

# Proposal 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**—Hyperspectral image (HSI) classification has become increasingly important in remote sensing applications, yet it faces significant challenges due to limited labeled samples and high computational costs. This research proposes the implementation of DMCM2 (Dual-Adjustment Cross-Domain Meta-Learner version 2), a lightweight few-shot learning framework specifically designed for HSI classification. The proposed method integrates TGAN2 (3D Ghost Attention Network version 2) as a feature extractor with only 260K parameters, significantly reducing computational requirements compared to conventional deep learning approaches. We employ a Class-Center based Mahalanobis (CCM) distance metric combined with intracorrection learning to enhance classification performance under extremely limited training samples. The framework will be evaluated on the Pavia University dataset using a 9-way 5-shot learning scenario, where only 5 labeled samples per class are available for training. This study aims to demonstrate that lightweight architectures can achieve competitive performance while maintaining computational efficiency, making HSI classification more accessible for resource-constrained environments.

**Index Terms**—Hyperspectral image classification, few-shot learning, meta-learning, lightweight deep learning, ghost convolution, attention mechanism

## I. INTRODUCTION

### A. Background

Hyperspectral imaging technology captures spectral information across hundreds of continuous narrow bands, enabling detailed material identification and classification in remote sensing applications [1]. Despite its rich spectral information, HSI classification remains challenging due to the curse of dimensionality and the scarcity of labeled training samples, which are expensive and time-consuming to obtain [2].

Traditional machine learning methods such as Support Vector Machines (SVM) and Random Forests have been widely applied to HSI classification [3]. However, these methods struggle to capture complex spatial-spectral relationships and require substantial labeled data to achieve satisfactory performance. Recent advances in deep learning, particularly Convolutional Neural Networks (CNNs), have shown promising results by automatically learning hierarchical feature representations [4]. Nevertheless, conventional deep learning approaches

demand large training datasets and significant computational resources, limiting their practical deployment.

Few-shot learning has emerged as a promising paradigm to address the limited labeled sample problem by enabling models to generalize from minimal training examples [5]. Meta-learning approaches, which learn to learn from few examples, have demonstrated remarkable success in computer vision tasks [6]. However, applying these techniques to HSI classification introduces unique challenges due to the high-dimensional spectral data and complex spatial-spectral correlations.

### B. Problem Statement

Current state-of-the-art deep learning models for HSI classification suffer from three major limitations:

- 1) **Data Scarcity:** HSI datasets typically contain limited labeled samples due to the high cost of ground truth annotation, leading to overfitting and poor generalization.
- 2) **Computational Complexity:** Existing deep learning architectures require millions of parameters and extensive computational resources, hindering deployment on edge devices and real-time applications.
- 3) **Domain Adaptation:** Models trained on one HSI dataset often fail to generalize to new scenes or sensor configurations without substantial retraining.

While recent few-shot learning methods such as DFSL, RN-FSC, and CMFSL have attempted to address these challenges [7]–[9], they still rely on heavyweight feature extractors with over 4 million parameters, making them impractical for resource-constrained scenarios.

### C. Research Objectives

This research aims to implement and evaluate the DMCM2 framework with the following specific objectives:

- 1) To develop a lightweight feature extraction network (TGAN2) with approximately 260K parameters that effectively captures spatial-spectral features from hyperspectral data.

- 2) To integrate the Class-Center based Mahalanobis (CCM) distance metric for robust few-shot classification under limited training samples.
- 3) To implement intracorrection learning mechanism that enhances model generalization by leveraging support set information during meta-training.
- 4) To evaluate the proposed framework on the Pavia University dataset under 9-way 5-shot learning scenario and compare performance against established baseline methods.
- 5) To demonstrate that lightweight architectures can achieve competitive accuracy while significantly reducing computational costs compared to existing approaches.

#### D. Research Contributions

The expected contributions of this research are:

- 1) Implementation of a complete few-shot learning pipeline for HSI classification with comprehensive preprocessing, training, and evaluation components.
- 2) Demonstration of parameter-efficient deep learning for remote sensing applications, showing that models with fewer than 300K parameters can effectively perform few-shot HSI classification.
- 3) Empirical analysis of the trade-off between model complexity and classification performance in few-shot learning scenarios.
- 4) Open-source implementation that facilitates reproducibility and enables further research in lightweight HSI classification methods.

## II. LITERATURE REVIEW

### A. Hyperspectral Image Classification

HSI classification aims to assign semantic labels to each pixel based on its spectral signature and spatial context. Early approaches relied on spectral features alone, treating each pixel independently [10]. The introduction of spatial information through morphological operations and spatial filtering improved classification accuracy [11]. More recently, joint spatial-spectral methods have become dominant, leveraging both types of information simultaneously [12].

