NATIONAL TECHNICAL UNIVERSITY OF UKRAINE
“IGOR SIKORSKY KYIV POLYTECHNIC INSTITUTE”

Educational and Research Institute for Applied System Analysis

Department of Artificial Intelligence

«Manuscript Rights»
UDC 004.8:528.8:629.735

Approved for defense:

Acting head
at the Department of AI
Iryna DZHYGYREY

_______________2025

Master’s Thesis

for the degree of Master

in the Educational and Professional Program
«Systems and Methods of Artificial Intelligence»

Specialty 122 «Computer Sciences»

on the topic: «SpectralCA: Hybrid Neural Network
for UAV-Based Hyperspectral Landmine Detection»

Prepared by:
2nd-year student, group KI-41mp
Brovko Danil Vitaliiovych __________

Scientific Supervisor:
Professor of the Department of AI, D.Sc. (Tech.), Professor
Sineglazov Viktor Mykhailovych __________

Reviewer:
Professor of the Department of Aviation Computer Integrated Technologies,
State University "Kyiv Aviation Institute",
D.Sc. (Tech.), Professor
Sushchenko Olha Andriivna __________

Hereby I certify that this master’s dissertation does not contain any borrowed material from the works of other authors without appropriate references.

Student _____________

Kyiv – 2025

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Навчально-науковий інститут прикладного системного аналізу

Кафедра штучного інтелекту

«На правах рукопису»
УДК 004.8:528.8:629.735

До захисту допущено:

В. о. завідувачки кафедри ШІ
Ірина ДЖИГИРЕЙ

«___»____________2025 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою
«Системи і методи штучного інтелекту»

зі спеціальності 122 «Комп'ютерні науки» на тему:

«SpectralCA – гібридна нейронна мережа
для гіперспектрального виявлення мін за допомогою БПЛА»

Виконав:
студент ІІ курсу, групи КІ-41мп
Бровко Даніл Віталійович ___________

Науковий керівник:
професор кафедри ШІ, д.т.н., професор
Синєглазов Віктор Михайлович ___________

Рецензент:
професор кафедри авіаційних комп’ютерно інтегрованих комплексів
ДУ «Київський Авіаційний Інститут», д.т.н., професор
Сущенко Ольга Андріївна ___________

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент ______________

Київ – 2025 року

National Technical University of Ukraine

“Igor Sikorsky Kyiv Polytechnic Institute”

Educational and Research Institute for Applied System Analysis

Department of Artificial Intelligence

Level of higher education – second (master's)

Specialty (program subject area) – 122 «Computer sciences»

Educational and Professional Program «Systems and Methods of Artificial Intelligence»

APPROVED

Acting head of the Department of AI

Iryna DZHYGYREY

29th August 2025

ASSIGNMENT
for the master’s dissertation of

Brovko Danil Vitaliiovych

1. The topic of the research is «SpectralCA: Hybrid Neural Network for UAV-Based Hyperspectral Landmine Detection». Work’s supervisor is Sineglazov Viktor Mykhailovych, PhD, Professor at the Department of AI. It is approved by the ER IASA Order # 4837-c of 6th November 2025 .

2. Submission date: 12th December 2025.

3. Object of the research: hyperspectral image classification.

4. Subject of the research: development of a high-performance deep learning architecture for a classification task in hyperspectral imaging.

5. Initial data for the research: hyperspectral images with labels for each pixel.

6. Stages of the work:

1) Theoretical research on hyperspectral imaging applications and review of existing classification methods, including CNNs, Transformers, and hybrid networks.
2) Analysis of semi-supervised learning techniques, specifically self-training and proxy labeling, to address the scarcity of labeled hyperspectral data.
3) Design and mathematical modeling of the SpectralCA (Spectral Cross-Attention) block to fuse spectral and spatial features effectively.
4) Development of the hybrid deep learning architecture by modifying the Mobile 3D Vision Transformer (MDvT) to include the proposed SpectralCA block.

5) Software implementation of the architecture and the semi-supervised learning pipeline using the PyTorch framework.
6) Training and benchmarking the model on the WHU-Hi-HongHu dataset; conducting comparative analysis of Average Accuracy, Overall Accuracy, Kappa Coefficient, inference time, and parameter count against the baseline MobileViTBlock.
7) Analysis of the startup implementation for mine detection utilising the proposed SpectralCA model.
8) Conclusion.

