Progress Report: Implementation of Lightweight Dual-Branch Meta-Learner for Few-Shot Hyperspectral Image Classification Using DMCM2 Framework

Muhammad Miqdad Ramadhan F, Muhammad Khairul Ikhsan
P4DSAI Program - Mega Project
Universitas Syiah Kuala Banda Aceh
mqdad@mhs.usk.ac.id, m.khairuli@mhs.usk.ac.id

***Abstract*—This progress report presents the current status of our implementation of DMCM2 (Dual-adjustment Cross-domain Meta-learner version 2) for few-shot hyperspectral image classification. We have successfully completed the core implementation including data preprocessing, TGAN2 architecture construction, meta-learning training pipeline, and preliminary evaluation. Our lightweight approach achieves parameter efficiency with only 260K parameters while demonstrating competitive performance on the Pavia University dataset under a 9-way 5-shot setting. This report details our accomplishments, preliminary experimental results, encountered challenges, and planned next steps toward the final project completion.**

I. Introduction

A. Project Overview

Hyperspectral image (HSI) classification with limited labeled samples remains a critical challenge in remote sensing applications. Our project focuses on implementing DMCM2, a lightweight meta-learning framework that addresses this challenge through efficient architecture design and advanced metric learning. The primary objective is to achieve competitive classification accuracy while maintaining minimal computational requirements suitable for practical deployment.

B. Motivation

Traditional deep learning approaches for HSI classification require extensive labeled datasets, which are often unavailable or expensive to obtain in real-world scenarios. Few-shot learning offers a promising solution by enabling models to generalize from minimal examples. However, existing meta-learning methods often employ complex architectures with millions of parameters, limiting their applicability in resource-constrained environments. Our implementation aims to bridge this gap by providing a lightweight yet effective solution.

C. Project Objectives

The main objectives of this project are:

- Implement a lightweight 3D Ghost Attention Network (TGAN2) with fewer than 300K parameters
- Integrate covariance-based class-wise metric (CCM) for robust distance measurement
- Apply intracorrection (IC) learning strategy to improve support set quality
- Achieve competitive accuracy on benchmark HSI datasets under few-shot settings
- Provide comprehensive evaluation and comparison with existing methods

II. Related Work

A. Deep Learning for HSI Classification

Deep learning has transformed HSI classification through architectures that exploit spatial-spectral information. 3D-CNN approaches [1] jointly process spatial and spectral dimensions, achieving superior performance compared to traditional methods. However, these models typically require thousands of labeled samples per class, which limits their practical applicability in scenarios with limited ground truth data.

Spectral-Spatial Residual Networks (SSRN) [2] introduced residual connections to improve feature learning and gradient flow in deep networks. While effective, SSRN contains approximately 2.8M parameters, making it computationally expensive for deployment on edge devices or in scenarios requiring rapid model adaptation.

B. Few-Shot Learning Approaches

Few-shot learning aims to recognize new classes from limited examples. Metric-based approaches have shown particular promise in this domain:

Prototypical Networks [3] learn a metric space where classification is performed by computing distances to class prototypes. This approach has been successfully adapted to HSI classification but often struggles with high-dimensional spectral data.

Relation Networks [4] learn a deep distance metric rather than using fixed distance functions. RN-FSC adapted this approach for HSI classification, achieving improved performance over prototypical methods but with increased computational complexity.

C. Meta-Learning for HSI

Recent works have explored meta-learning specifically for HSI classification:

DFSL (Deep Few-Shot Learning) introduced episode-based training for HSI, demonstrating significant improvements over traditional transfer learning approaches.

DCFSL (Deep Cross-Domain Few-Shot Learning) addressed domain shift issues in few-shot HSI classification through adversarial learning.

Gia-CFSL incorporated graph-based inference to model relationships between support and query samples.

CMFSL (Cross-Modal Meta Few-Shot Learning) leveraged cross-modal information to improve feature representations.

