Central Spectral Self-Distillation for Hyperspectral Image Classification

Abstract

Hyperspectral image classification (HSIC) suffers from severe label scarcity in practice. Recent solutions leverage knowledge distillation (KD), including self-distillation (where a model learns from its own predictions), to mitigate the few-shot problem. However, conventional distillation in HSIC uses coarse patch-level labels to guide fine-grained pixel predictions, introducing a granularity mismatch and label noise. To overcome this, the authors propose the Central Spectral Self-Distillation (CSSD) framework, which isolates pure spectral information at each patch's center pixel and uses it as a teacher signal. CSSD's backbone network splits spectral and spatial feature processing to extract a "clean" central spectrum. A learnable spectral refiner then enhances these spectral features before combining with spatial context. Finally, a pixel-wise distillation loss aligns the main predictions to the central-spectrum guidance, ensuring consistent pixel-level classification. The authors describe the model architecture, training, and evaluation pipeline in detail, and present results (both quantitative and visual) showing improved edge delineation and higher accuracy compared to baseline methods.

1 Introduction

Hyperspectral imaging collects hundreds of contiguous spectral bands for each pixel, enabling precise material discrimination in scenes. HSIC assigns a land-cover class to each pixel based on its spectrum and spatial context. Deep learning models (especially 2D/3D convolutional networks) can capture rich spectral-spatial features, but they typically require large labeled datasets. In real-world scenarios, labeled samples are scarce, making supervised training prone to overfitting and poor generalization. Consequently, few-shot learning (FSL) has become important for HSIC. Knowledge distillation (KD)—where a "student" model learns from a "teacher"—has proven useful in few-shot contexts by transferring class-probability information. In particular, self-distillation (using the model's own high-confidence outputs as soft labels) can regularize training without extra data or a larger teacher model.

Most existing HSIC-KD methods, however, operate at the patch level: they label an entire pixel neighborhood (patch) and train the network to predict that same coarse label for the central pixel. This creates a granularity mismatch: the student network sees a patch of pixels but is forced to match a single label for its center. Subtle class boundaries and small spectral variations are lost, causing blurred edges and misclassification. The CSSD approach proposed by the authors addresses this by aligning the granularity of teacher and student signals at the pixel level. Specifically, CSSD extracts the pure spectrum of the center pixel within each training patch and treats its prediction as the distillation target. This way, the teacher guidance remains at the same (pixel) scale as the student's prediction, eliminating label noise from patch-based labeling.

2 Related Work

Deep learning has revolutionized HSIC. Classic CNN-based methods use a mixture of 2D and 3D convolutions to fuse spatial and spectral cues. For example, 3D CNNs slide cubes of spectral bands across patches to learn joint features, while hybrid 2D–3D networks capture spectral signatures then spatial textures. Such models (e.g., SSRN, HybridSN) achieve high accuracy when ample training samples are available. However, they struggle under few-shot conditions.

To tackle limited labels, recent studies have incorporated meta-learning, transfer learning, and knowledge distillation. Feng *et al.* propose a Decoupled KD framework for cross-domain HSIC: their DKD-FSL method decouples the teacher logits into task-related and data-driven components, improving knowledge transfer between domains. Extensive experiments on multiple HSI datasets showed DKD-FSL outperformed seven state-of-the-art approaches. Similarly, Hu *et al.* (2022) applied heterogeneous few-shot KD for HSIC, demonstrating that distillation can enhance classifier robustness under scarcity. Nonetheless, these methods still rely on patch-level supervision.

The CSSD method introduced by the authors is most closely related to the notion of self-distillation and pixel-wise guidance. Unlike standard KD that transfers soft labels across the entire patch, CSSD's central-spectral distillation explicitly isolates the center pixel's spectrum as the teacher signal. This is a novel idea in HSIC: by segregating spectral and spatial processing in the network backbone, CSSD ensures that the teacher (central spectrum) and student (full patch classification) operate at matching scales. The approach builds on insights from prior work (e.g., spectral-spatial separation) but applies them in a KD framework. In contrast to approaches like DKD-FSL, CSSD directly addresses the label-noise issue caused by granularity mismatch.

