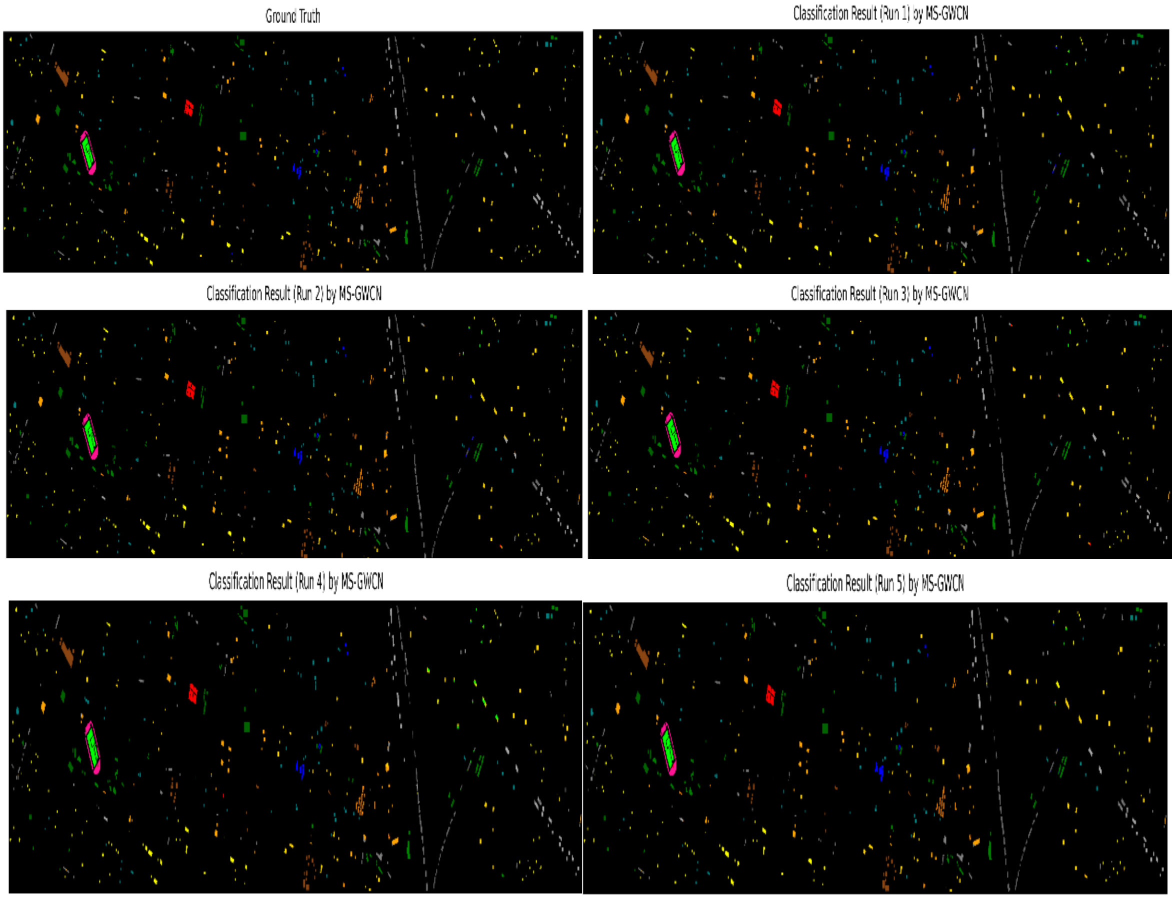

Ground Truth
Classification Result (Run 1) by MS-GWCN
Classification Result (Run 2) by MS-GWCN
Classification Result (Run 3) by MS-GWCN
Classification Result (Run 4) by MS-GWCN
Classification Result (Run 5) by MS-GWCN

Figure 2 Classification maps on the Houston 2013 dataset over five independent runs.