These methods demonstrate the effectiveness of meta-learning for HSI classification. However, they generally employ large feature extractors (typically 1-4M parameters), which may be prohibitive for practical applications with computational constraints.

D. Research Gap

While existing meta-learning methods for HSI classification achieve strong performance, there remains a gap in developing lightweight architectures that maintain competitive accuracy while significantly reducing computational requirements. Our implementation of DMCM2 addresses this gap through ghost convolutions, dimension-wise attention mechanisms, and efficient metric learning.

III. Methodology

A. Overall Framework

Our DMCM2 framework consists of three core components working in synergy:

1) **TGAN2 Feature Extractor:** Extracts discriminative spatial-spectral features using ghost convolutions and attention mechanisms
2) **CCM Distance Metric:** Computes class-wise covariance-based distances for robust similarity measurement
3) **IC Learning Strategy:** Refines support set representations through intracorrection during meta-training

The framework operates in an episodic training paradigm where each episode simulates a few-shot learning task with randomly sampled classes and support/query splits.

B. Data Preprocessing Pipeline

We implement a comprehensive preprocessing pipeline consisting of four stages:

1) Spectral Dimensionality Reduction: Principal Component Analysis (PCA) reduces the spectral dimension from 103 bands to 100 while retaining over 99% of the variance:

$$X_{\text{reduced}} = X \cdot W_{\text{PCA}} \tag{1}$$

where $X \in \mathbb{R}^{N\times 103}$ represents the original data and $W_{\text{PCA}} \in \mathbb{R}^{103\times 100}$ contains the principal components.

2) Normalization: Min-max normalization scales features to $[0, 1]$:

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{2}$$

3) Spatial Patch Extraction: We extract 9×9 spatial patches centered at each labeled pixel, providing local context while maintaining computational efficiency. For a pixel at position (i, j), the patch is:

$$P_{i,j} = X[i-4 : i+4, j-4 : j+4, :] \tag{3}$$

4) Few-Shot Data Split: For each class c, we allocate samples as follows:

- Support set: 5 samples (for meta-training and testing)
- Query set: 15 samples (for meta-training)
- Test set: Remaining samples (for final evaluation)

C. TGAN2: Lightweight 3D Ghost Attention Network

1) Ghost Module v2: The Ghost Module v2 achieves parameter efficiency through a two-stage feature generation process:

Stage 1: Intrinsic Feature Generation

$$F_{\text{intrinsic}} = \sigma(\text{BN}(\text{Conv}_{3D}(X; \theta_1))) \tag{4}$$

Stage 2: Ghost Feature Generation

$$F_{\text{ghost}} = \sigma(\text{BN}(\text{DWConv}_{3D}(F_{\text{intrinsic}}; \theta_2))) \tag{5}$$

Feature Concatenation

$$F_{\text{out}} = [F_{\text{intrinsic}}, F_{\text{ghost}}][: C_{\text{out}}] \tag{6}$$

where σ denotes ReLU activation, BN represents batch normalization, and DWConv_{3D} is depthwise 3D convolution.

2) GA2 Block: Ghost Attention Block v2: The GA2 block integrates ghost convolutions with dimension-specific feature extraction:

$$F = \text{GhostModule}(X) \tag{7}$$
$$F_d = \text{DWConv}_{3D}^{(5,1,1)}(F) \tag{8}$$
$$F_h = \text{DWConv}_{3D}^{(1,5,1)}(F) \tag{9}$$
$$F_w = \text{DWConv}_{3D}^{(1,1,5)}(F) \tag{10}$$
$$F_{\text{fused}} = \text{BN}(F_d + F_h + F_w) \tag{11}$$
$$F_{\text{out}} = \text{DFC-Attention}(F_{\text{fused}}) \tag{12}$$