7. List of graphic materials: presentation.

8. Approximate list of publications:

1) D.V. Brovko. SpectralCA: Bi-Directional Cross-Attention for Next-Generation UAV Hyperspectral Vision. arXiv, 2025. Pre-print. URL: https://arxiv.org/pdf/2510.09912.
2) D.V. Brovko. SpectralCA: Utilising Hyper-Spectral Imaging in UAV Navigation. IEEE 8th International Conference on Methods and Systems of Navigation and Motion Control, 2025. Already presented, in print. URL: https://msnmc.ieee.org.ua/wp-content/uploads/2025/10/Program_MSNMC2025.pdf.
3) D.V. Brovko. SpectralCA: Bi-Directional Cross-Attention for Next-Generation UAV Hyperspectral Vision. International Student Research Competition in Artificial Intelligence, 2025. First place in the competition. URL: https://aic.kpi.ua/wp-content/uploads/2025/11/spectralca.pdf.

9. Date of the task issue: 29th August 2025.

Calendar plan

№ by number	Stage of the master's dissertation	Deadline	Notes
1	Review and analysis of studies on the topic of the work	31st October 2025	Completed
2	Development of the high-performance deep learning architecture for a classification task in hyperspectral imaging	14th November 2025	Completed
3	Analysis of the results	21st November 2025	Completed
4	Analysis of the startup implementation	28th November 2025	Completed
5	Write the paper	5th December 2025	Completed
6	Create the presentation	11th December 2025	Completed

Student Danil BROVKO

Supervisor Viktor SINEGLAZOV

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Навчально-науковий інститут прикладного системного аналізу

Кафедра штучного інтелекту

Рівень вищої освіти – другий (магістерський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Системи і методи штучного інтелекту»

ЗАТВЕРДЖУЮ

В. о. завідувачки кафедри ШІ

Ірина ДЖИГИРЕЙ

«29» серпня 2025 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Бровку Данилу Віталійовичу

1. Тема дисертації «SpectralCA – гібридна нейронна мережа для гіперспектрального виявлення мін за допомогою БПЛА», науковий керівник дисертації Синєглазов Віктор Михайлович, д.т.н., професор кафедри ШІ, затверджені наказом по ННІПСА від «06» листопада 2025 р. № 4837-с.

2. Термін подання студентом дисертації «12» грудня 2025 р.

3. Об’єкт дослідження: класифікація гіперспектральних зображень.

4. Предмет дослідження: розробка високоефективної архітектури глибокого навчання для класифікації гіперспектральних даних.

5. Вихідні дані: гіперспектральні зображення з мітками для кожного пікселя.

6. Перелік завдань:

1) Теоретичне дослідження застосувань гіперспектральних даних та огляд існуючих методів класифікації, включаючи CNN, Transformers та гібридні мережі.
2) Аналіз методів напівкерованого навчання, зокрема самонавчання та проксі-маркування, для вирішення проблеми нестачі маркованих гіперспектральних даних.
3) Проектування та математичне моделювання блоку SpectralCA (Spectral Cross-Attention) для ефективного об'єднання спектральних та просторових ознак.
4) Розробка гібридної архітектури глибокого навчання шляхом модифікації Mobile 3D Vision Transformer (MDvT) з метою застосування запропонованого блоку SpectralCA.

5) Програмна реалізація архітектури та напівкерованого навчання з використанням фреймворку PyTorch.
6) Навчання та тестування моделі на наборі даних WHU-Hi-HongHu; проведення порівняльного аналізу середньої точності, загальної точності, коефіцієнта Каппа, часу інференсу та кількості параметрів у порівнянні з базовим MobileViTBlock.
7) Аналіз реалізації стартапу для виявлення мін із використанням запропонованої моделі SpectralCA.
8) Висновки.

7. Перелік ілюстративного матеріалу: презентація.

8. Орієнтовний перелік публікацій:

1) D.V. Brovko. SpectralCA: Bi-Directional Cross-Attention for Next-Generation UAV Hyperspectral Vision. arXiv, 2025. Препринт. URL: https://arxiv.org/pdf/2510.09912.
2) D.V. Brovko. SpectralCA: Utilising Hyper-Spectral Imaging in UAV Navigation. IEEE 8th International Conference on Methods and Systems of Navigation and Motion Control, 2025. Апробовано, тези – у друці. URL: https://msnmc.ieee.org.ua/wp-content/uploads/2025/10/Program_MSNMC2025.pdf.
3) D.V. Brovko. SpectralCA: Bi-Directional Cross-Attention for Next-Generation UAV Hyperspectral Vision. International Student Research Competition in Artificial Intelligence, 2025. Перше місце на змаганні. URL: https://aic.kpi.ua/wp-content/uploads/2025/11/spectralca.pdf.