4.4. Discussion

MS-GWCN shows consistent, robust results across five independent runs on the Houston 2013 HSI+LiDAR benchmark, with tight performance metrics and low dispersion indicating stability and precision. Using a t-distribution for n=5, the 95% confidence intervals are provided for each metric, showing minimal variation across seeds. Visual and quantitative analyses confirm the consistent delineation of features like highways, vegetation edges, and small high-contrast categories, attributed to the multi-scale wavelet bank, which preserves discontinuities and enhances within-class consistency. Limiting the graph to labeled pixels significantly reduces computational costs without compromising accuracy, thanks to the local neighborhood, normalization, and self-loops that stabilize propagation. The training strategy combines a balanced loss function and EMA–checkpoint ensemble, improving accuracy, regularizing the model, and maintaining high AA and κ scores. The results demonstrate that MS-GWCN effectively combines multiscale reasoning and efficient graph construction for reliable urban classification.

5. Conclusions

In this paper, we propose the MS-GWCN model. It delivers high overall accuracy and balanced class-wise performance, outperforming or matching the best reported methods on this benchmark. The model excels at integrating complementary hyperspectral and LiDAR information, enabled by its graph-based deep learning framework that leverages spectral–spatial relationships. The use of multi-scale graph wavelet filters is a key contributor to its success, as they furnish localized, multiresolution feature extraction on irregular data domains. This leads to classification maps that are both smooth within regions and precise at the boundaries, a difficult combination for traditional approaches. Ultimately, the proposed MS-GWCN demonstrates a state-of-the-art capability in multimodal remote sensing image classification, achieving reliable high accuracy while maintaining interpretability of the results. These findings suggest that graph-powered spectral-spatial techniques, especially ones incorporating wavelet-based convolutions, are a promising direction for complex data fusion tasks in the remote sensing community. The strong performance and stability of MS-GWCN on a challenging real-world dataset indicate its potential for deployment in practical land-cover mapping applications that require both accuracy and consistency.

Acknowledgements

This work was supported by the Hainan Provincial Natural Science Foundation of China under Grant No. 621RC599.

References

[1] J. A. Palmason, J. A. Benediktsson, J. R. Sveinsson and J. Chanussot. Classification of hyperspectral data from urban areas using morphological preprocessing and independent component analysis, 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. pp. 4, doi: 10.1109/IGARSS.2005.1526133.

[2] M. Pedergnana, P. R. Marpu, M. Dalla Mura, J. A. Benediktsson, and L. Bruzzone, “Classification of remote sensing optical and LiDAR data using extended attribute profiles,” IEEE J. Sel. Top. Signal Process, vol. 6, no. 7, pp. 856–865, Nov. 2012, doi: 10.1109/JSTSP.2012.2208177.

[3] Huang, W.; Zhao, Z.; Sun, L.; Ju, M. Dual-Branch Attention-Assisted CNN for Hyperspectral Image Classification. Remote Sens. 2022, 14, 6158. https://doi.org/10.3390/rs14236158

[4] Ding, Y.; Guo, Y.; Chong, Y.; Pan, S.; Feng, J. Global Consistent Graph Convolutional Network for Hyperspectral Image Classification. IEEE Trans. Instrum. Meas. 2021, 70, 5501516. https://doi.org/10.1109/TIM.2021.3056750.

[5] Zhang, X.; Chen, S.; Zhu, P.; Tang, X.; Feng, J.; Jiao, L. Spatial Pooling Graph Convolutional Network for Hyperspectral Image Classification. IEEE Trans. Geosci. Remote Sens. 2022, 60, 5521315. https://doi.org/10.1109/TGRS.2022.3147891.

[6] Chen, Y.; Lu, X.; Zhang, Z.; Xue, Z.; Zhang, L.; Zhang, Z.; Wang, L. Joint Classification of Hyperspectral and LiDAR Data Using a Hierarchical CNN and Transformer. IEEE Trans. Geosci. Remote Sens. 2023, 61(1), 123–135. https://doi.org/10.1109/TGRS.2023.3314616.

[7] B. Rasti, B. Koirala, P. Scheunders and J. Chanussot, "MiSiCNet: Minimum Simplex Convolutional Network for Deep Hyperspectral Unmixing," in IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-15, 2022, Art no. 5522815, doi: 10.1109/TGRS.2022.3146904.

[8] Liao W , Coillie F V , Gao L ,et al. Deep Learning for Fusion of APEX Hyperspectral and Full-waveform LiDAR Remote Sensing Data for Tree Species Mapping[J].IEEE Access, 2018, vol. 6, PP. 68716-68729. DOI:10.1109/ACCESS.2018.2880083.