### B. Deep Learning for HSI Classification

The application of deep learning to HSI classification began with 1D CNNs operating on spectral vectors [13]. Subsequently, 2D CNNs were introduced to capture spatial context [14], followed by 3D CNNs that jointly model spatial-spectral features [15]. Notable architectures include:

- **3D-CNN:** Pioneering work applying 3D convolutions to HSI classification [15].
- **SSRN:** Spectral-Spatial Residual Network using skip connections [16].
- **HybridSN:** Hybrid spectral-spatial CNN combining 3D and 2D convolutions [17].

Despite their success, these methods require thousands of labeled samples per class and millions of parameters, limiting their applicability in few-shot scenarios.

### C. Few-Shot Learning

Few-shot learning enables models to recognize new classes from minimal examples. Meta-learning approaches train models on multiple tasks, learning transferable knowledge that facilitates rapid adaptation [?]. Key paradigms include:

- **Metric Learning:** Learning discriminative embeddings where class prototypes can be computed from few examples. Representative methods include Matching Networks [?], Prototypical Networks [?], and Relation Networks [?].
- **Optimization-based:** Learning initialization parameters that enable rapid fine-tuning. Model-Agnostic Meta-Learning (MAML) is the seminal work [6].
- **Hallucination-based:** Generating synthetic samples to augment limited training data [?].

### D. Few-Shot Learning for HSI Classification

Recent works have adapted few-shot learning to HSI classification:

- **DFSL:** Deep Few-Shot Learning combining metric learning with data augmentation [7].
- **RN-FSC:** Relation Network for Few-Shot Classification using learnable distance metrics [8].
- **DCFSL:** Deep Cross-domain Few-Shot Learning addressing domain shift [?].
- **CMFSL:** Cross-domain Meta-learning for Few-Shot Learning with covariance-based metrics [9].

While these methods show improved performance over traditional approaches, they still employ heavyweight architectures with 4-10 million parameters.

### E. Lightweight Deep Learning

Lightweight architectures aim to reduce computational costs while maintaining performance:

- **MobileNets:** Depthwise separable convolutions reducing parameters and computations [?].
- **ShuffleNet:** Channel shuffle operations enabling efficient group convolutions [?].
- **GhostNet:** Generating more feature maps from cheap linear operations [?].

GhostNet version 2 introduces additional improvements including DFC attention and enhanced feature reuse [?], forming the basis for our TGAN2 architecture.

### F. Research Gap

Existing few-shot HSI classification methods have not adequately addressed the computational efficiency challenge. Most approaches prioritize accuracy over parameter efficiency, resulting in models unsuitable for edge deployment or real-time processing. Furthermore, limited work has explored the integration of lightweight architectures with advanced meta-learning techniques specifically for hyperspectral data. This

research addresses this gap by implementing a lightweight dual-branch meta-learner that balances performance and efficiency.

### III. METHODOLOGY

#### A. Dataset

We will use the Pavia University dataset, a widely-used benchmark for HSI classification research. The dataset specifications are:

- **Sensor:** ROSIS (Reflective Optics System Imaging Spectrometer)
- **Spatial dimensions:**  $610 \times 340$  pixels
- **Spectral bands:** 103 bands after removing noisy channels
- **Spectral range:** 430-860 nm
- **Spatial resolution:** 1.3 meters per pixel
- **Ground truth classes:** 9 land cover types

The nine classes include: Asphalt, Meadows, Gravel, Trees, Painted metal sheets, Bare Soil, Bitumen, Self-Blocking Bricks, and Shadows.

#### B. Data Preprocessing

1) *Spectral Dimensionality Reduction:* To reduce computational complexity and remove redundant information, we will apply Principal Component Analysis (PCA) to reduce spectral bands from 103 to 100, retaining over 99% of the variance:

$$X_{\text{PCA}} = X \cdot W_{\text{PCA}} \quad (1)$$

where  $X \in \mathbb{R}^{N \times 103}$  is the original data reshaped to  $N$  pixels, and  $W_{\text{PCA}} \in \mathbb{R}^{103 \times 100}$  are the principal components.

2) *Normalization:* Min-Max normalization will be applied to scale features to  $[0, 1]$ :

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (2)$$

3) *Patch Extraction:* Spatial patches of size  $9 \times 9 \times 100$  centered at each labeled pixel will be extracted using symmetric padding:

$$P_i = X_{\text{padded}}[h_i : h_i + 9, w_i : w_i + 9, :] \quad (3)$$

where  $(h_i, w_i)$  is the spatial location of pixel  $i$ .