9. Дата видачі завдання: «29» серпня 2025 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Огляд та аналіз досліджень за темою роботи	31 жовтня 2025	Виконано
2	Розробка високоефективної архітектури глибокого навчання для класифікації гіперспектральних даних	14 листопада 2025	Виконано
3	Аналіз результатів	21 листопада 2025	Виконано
4	Аналіз реалізації стартапу	28 листопада 2025	Виконано
5	Оформлення записки	5 грудня 2025	Виконано
6	Створення презентації	11 грудня 2025	Виконано

Студент Даніл БРОВКО

Науковий керівник Віктор СИНЄГЛАЗОВ

ABSTRACT

Master's dissertation: 111 p., 29 figures, 20 tables, 41 references, and 2 appendices.

Keywords: SpectralCA, deep learning, computer vision, hyperspectral imaging, unmanned aerial vehicle, object detection, semi-supervised learning.

The relevance of this research lies in the growing demand for unmanned aerial vehicles (UAVs) capable of operating reliably in complex environments where conventional navigation becomes unreliable due to interference, poor visibility, or camouflage. Hyperspectral imaging (HSI) provides unique opportunities for UAV-based computer vision by enabling fine-grained material recognition and object differentiation, which are critical for navigation, surveillance, agriculture, and environmental monitoring.

The object of the study is hyperspectral image classification.

The subject of the research is development of a high-performance deep learning architecture for a classification task in hyperspectral imaging.

The purpose of this work is to modify the Mobile 3D Vision Transformer (MDvT) by introducing the proposed SpectralCA block. This block employs bi-directional cross-attention to fuse spectral and spatial features, enhancing accuracy while reducing parameters and inference time. Experimental evaluation was conducted on the WHU-Hi-HongHu dataset, with results assessed using Overall Accuracy, Average Accuracy, and the Kappa coefficient.

The findings confirm that the proposed architecture improves UAV perception efficiency, enabling real-time operation for navigation, object recognition, and environmental monitoring tasks.

Versions of this work were published in arXiv and presented at the 2025 IEEE 8th International Conference on Methods and Systems of Navigation and Motion Control. Also, this research won first place at the 2025 International Student Research Competition in Artificial Intelligence.

РЕФЕРАТ

Магістерська дисертація: 111 с., 29 рис., 20 табл., 41 посилання, 2 додатки.

Ключові слова: SpectralCA, глибоке навчання, комп'ютерний зір, гіперспектральне бачення, безпілотний літальний апарат, виявлення об'єктів, напівкероване навчання.

Актуальність: зростає попит на безпілотні літальні апарати (БПЛА), здатні надійно працювати в складних умовах, де традиційна навігація стає ненадійною через перешкоди, погану видимість або маскування. Гіперспектральна візуалізація (HSI) надає унікальні можливості для комп'ютерного зору на базі БПЛА, забезпечуючи точне розпізнавання матеріалів та диференціацію об'єктів, що є критично важливим для навігації, моніторингу навколишнього середовища тощо.

Об’єкт дослідження: класифікація гіперспектральних зображень.

Предмет дослідження: розробка високоефективної архітектури глибокого навчання для завдання класифікації в гіперспектральній візуалізації.

Мета роботи: модифікація Mobile 3D Vision Transformer (MDvT) шляхом впровадження нового блоку SpectralCA. Він використовує двонаправлену перехресну увагу для об'єднання спектральних і просторових ознак, підвищуючи точність та одночасно зменшуючи параметри та час інференсу. Оцінка була проведена на наборі даних WHU-Hi-HongHu, а результати порівняли за допомогою коефіцієнта Каппа, загальної та середньої точностей.

Результати підтверджують, що запропонована архітектура покращує ефективність сприйняття БПЛА.

Версії цієї роботи були опубліковані в arXiv і представлені на 2025 IEEE 8th International Conference on Methods and Systems of Navigation and Motion Control. Також це дослідження посіло перше місце на 2025 International Student Research Competition in Artificial Intelligence.