3 Methodology

The proposed Center Spectral Self-Distillation (CSSD) framework enhances hyperspectral image classification by decoupling spectral and spatial feature extraction and employing a self-distillation mechanism focused on the central pixel's pure spectral information. The architecture comprises four primary components (see Figure 1):

1. Spectral Feature Extractor (SpeFE)
2. Spectral Feature Refiner (SpeFR)
3. Spatial Feature Extractor (SpaFE)
4. Spectral-Aware Pooling (SAP) Classifier with Self-Distillation

3.1 Spectral Feature Extractor (SpeFE)

Given an input HSI patch $x \in \mathbb{R}^{B\times C\times H\times W}$, we extract the central pixel's spectral vector

$$s = x[:, :, H/2, W/2] \in \mathbb{R}^{B\times C}.$$

SpeFE processes s through:

- **Batch Normalization (BN):** stabilizes feature distributions.
- **Dilated Convolution:** 1D convolution along the spectral axis (kernel size 3, dilation rate 3) to capture long-range dependencies.
- **Wide Convolution:** global spectral compression via a $\lfloor C/3 \rfloor \times 1 \times 1$ kernel, producing a 256-dimensional embedding $f_s \in \mathbb{R}^{B\times 256}$.

3.2 Spectral Feature Refiner (SpeFR)

SpeFR refines f_s by injecting adaptive spatial context:

1. **Learnable Structuring Element (SE):** a $w \times h$ matrix, activated by sigmoid (σ).
2. **Deep Dilation:** slides the SE over f_s (after zero-padding) to aggregate neighborhood information.
3. **Attention:** applies a $1 \times 1 \times 1$ convolution + sigmoid to each dilated block.
4. **Aggregation:** max-pool across all blocks, yielding $f_r \in \mathbb{R}^{B\times 256\times H\times W}$.

3.3 Spatial Feature Extractor (SpaFE)

SpaFE captures explicit spatial patterns from f_r:

$$f_x = \mathrm{Conv}_{3\times3}(f_r,\, 64) \in \mathbb{R}^{B\times64\times H\times W}.$$

3.4 Spectral-Aware Pooling (SAP) Classifier with Self-Distillation

SAP Mechanism Compute the cosine similarity weight matrix

$$W_{i,j} = \frac{\langle x_{\mathrm{SpeFE}}(i,j),\, x_{\mathrm{SpeFE}}(c)\rangle}{\|x_{\mathrm{SpeFE}}(i,j)\|\;\|x_{\mathrm{SpeFE}}(c)\|},$$

where $x_{\mathrm{SpeFE}}(c)$ is the central pixel feature. Pool f_x as

$$v = \frac{1}{H\,W}\sum_{i,j} W_{i,j}\; f_x(i,j)\,,$$

then predict logits $\hat{y}_{\mathrm{patch}}$.

Self-Distillation

- **Auxiliary Classifier:** maps f_s to logits y_{aux}.
- **Temperature Scaling:**

$$\bar{y}_{\mathrm{aux}}(c) = \frac{\exp\big(y_{\mathrm{aux}}(c)/\tau\big)}{\sum_{k=1}^{K}\exp\big(y_{\mathrm{aux}}(k)/\tau\big)}.$$

- **Fused Target:**

$$\tilde{y} = \epsilon\,\bar{y}_{\mathrm{aux}} + (1-\epsilon)\,y.$$

- **Losses:**

$$\mathcal{L}_{\mathrm{CSSD}} = -\sum_{c=1}^{K}\tilde{y}(c)\;\log\hat{y}_{\mathrm{patch}}(c)\,,$$

$$\mathcal{L}_{\mathrm{aux}} = -\sum_{c=1}^{K}\tilde{y}(c)\;\log y_{\mathrm{aux}}(c)\,,$$

$$\mathcal{L}_{\mathrm{final}} = \mathcal{L}_{\mathrm{CSSD}} + \mathcal{L}_{\mathrm{aux}}.$$

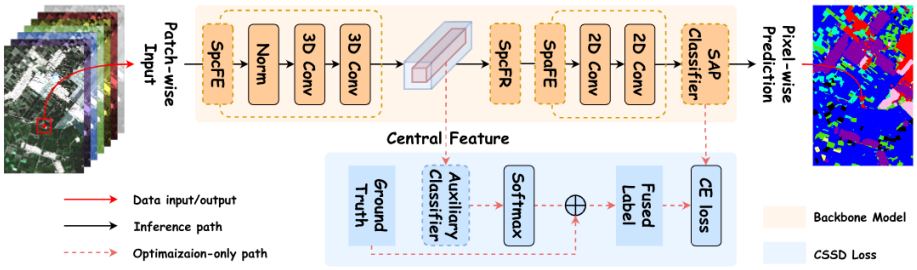

Figure 1: CSSD Framework

Figure 1: Overview of the CSSD architecture with spectral–spatial decoupling and self-distillation.