#### C. Few-Shot Learning Setup

1) *Task Definition:* We adopt the  $N$ -way  $K$ -shot classification paradigm where  $N = 9$  classes and  $K = 5$  support samples per class. Each episode consists of:

- **Support set:**  $S = \{(x_i^s, y_i^s)\}_{i=1}^{NK}$  with  $K = 5$  examples per class
- **Query set:**  $Q = \{(x_j^q, y_j^q)\}_{j=1}^{NQ}$  with  $Q = 15$  examples per class
- **Test set:** Remaining samples for final evaluation

2) *Data Splitting Strategy:* For each class  $c$ :

- 1) Randomly select 5 samples for support set
- 2) Select next 15 samples for query set
- 3) Use remaining samples for test set

This ensures no overlap between support, query, and test sets while maintaining class balance.

#### D. Proposed Architecture: DMCM2

The DMCM2 framework consists of three main components:

1) *Feature Extractor:* TGAN2: TGAN2 is a lightweight 3D CNN based on Ghost Convolution version 2 [?]. The architecture comprises:

**GhostModule v2:** Generates feature maps efficiently by splitting into primary and cheap operations:

$$y = \Phi(x) = [\phi(x), \psi(\phi(x))] \quad (4)$$

where  $\phi$  is primary convolution and  $\psi$  is depthwise cheap operation.

**GA2 Block:** Combines GhostModule with multi-dimensional convolutions:

$$F_{\text{GA2}} = \text{DFC}(f_d + f_h + f_w) \quad (5)$$

where  $f_d, f_h, f_w$  are convolutions along depth, height, and width dimensions, and DFC is the Decoupled Fully Connected attention mechanism.

**NAM Attention:** Normalizes and weights feature channels:

$$F_{\text{out}} = \sigma \left( F \cdot \frac{\gamma}{\sum \gamma + \epsilon} \right) \cdot F \quad (6)$$

where  $\gamma$  are learnable weights and  $\sigma$  is sigmoid activation.

#### Network Structure:

- 1) Initial 3D convolution:  $1 \rightarrow 8$  channels
- 2) First block: NAM + GA2 + NAM + Residual + AvgPool
- 3) Second block:  $1 \times 1$  conv + NAM + GA2 + NAM + Residual + AvgPool
- 4) Global feature extraction through flattening

The output is a feature vector  $f \in \mathbb{R}^d$  where  $d$  is the feature dimension.

2) *CCM Distance Metric:* The Class-Center based Mahalanobis distance measures similarity considering class covariance:

$$d_{\text{CCM}}(x, c) = (f(x) - \mu_c)^T \Sigma_c^{-1} (f(x) - \mu_c) \quad (7)$$

where:

- $f(x)$  is the feature representation
- $\mu_c$  is the class prototype (mean of support features)
- $\Sigma_c$  is the class covariance matrix with regularization:

$$\Sigma_c = \frac{1}{K-1} \sum_{i=1}^K (f_i - \mu_c)(f_i - \mu_c)^T + \lambda I \quad (8)$$

Classification is performed by:

$$\hat{y} = \arg \min_c d_{\text{CCM}}(x, c) \quad (9)$$

3) *Intracorrection Learning:* Intracorrection (IC) enhances meta-learning by also classifying support samples:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{query}} + \lambda_{\text{IC}} \mathcal{L}_{\text{support}} \quad (10)$$

where:

$$\mathcal{L}_{\text{query}} = - \sum_{i=1}^{NQ} \log p(y_i^q | x_i^q, S) \quad (11)$$

$$\mathcal{L}_{\text{support}} = - \sum_{i=1}^{NK} \log p(y_i^s | x_i^s, S) \quad (12)$$

This forces the model to learn better class boundaries even within the support set.