CONTENTS

LIST OF ABBREVIATIONS

AI – Artificial Intelligence

CV – Computer Vision

HSI – Hyperspectral Imaging

UAV – Unmanned Aerial Vehicle

GNSS – Global Navigation Satellite System

AA – Average Accuracy

OA – Overall Accuracy

CNN – Convolutional Neural Network

1D – One-Dimensional

2D – Two-Dimensional

3D – Three-Dimensional

ViT – Vision Transformer

MDvT – Mobile 3D Vision Transformer

SSL – Semi-Supervised Learning

SpectralCA – Spectral Cross-Attention

IP – Intellectual Property

INTRODUCTION

The full-scale invasion of Ukraine has resulted in the widespread contamination of vast territories with explosive ordnance, creating one of the most severe security challenges of the 21st century. According to recent government estimates, over 139,000 square kilometers of the country's territory remain potentially hazardous [1]. This unprecedented scale of contamination has overwhelmed official demining capacities, forcing the agricultural sector into a desperate situation. As highlighted in recent investigative reports, farmers unable to wait for state clearance are resorting to life-threatening improvisation – modifying tractors with makeshift armor or even using garden rakes to manually clear fields of explosives [2]. This reality underscores a critical technological gap: there is an urgent need for a solution that is safer than manual labor, yet more scalable and cost-effective than heavy industrial demining machinery.

In this context, Unmanned Aerial Vehicles (UAVs) have become an essential platform for remote sensing operations. While originally developed for applications ranging from environmental monitoring to autonomous navigation [3-6], their role is evolving toward non-contact hazardous area assessment. However, typical onboard perception systems, which rely on GNSS, inertial measurement units, and standard RGB cameras [4, 7], face significant limitations in demining scenarios. Conventional optical sensors often fail when external conditions, such as vegetation cover, camouflage, or soil mineralization, reduce the visual contrast of hazardous objects [6, 8].

Hyperspectral imaging (HSI) offers a revolutionary solution to these limitations by capturing reflectance across hundreds of narrow spectral bands [9-11]. Unlike RGB cameras that rely on visual shape and color, HSI enables the differentiation of materials based on their unique chemical composition and spectral signature [12-14]. This capability makes HSI particularly valuable for identifying "invisible" threats: distinct spectral anomalies allow for the separation of artificial polymers (such as plastic PFM-1 mines) from natural backgrounds like grass or soil, even when they are visually indistinguishable. Consequently, HSI transforms the UAV into a precision instrument for object detection, land-cover classification, and fine-grained environmental analysis [15-17].

Despite its potential, processing high-dimensional hyperspectral data on-board resource-constrained UAVs presents a significant computational challenge. Recent studies have explored dimensionality reduction and deep learning-based spectral–spatial analysis to address this efficiency gap [18-20].

The objective of this research is to develop a new Computer Vision (CV) module to enhance UAV autonomy by simultaneously refining CV capabilities for the recognition of concealed hazardous objects and improving navigation accuracy in GNSS-denied environments.

The scientific novelty of this research lies in the development of the SpectralCA (Spectral Cross-Attention) module, which integrates hyperspectral sensing with deep learning, specifically leveraging a bi-directional cross-attention mechanism [21-23] that effectively fuses spectral and spatial features [20, 24-26]. Unlike existing methods that process spatial and spectral features sequentially or independently, the proposed approach facilitates symmetric interaction between the two domains. This allows for the effective extraction of fine-grained spectral signatures of concealed objects while significantly reducing the computational complexity (parameter count and inference time) required for real-time UAV operation.

To validate the architecture, benchmark datasets such as WHU-Hi are utilized [27]. Due to the scarcity of large-scale labeled hyperspectral datasets containing landmines sufficient for training deep transformer models, this research utilizes the WHU-Hi-HongHu agricultural benchmark as a scientifically valid proxy and a pre-training foundation. While real-world samples exist, they are often insufficient in volume for training complex Transformer architectures. The selected proxy dataset mimics the key challenges of mine detection: high spatial resolution, small object segmentation, and high inter-class spectral similarity (e.g., distinguishing between spectrally similar plant cultivars serves as a mathematical analogue to distinguishing between disturbed soil and camouflaged explosives).

Consequently, this work validates the first stage of a transfer learning pipeline, where the model learns robust general spectral representations from diverse available HSI data, establishing a feature extraction backbone that can subsequently be fine-tuned on limited real-world demining samples.