3) DFC Attention Mechanism: The Dimension-wise Feature Channel (DFC) attention module adaptively recalibrates features:

$$A_h = \sigma(\text{FC}_{\text{expand}}(\text{FC}_{\text{reduce}}(\text{AvgPool}_h(F)))) \tag{13}$$
$$A_w = \sigma(\text{FC}_{\text{expand}}(\text{FC}_{\text{reduce}}(\text{AvgPool}_w(F)))) \tag{14}$$
$$A_{\text{total}} = A_h + A_w \tag{15}$$
$$F_{\text{attended}} = F \odot A_{\text{total}} \tag{16}$$

4) Normalization-based Attention Module (NAM): NAM learns channel-wise attention through batch normalization parameters:

$$\alpha_c = \frac{|\gamma_c|}{\sum_{i=1}^{C} |\gamma_i| + \epsilon} \tag{17}$$

$$F_{\text{out}} = \sigma(\text{BN}(F) \cdot \alpha) \odot \text{BN}(F) \tag{18}$$

D. Covariance-based Class-wise Metric (CCM)

Unlike Euclidean distance or cosine similarity, CCM considers intra-class variance through covariance matrices:

1) Class Prototype Computation: For class c with support set $\{f_1^c, f_2^c, \ldots, f_K^c\}$:

$$\mu_c = \frac{1}{K} \sum_{i=1}^{K} f_i^c \tag{19}$$

2) Class Covariance Matrix:

$$\Sigma_c = \frac{1}{K-1} \sum_{i=1}^{K} (f_i^c - \mu_c)(f_i^c - \mu_c)^T + \lambda I \tag{20}$$

where λI is a regularization term (we use $\lambda = 0.01$).

3) Mahalanobis Distance: For a query feature q, the distance to class c is:

$$d(q, c) = (q - \mu_c)^T \Sigma_c^{-1} (q - \mu_c) \tag{21}$$

4) Classification: The predicted class is:

$$\hat{y} = \arg\min_c d(q, c) \tag{22}$$

E. Intracorrection (IC) Learning Strategy

The IC strategy improves support set feature quality by treating support samples as additional queries during training:

1) Query Loss: Standard cross-entropy loss on query samples:

$$\mathcal{L}_{\text{query}} = -\frac{1}{M} \sum_{i=1}^{M} \log P(y_i | x_i; S) \tag{23}$$

2) Intracorrection Loss: Additional loss on support samples:

$$\mathcal{L}_{\text{IC}} = -\frac{1}{NK} \sum_{j=1}^{NK} \log P(y_j | x_j; S) \tag{24}$$

3) Total Training Objective:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{query}} + \lambda_{\text{IC}} \mathcal{L}_{\text{IC}} \tag{25}$$

We set $\lambda_{\text{IC}} = 1.0$ to balance both objectives.

IV. Preliminary Experiments

A. Experimental Setup

1) Dataset Description: Pavia University Dataset:

- Spatial dimensions: 610×340 pixels
- Spectral bands: 103 (after removing noisy bands)
- Spectral range: 0.43-0.86 μm
- Spatial resolution: 1.3 meters
- Number of classes: 9 urban land-cover types
- Total labeled samples: 42,776 pixels

2) Implementation Details:

- Framework: PyTorch 1.12.0
- Hardware: NVIDIA Tesla T4 GPU (16GB)
- CUDA version: 11.6
- Training time: Approximately 18 minutes
- Inference time: 2.5 ms per sample
- Memory usage: <2GB GPU memory

B. Training Results

1) Training Convergence: Our model demonstrates stable convergence over 20 epochs as shown in Table I.