### E. Training Procedure

1) *Episodic Training*: Training follows an episodic paradigm as shown in Algorithm 1.

```

for epoch = 1 to  $N_{\text{epochs}}$  do
  for episode = 1 to  $N_{\text{episodes}}$  do
    Sample  $N$ -way  $K$ -shot task from dataset
    Extract features:  $F_s = f_\theta(S)$ ,  $F_q = f_\theta(Q)$ 
    Compute prototypes and covariances
    Calculate  $\mathcal{L}_{\text{total}}$ 
    Update  $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\text{total}}$ 
  end for
end for

```

2) *Hyperparameters*: Planned training configuration:

- **Optimizer**: Adam
- **Learning rate**: 0.001
- **Learning rate schedule**: StepLR (decay 0.5 every 10 epochs)
- **Batch size**: 1 episode per iteration
- **Episodes per epoch**: 300
- **Total epochs**: 20
- **IC weight**  $\lambda_{\text{IC}}$ : 1.0
- **Covariance regularization**  $\lambda$ : 0.01

### F. Evaluation Metrics

1) *Classification Metrics*: We will report the following standard metrics:

$$\text{OA} = \frac{\sum_{i=1}^C n_{ii}}{N} \quad (13)$$

$$\text{AA} = \frac{1}{C} \sum_{i=1}^C \frac{n_{ii}}{n_i} \quad (14)$$

$$\kappa = \frac{N \sum_{i=1}^C n_{ii} - \sum_{i=1}^C (n_i \cdot n_i)}{N^2 - \sum_{i=1}^C (n_i \cdot n_i)} \quad (15)$$

where:

- OA is Overall Accuracy
- AA is Average Accuracy
- $\kappa$  is Cohen's Kappa coefficient
- $n_{ii}$  is the number of correctly classified samples in class  $i$
- $n_i$  is the total number of samples in class  $i$
- $C$  is the number of classes
- $N$  is the total number of test samples

2) *Computational Efficiency*: We will measure:

- Total parameters
- Training time per epoch
- Inference time per sample
- GPU memory consumption

3) *Visualization*: We will generate:

- Confusion matrices (absolute and normalized)
- Classification maps comparing predictions with ground truth
- Training curves (loss and accuracy)
- Per-class accuracy analysis

### G. Baseline Comparisons

We will compare our results against the following methods as reported in the literature:

- **Classical ML**: RBF-SVM
- **Deep Learning**: 3D-CNN, SSRN
- **Few-Shot Learning**: DFSL, RN-FSC, DCFSL, Gi-CFSL, CMFSL
- **Lightweight Meta-learning**: DMCM (v1)
- **State-of-the-art**: DMCM2 (as reported in paper)

## IV. EXPECTED RESULTS AND DISCUSSION

### A. Expected Performance

Based on the original DMCM2 paper, we anticipate achieving:

- Overall Accuracy (OA): 85-95%
- Average Accuracy (AA): 85-95%
- Kappa coefficient: 0.80-0.90

Note that our implementation may achieve lower accuracy than the paper's reported 97.95% OA due to:

- 1) Reduced training iterations (20 epochs vs. 10,000 iterations)
- 2) Fewer episodes per epoch (300 vs. 1,000)
- 3) Simplified implementation for demonstration purposes

### B. Parameter Efficiency

TGAN2 is expected to contain approximately 260K parameters, which is:

- 94% fewer parameters than SSRN ( $\sim 4.2\text{M}$ )
- 90% fewer parameters than typical 3D-CNNs
- Comparable to MobileNet-based architectures

This dramatic reduction enables:

- Faster training and inference
- Lower memory footprint
- Deployment on edge devices
- Reduced environmental impact

### C. Ablation Studies

We plan to conduct ablation studies examining:

- 1) Impact of IC learning: Comparing  $\lambda_{\text{IC}} = 0$  vs.  $\lambda_{\text{IC}} = 1$
- 2) Effect of attention mechanisms: Removing NAM or DFC attention
- 3) Patch size sensitivity: Testing  $5 \times 5$ ,  $7 \times 7$ ,  $9 \times 9$ ,  $11 \times 11$
- 4) Support set size: Varying  $K$  from 1 to 10 shots

#### D. Limitations

Anticipated limitations include:

- Performance gap compared to heavyweight models when abundant training data is available
- Sensitivity to initial support set selection
- Potential difficulty in extremely imbalanced scenarios
- Limited evaluation on a single dataset

#### E. Future Improvements

Potential enhancements for future work:

- Cross-dataset evaluation to assess generalization
- Integration with self-supervised pre-training
- Application to other HSI datasets (Indian Pines, Salinas, etc.)
- Extension to transductive few-shot learning
- Exploration of neural architecture search for automated design

### V. PROJECT TIMELINE

TABLE I  
PROJECT TIMELINE

Week	Activity	Status
1-2	Literature review and proposal	Planned
3-4	Dataset preparation and preprocessing	Planned
5-6	TGAN2 architecture implementation	Planned
7-8	CCM metric and IC learning implementation	Planned
9-10	Model training and hyperparameter tuning	Planned
11-12	Evaluation and result analysis	Planned
13-14	Report writing and documentation	Planned
15	Final presentation preparation	Planned