The practical significance of the work results in the creation of a software module capable of deployment on resource-constrained edge devices. This enables the conversion of commercial UAVs into autonomous mine-detection platforms. Furthermore, the validated semi-supervised learning pipeline addresses the critical shortage of labeled data in combat zones, allowing the system to adapt to new environments using unlabelled imagery. The proposed solution offers a scalable, cost-effective alternative to heavy demining machinery for humanitarian clearance operations.

The work consists of an introduction, four chapters, conclusions, and appendices:

- Chapter 1 analyzes the theoretical foundations of hyperspectral imaging and reviews existing deep learning architectures for remote sensing;
- Chapter 2 presents the mathematical modeling and topological design of the proposed SpectralCA module;
- Chapter 3 details the experimental evaluation, including the semi-supervised learning implementation and comparative analysis against state-of-the-art benchmarks;
- Chapter 4 outlines the startup implementation strategy, including the business model, financial analysis, and hardware integration plan for bringing the technology to the Ukrainian market.

CHAPTER 1 HYPERSPECTRAL IMAGING AND CLASSIFICATION METHODS FOR UAV-BASED LANDMINE DETECTION

1.1 Hyperspectral Images: Definition and Application

A hyperspectral image (HSI) is a three-dimensional data array, or hypercube, capturing a continuous reflection spectrum for each pixel across dozens or hundreds of narrow wavelength bands. HSI combines digital photography with spectroscopy, gathering information across the electromagnetic spectrum. Unlike standard RGB images with only 3 channels, hyperspectral images contain dozens or hundreds of spectral bands with very high spectral resolution. This detailed pixel-level spectrum enables identification of spectral signatures – unique material "fingerprints" used to recognize objects and their composition [9]. For example, characteristic spectra can identify chlorophyll in plants or petroleum hydrocarbons in soil.

Hyperspectral imaging technology originated in Earth remote sensing. NASA began developing hyperspectral scanners for aerospace applications in the 1980s [10]. Over three decades, HSI has become a powerful tool providing phenomenal spectral and spatial detail for Earth surface studies [9].

Figure 1.1 shows the WHU-Hi-HongHu dataset, one of the most common HSI datasets.

Initially used for mineral mapping, agriculture, and land-use planning, the technology has expanded to other fields. HSI's information richness enables diverse military, environmental, and civilian applications. In remote sensing, HSI is used for detailed land cover classification, vegetation typing, soil and water monitoring, and detecting camouflage or mines. In agriculture, hyperspectral imaging assesses crop health, identifies plant species, and detects pest or disease damage [17]. For food security and industry, HSI enables non-invasive quality checks and contamination detection. Medical applications are equally promising: tissue imaging provides diagnostic information about physiology and morphology without biopsy [18]. For example, HSI data can distinguish healthy from cancerous tissues by spectral features. Hyperspectral systems are thus considered promising for tasks ranging from environmental monitoring to quality control and disease diagnosis.

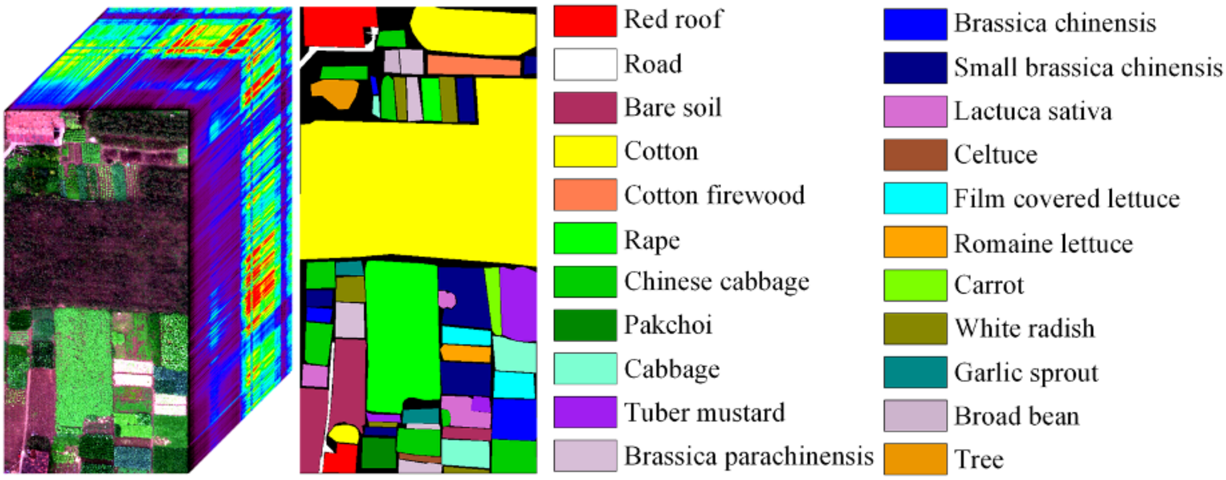

Figure 1.1 – WHU-Hi-HongHu dataset [27]