TABLE I
TRAINING PROGRESS (SELECTED EPOCHS)

Epoch	Total Loss	Query Loss	IC Loss	Train Acc
1	2.4521	1.8234	0.6287	0.3156
5	1.2847	0.7923	0.4924	0.6234
10	0.8234	0.4512	0.3722	0.7823
15	0.5621	0.2943	0.2678	0.8645
20	0.4123	0.2156	0.1967	0.9012

C. Architecture Validation

TABLE II
MODEL ARCHITECTURE STATISTICS

Component	Parameters	Percentage
Stage 1 Conv	21,632	8.3%
Stage 1 GA2 Block	84,256	32.4%
Stage 2 Conv	2,048	0.8%
Stage 2 GA2 Block	149,568	57.5%
Batch Norm	2,496	1.0%
Total	**260,000**	**100%**

TABLE III
ARCHITECTURE COMPARISON

Model	Params	FLOPs (M)	Memory (MB)
3D-CNN	1.2M	245	892
ResNet-12	4.26M	892	1,234
SSRN	2.8M	567	1,056
ConvNet-4	950K	178	456
TGAN2 (Ours)	**260K**	**89**	**234**

D. Preliminary Test Results

Based on our preliminary 20-epoch training:

- Overall Accuracy (OA): 75-82% (varies by run)
- Average Accuracy (AA): 73-80%
- Kappa Coefficient: 0.70-0.78

E. Ablation Study

F. Comparison with Baseline Methods

V. Challenges and Solutions

A. Technical Challenges

1) Memory Constraints: **Challenge:** Processing entire HSI at once exceeded GPU memory for classification map generation.

TABLE IV
ABLATION STUDY RESULTS (PRELIMINARY)

Configuration	Train Acc	Params
Baseline (Standard Conv)	0.82	520K
+ Ghost Module	0.85	280K
+ GA2 Block	0.88	260K
+ NAM	0.89	260K
+ CCM	0.90	260K
Full TGAN2 + CCM + IC	0.90	260K

TABLE V
COMPARISON WITH BASELINE METHODS (PAVIA UNIVERSITY)

Method	OA	AA	Kappa
Classical Methods			
RBF-SVM	0.6523	0.7429	0.5636
3DCNN	0.6574	0.7372	0.5737
Deep Learning			
SSRN	0.7696	0.8182	0.7118
Few-Shot Learning			
DFSL	0.7963	0.7641	0.7305
RN-FSC	0.8019	0.7712	0.7373
DCFSL	0.8042	0.8114	0.7471
Gia-CFSL	0.8179	0.8223	0.7629
CMFSL	0.8313	0.8393	0.7811
DMCM	0.8677	0.8485	0.8220
Our Implementation (Preliminary)			
DMCM2 (20 epochs)	0.75-0.82	0.73-0.80	0.70-0.78
Target (Extended Training)			
DMCM2 (Paper)	0.9795	0.9550	0.9715

Solution: Implemented batch processing with batch size of 512 patches, reducing memory footprint from 8GB to <2GB while maintaining efficiency.

2) Training Instability: **Challenge:** Early epochs showed high variance in episode-wise accuracy.

Solution:

- Added covariance regularization ($\lambda = 0.01$)
- Increased episodes per epoch from 100 to 300
- Applied gradient clipping (max norm = 1.0)

B. Implementation Challenges

1) 3D Convolution Compatibility: **Challenge:** Ensuring dimension consistency across 3D convolutions with different kernel sizes.

Solution: Carefully designed padding schemes with symmetric padding for boundary handling.

2) Covariance Matrix Inversion: **Challenge:** Singular or near-singular covariance matrices causing numerical instability.

Solution:

- Added regularization: $\Sigma_c = \Sigma_c + 0.01 \cdot I$
- Used torch.inverse() with exception handling
- Validated positive definiteness before inversion

VI. ERROR ANALYSIS

A. Confusion Pattern Analysis

Preliminary confusion matrix analysis reveals several patterns:

1) High Confusion Pairs:

- Shadows vs. Asphalt: Both dark materials with similar spectral signatures
- Meadows vs. Trees: Vegetation classes with overlapping spectral characteristics
- Bricks vs. Bare soil: Similar reddish materials causing spectral ambiguity

B. Failure Case Analysis

1) Small Object Classes: Classes with small spatial extent (Metal sheets, Shadows) show higher error rates due to fewer training samples and mixed pixels at class boundaries.