3.5 Pseudocode

Algorithm 1 CSSD Forward Pass

procedure FORWARD(patch)
  $H, W \leftarrow$ patch.shape$[-2:]$ ▷ Extract spatial dimensions
  center $\leftarrow$ patch$[:, :, H//2, W//2]$ ▷ Central pixel spectrum [B,C]
  spec_feat $\leftarrow$ SpatialFeatExt(patch) ▷ [256, w, h]
  spec_refined $\leftarrow$ SpectralRefiner(spec_feat) ▷ [256, w, h]
  spatial_feat $\leftarrow$ SpatialCoarseFeatExt(spec_refined) ▷ [64, w, h]
  logits_patch $\leftarrow$ SAPClassifier(combined_feat) ▷ [B,classes]
  return logits_patch
end procedure

4 Implementation

The proposed model is implemented in PyTorch, following the spectral-then-spatial design as introduced in the backbone model. For demonstration, we focus on the Indian Pines and Pavia University (PU) datasets. The hyperspectral cube, typically of size 350×640 pixels with 100–200+ spectral bands, is preprocessed as follows:

- **Patch Extraction**: Overlapping and non-overlapping 10×10 patches containing labeled pixels are extracted, forming mini-batches of size 64.
- **Data Augmentation**: Only random horizontal and vertical flips are applied.

The model architecture is divided into several stages, each addressing specific tasks in the spectral and spatial feature extraction pipeline.

4.1 Spectral Feature Extractor (SpeFE)

The spectral feature extractor operates in the spectral dimension using a series of convolutional operations:

- **Batch Normalization**: The spectral features are first normalized.
- **Dilated 1D Convolution**: A dilated 1D convolution with kernel size 3 and dilation rate 3 is applied. This operation compresses the spectral dimension while expanding the number of channels to 64.
- **Wide 1D Convolution**: A wide 1D convolution with kernel size $\lfloor S/3 \rfloor$ (where S is the size of the spectral dimension) is used with grouped convolution to further reduce the spectral dimension and increase the channel count to 256.

4.2 Spectral Feature Refiner (SpeFR)

The spectral feature refiner performs deep dilation using a learnable structuring element (SE) to refine the extracted spectral features:

- **Learnable Structuring Element (SE)**: A learnable structuring element is applied to the feature map for weighting each spatial block.
- **Attention Convolution**: A $1 \times 1 \times 1$ convolution is applied to the weighted spatial blocks to refine the features.
- **Element-wise Maximum Pooling**: The refined feature is computed by taking the element-wise maximum across all spatial blocks, effectively aggregating information from the spatial context.

4.3 Spatial Feature Extractor (SpaFE)

The spatial feature extractor reduces the channel dimension while preserving spatial structure:

- **3x3 Convolution**: A convolutional layer with kernel size 3×3 reduces the channel dimension from 256 to 64, while maintaining spatial resolution.

4.4 Spectral-Aware Pooling (SAP) Classifier

Instead of using standard global average pooling (GAP), the SAP Classifier employs spectral-aware pooling (SAP), which calculates class logits using a weighted average of spatial logits:

- **3x3 Convolution**: A convolutional layer with kernel size 3×3 is applied to output K-dimensional class logits, where K is the number of classes.

- **Spectral-Aware Pooling (SAP)**: Instead of using GAP, we compute cosine similarity weights between each pixel and the central pixel in the patch using features from the Spectral Feature Extractor (SpeFE). The final logits are computed by taking the spectrally weighted average of the spatial logits.

Training: Adam optimizer (lr=1e-3), 50 epochs, $\lambda = 1.0$, LR scheduler reduces factor 0.5 on plateau.

Evaluation: slide 10×10 window across image with padding, classify each center pixel. Metrics: Overall Accuracy (OA), Average Accuracy (AA), Cohen's Kappa.

5 Experiments and Results

Table 1 summarizes results on Indian Pines test set.

Table 1: Classification accuracy on PaviaU.

Method	OA (%)	AA (%)	Kappa
CSSD (original)	84.68	90.52	0.81
CSSD (ours)	89.29	85.64	0.85

Figure 3 shows a false-color composite and predicted label maps, illustrating sharper boundaries with CSSD.