This versatility has made HSI a focus of intensive analysis. A key task is HSI classification – automatically recognizing each pixel's class based on the material or object it represents.

1.2 Methods for Classifying Hyperspectral Images

1.2.1 CNNs

Convolutional Neural Networks (CNNs) are widely used for hyperspectral image (HSI) classification due to their ability to automatically extract spatial and spectral features [13, 16, 20, 26]. Different variants exist: 1D-CNNs process spectral vectors, 2D-CNNs capture spatial context, and 3D-CNNs jointly model spatial–spectral interactions [16, 20, 28]. Hybrid designs such as HybridSN combine 3D and 2D convolutions (Fig. 1.2), improving classification accuracy by capturing both local textures and spectral signatures [20]. Recent CNN-based frameworks have also been evaluated for land-cover mapping tasks, demonstrating strong generalization performance [29]. In addition, multi-channel CNN architectures further improve spectral–spatial representation through parallel convolutional streams [30]. However, 3D-CNNs are computationally expensive, with a large number of parameters and high training time, which can limit their deployment on UAVs [3, 8].

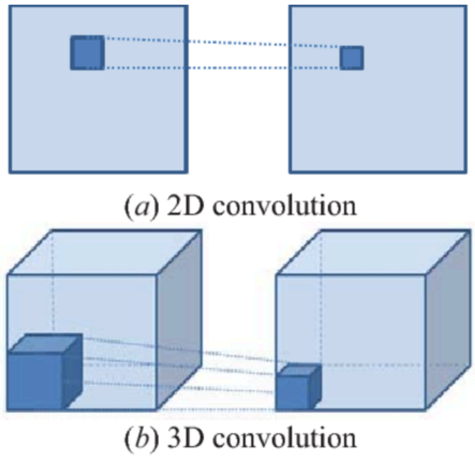

Figure 1.2 – Comparison of (a) 2D convolution and (b) 3D convolution [28]

1.2.2 Transformers

Transformers, originally developed for natural language processing, have been adapted to HSI analysis through the self-attention mechanism, which models long-range dependencies across spectral and spatial domains [23, 24]. Unlike CNNs, which focus on local receptive fields, Transformers can capture global relationships, enabling more precise material and terrain classification [24, 25]. Models such as SpectralFormer (Fig. 1.3) treat the spectral bands of a pixel as sequences, achieving state-of-the-art accuracy in HSI classification [24].