2) Spectral Variability: Some classes exhibit high intra-class variance (e.g., Asphalt due to age and wear, Meadows due to vegetation health).

VII. NEXT STEPS

A. Immediate Next Steps (Week 1-2)

1) **Extended Training:** Increase training to 100 epochs with 500 episodes per epoch
2) **Hyperparameter Tuning:** Systematic grid search over key parameters

B. Medium-term Goals (Week 3-4)

1) **Advanced Data Augmentation:** Implement HSI-specific augmentations
2) **Architecture Enhancements:** Test deeper TGAN2 variants
3) **Alternative Metrics:** Compare CCM with other distance measures

C. Final Phase (Week 5-6)

1) **Comprehensive Evaluation:** Multiple runs with statistical analysis
2) **Additional Dataset Evaluation:** Extend to Salinas and Indian Pines datasets
3) **Visualization and Analysis:** Generate t-SNE plots and attention maps
4) **Documentation:** Complete code documentation and repository organization

VIII. TIMELINE AND MILESTONES

TABLE VI
PROJECT TIMELINE

Week	Tasks	Deliverables
1-2	Core implementation	Working code
3	Extended training	Improved results
4	Advanced features	Ablation studies
5	Final experiments	Complete results
6	Report writing	Final report
7	Presentation prep	Slides + demo

IX. Team Contributions

A. Muhammad Miqdad Ramadhan F

Contributions:

- TGAN2 architecture design and implementation
- Ghost Module v2 and GA2 Block coding
- Data preprocessing pipeline
- Training loop implementation
- Preliminary experiments execution

Estimated effort: 50%

B. Muhammad Khairul Ikhsan

Contributions:

- CCM metric implementation
- IC learning strategy integration
- Evaluation metrics computation
- Visualization generation
- Error analysis and report writing

Estimated effort: 50%

X. Conclusion

This progress report demonstrates substantial advancement in implementing DMCM2 for few-shot hyperspectral image classification. We have successfully implemented a lightweight TGAN2 architecture with only 260K parameters, integrated covariance-based metric learning, and achieved preliminary results competitive with classical methods.

A. Key Achievements

Technical Accomplishments:

- 93% parameter reduction vs. ResNet-12
- 75-82% preliminary test accuracy (20 epochs only)
- Stable training convergence
- Efficient inference (<3ms per sample)

B. Expected Final Outcomes

With extended training and optimization, we expect to achieve:

- Overall Accuracy: 90-95%
- Kappa Coefficient: 0.88-0.93
- Competitive performance with state-of-the-art methods

C. Research Impact

This implementation demonstrates that lightweight architectures can achieve competitive performance while enabling edge device deployment through parameter efficiency.

References

[1] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, “Deep feature extraction and classification of hyperspectral images based on convolutional neural networks,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[2] Z. Zhong, J. Li, Z. Luo, and M. Chapman, “Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.

[3] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in *Proc. NIPS*, 2017, pp. 4077–4087.

[4] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, “Learning to compare: Relation network for few-shot learning,” in *Proc. CVPR*, 2018, pp. 1199–1208.

[5] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, “GhostNet: More features from cheap operations,” in *Proc. CVPR*, 2020, pp. 1580–1589.

[6] H. Liu, F. Cao, Y. Yang, D. Tao, and Y. Xu, “Hyperspectral image classification with covariance descriptors on manifold,” *IEEE Trans. Image Process.*, vol. 29, pp. 5858–5871, 2020.

[7] X. Ma, H. Wang, and J. Geng, “Cross-domain few-shot learning for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–18, 2022.

[8] L. Liu, Y. Liang, and Q. Li, “A lightweight dual-branch meta-learner for few-shot hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–16, 2023.