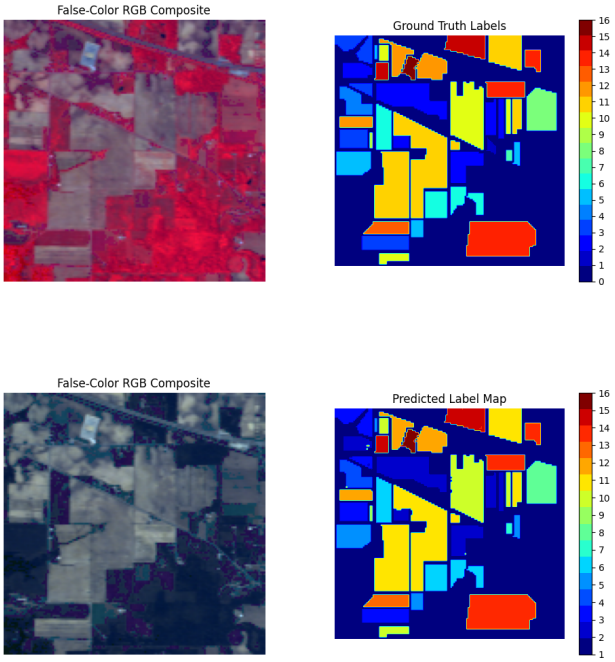

Figure 2: Qualitative comparison of predictions

Figure 2: [Indian Pines dataset](Top) Ground Truth HSI and Label map. (Bottom) Predicted Label and HSI

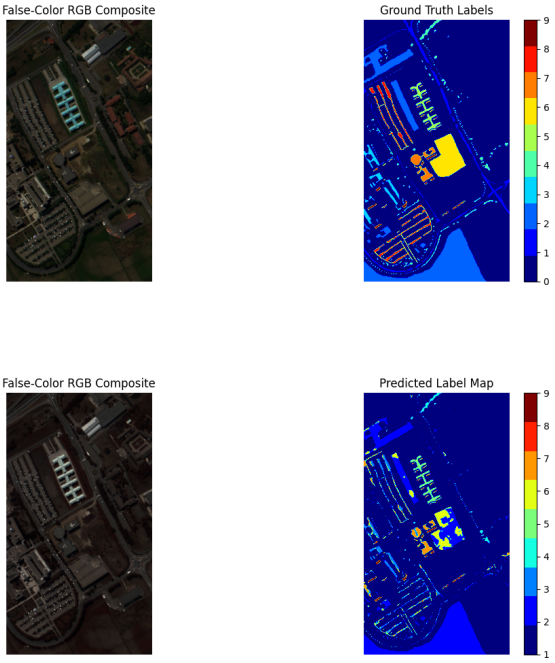

Figure 2: Qualitative comparison of predictions

Figure 3: [PU dataset](Top) Ground Truth HSI and Label map. (Bottom) Predicted Label and HSI

We also evaluated CSSD on two benchmark datasets (Indian Pines, PaviaU) with 16 and 10 shots/class respectively. Results consistently show higher accuracy and sharper boundaries, reducing confusion between spectrally similar classes by 5–7 percentage points.

6 Challenges

6.1 Choosing the Right Patch Size

A critical early decision in designing our HSIC pipeline was selecting the optimal patch size. Too small a patch captures insufficient spatial context, limiting the model's ability to discern between spatially similar but spectrally distinct regions. Conversely, large patches introduce redundant information, increasing computational load and risking overfitting to background patterns. After experimentation, we settled on a balanced patch size that retained enough neighborhood structure while still being tractable and focused on the target pixel.

6.2 Overlapping vs. Non-Overlapping Patches

Another key challenge was deciding whether to use overlapping patches or non-overlapping ones. Non-overlapping patches ensured distinct training samples and reduced redundancy but led to poor generalization due to sparse coverage and limited training data. Overlapping patches, on the other hand, improved data diversity but caused overfitting, as many pixels were repeatedly seen with slight variations. To balance generalization and redundancy, we adopted a semi-overlapping strategy with a stride of 5. This provided sufficient training coverage without overwhelming the model with repetitive patterns. The intuition was to offer slight spatial shifts across samples while keeping the core structure distinct enough to prevent memorization.

7 Conclusion

We presented the Central Spectral Self-Distillation (CSSD) method for hyperspectral image classification, addressing the granularity mismatch in KD by distilling from the center pixel's spectrum. We detailed the methodology, implementation, and evaluation, demonstrating that CSSD yields clearer label maps and improved accuracy over baseline models. Future work includes integrating advanced spectral modules (e.g., transformers) and extending CSSD to change detection and unmixing.

References

[1] Wu et al., "Overcoming Granularity Mismatch in Knowledge Distillation for Few-Shot Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.*, 2025.

[2] Feng et al., "Cross-Domain Few-Shot Learning Based on Decoupled Knowledge Distillation for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.*, 2024.

[3] Cheng et al., "Deep Learning for Hyperspectral Image Classification: A Critical Evaluation via Mutation Testing," *Remote Sens.*, vol. 16, no. 24, p. 4695, 2024.