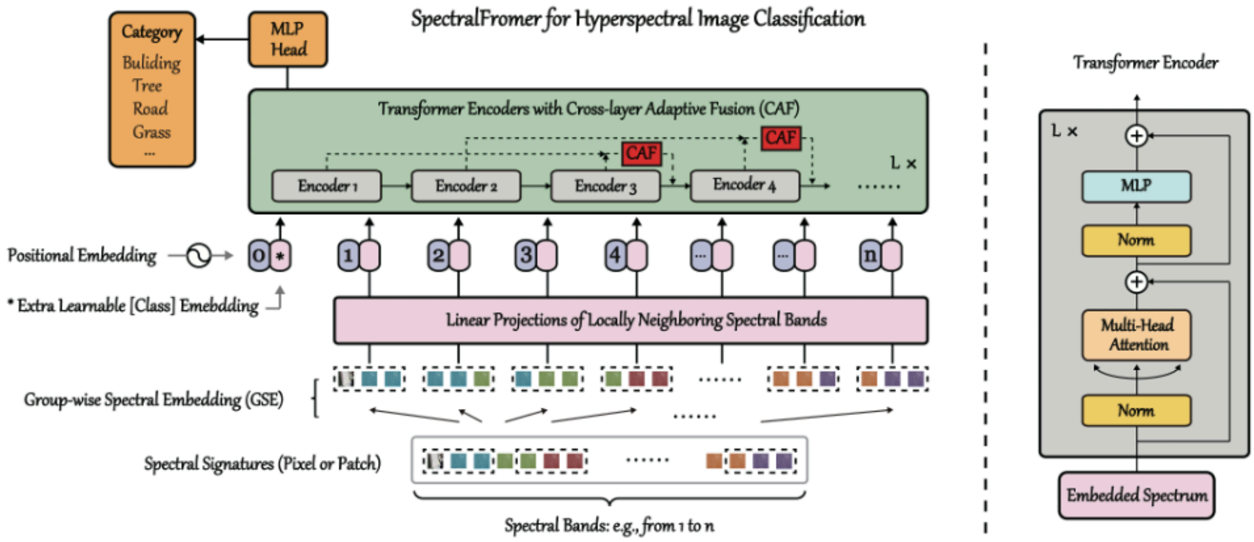

Figure 1.3 – Topology of the SpectralFormer model [24]

In UAV remote sensing, fusing hyperspectral imagery with LiDAR data provides complementary information: spectral features describe material properties, while LiDAR offers structural and elevation data. Recent methods employ cross-attention Transformers, where LiDAR tokens act as queries and HSI features as keys/values (Fig. 1.4).

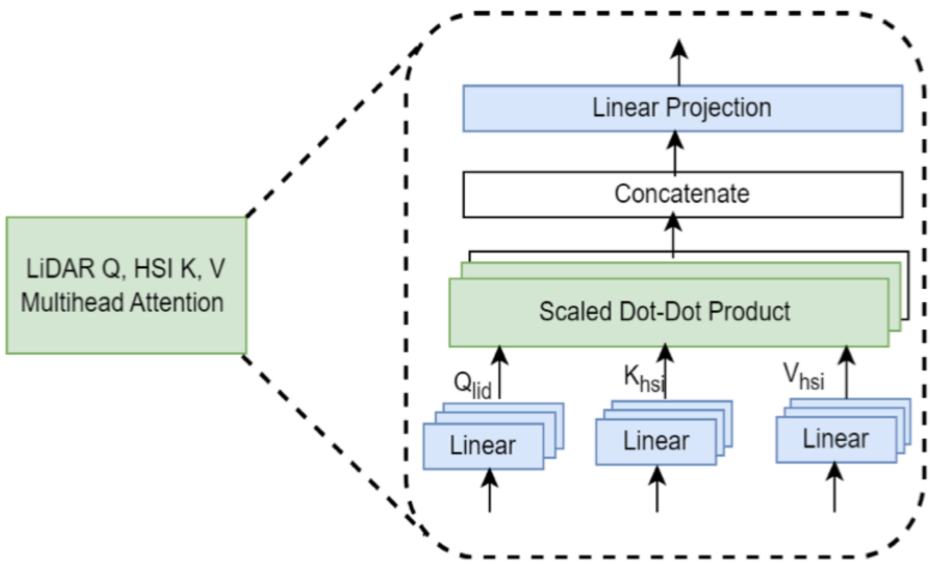

Figure 1.4 – Cross-attention combining LiDAR and HSI [21]

This allows the model to align elevation and spectral features, enhancing land-cover classification and terrain analysis. For UAV navigation, such fusion enables more robust obstacle detection, terrain mapping, and environmental monitoring in complex environments [21, 22, 31].

1.2.3 Hybrid Networks

The Mobile 3D Convolutional Vision Transformer (MDvT) integrates lightweight 3D convolutions with Vision Transformer blocks (Fig. 1.5). The 3D convolutions extract local spectral–spatial features, while the Transformer captures global dependencies. By embedding mobile convolutional layers (inspired by MobileNet inverted residual blocks), MDvT reduces parameter count and accelerates inference, making it suitable for UAV onboard processing. This hybrid approach achieves higher accuracy than pure CNN or Transformer architectures while maintaining efficiency [8, 25].

1.3 Semi-Supervised Learning

Deep learning demonstrates high accuracy in classifying hyperspectral images (HSI), but its effectiveness largely depends on the amount of labeled data available. In HSI tasks, labeling pixels for training is extremely time-consuming and resource-intensive, as each pixel is associated with a spectrum and often requires consultation with experts in geology, agronomy, or military affairs [19]. This results in a limited number of labeled examples, especially in new or critical regions such as combat zones or contaminated areas.