### VI. RESOURCE REQUIREMENTS

#### A. Computational Resources

- **GPU:** NVIDIA T4 or equivalent (Google Colab)
- **RAM:** 16GB minimum
- **Storage:** 5GB for dataset and models
- **Software:** Python 3.8+, PyTorch 1.10+, CUDA 11.0+

#### B. Software Dependencies

- **PyTorch:** Deep learning framework
- **NumPy:** Numerical computations
- **scikit-learn:** PCA, metrics, preprocessing
- **SciPy:** Mathematical operations
- **Matplotlib, Seaborn:** Visualization
- **Spectral Python:** HSI data handling

### VII. CONCLUSION

This research proposal outlines a comprehensive plan to implement and evaluate DMCM2, a lightweight dual-branch meta-learner for few-shot hyperspectral image classification. By integrating TGAN2 feature extraction with CCM distance metric and intracorrection learning, we aim to demonstrate that parameter-efficient architectures can achieve competitive performance in few-shot scenarios.

The proposed approach addresses critical challenges in HSI classification: limited labeled samples, high computational costs, and poor generalization. With only 260K parameters, TGAN2 offers a practical solution for resource-constrained environments while maintaining classification accuracy comparable to heavyweight alternatives.

Upon successful completion, this project will contribute to the growing body of research on efficient deep learning for remote sensing, demonstrating that architectural innovations can balance performance and efficiency. The open-source implementation will enable reproducibility and facilitate further research in lightweight HSI classification methods.

The expected outcomes include comprehensive evaluation results, detailed performance analysis, and practical insights into the trade-offs between model complexity and few-shot learning capability. These findings will be valuable for both academic research and practical applications in remote sensing, precision agriculture, environmental monitoring, and urban planning.

### REFERENCES

- [1] A. Plaza, J. A. Benediktsson, J. W. Boardman et al., “Recent advances in techniques for hyperspectral image processing,” *Remote Sensing of Environment*, vol. 113, pp. S110–S122, 2009.
- [2] M. Fauvel, Y. Tarabalka, J. A. Benediktsson et al., “Advances in spectral-spatial classification of hyperspectral images,” *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, 2013.
- [3] F. Melgani and L. Bruzzone, “Classification of hyperspectral remote sensing images with support vector machines,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.
- [4] L. Zhang, L. Zhang, and B. Du, “Deep learning for remote sensing data: A technical tutorial on the state of the art,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 22–40, 2016.
- [5] Y. Wang et al., “Generalizing from a few examples: A survey on few-shot learning,” *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–34, 2020.
- [6] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” *Proc. ICML*, pp. 1126–1135, 2017.
- [7] Y. Gao et al., “Deep feature-based framework for hyperspectral image classification,” *Remote Sensing*, vol. 11, no. 14, p. 1617, 2019.
- [8] Y. Li et al., “Relation network for few-shot classification of hyperspectral images,” *IEEE GRSL*, vol. 17, no. 9, pp. 1555–1559, 2020.
- [9] J. Zhang et al., “Cross-domain meta-learning for few-shot hyperspectral image classification,” *IEEE TGRS*, vol. 59, no. 8, pp. 6856–6871, 2021.
- [10] D. Landgrebe, “Hyperspectral image data analysis,” *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 17–28, 2002.
- [11] M. Dalla Mura et al., “Morphological attribute profiles for the analysis of very high resolution images,” *IEEE TGRS*, vol. 48, no. 10, pp. 3747–3762, 2010.
- [12] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, “Deep learning-based classification of hyperspectral data,” *IEEE JSTARS*, vol. 7, no. 6, pp. 2094–2107, 2014.
- [13] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, “Deep convolutional neural networks for hyperspectral image classification,” *Journal of Sensors*, 2015.
- [14] W. Zhao and S. Du, “Spectral-spatial feature extraction for hyperspectral image classification,” *IEEE JSTARS*, vol. 9, no. 10, pp. 4799–4807, 2016.
- [15] Y. Li, H. Zhang, and Q. Shen, “Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network,” *Remote Sensing*, vol. 9, no. 1, p. 67, 2017.
- [16] L. Zhong et al., “Spectral-spatial residual network for hyperspectral image classification,” *Remote Sensing*, vol. 10, no. 1, p. 107, 2018.
- [17] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, “HybridSN: Exploring 3-D-2-D CNN feature hierarchy for hyperspectral image classification,” *IEEE GRSL*, vol. 17, no. 2, pp.