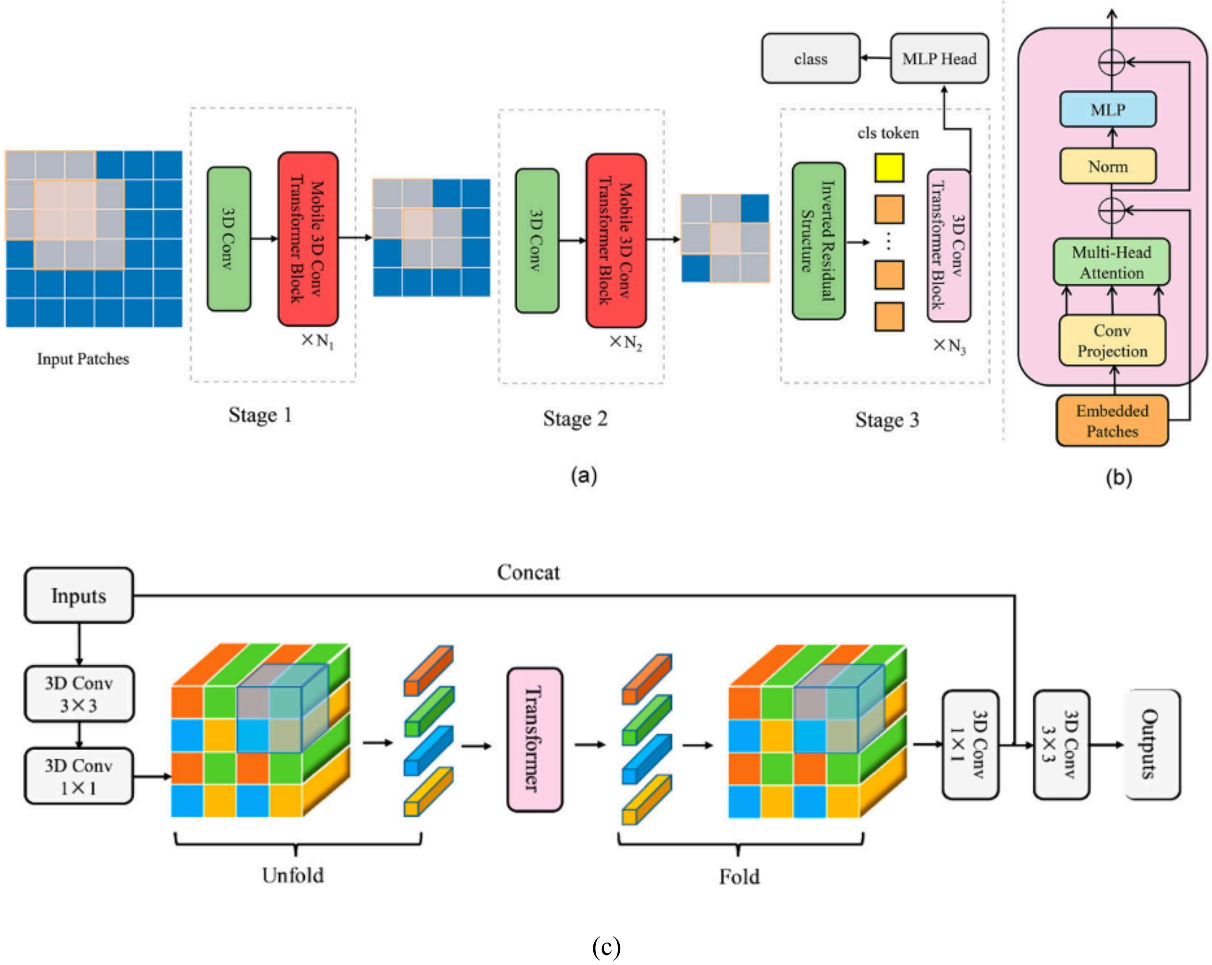

Figure 1.5 – (a) overall topology of the MDvT model; (b) 3D Conv Transformer Block; (c) Mobile 3D Conv Transformer Block [25]

In such conditions, semi-supervised learning (SSL) is a promising direction, allowing both labeled and unlabeled samples to be used during model training. The main idea behind SSL is that, despite the small number of annotated examples, the model can learn the general structure of data distribution using a large amount of unlabeled examples. The main approaches to SSL in HSI are listed below.

1. Self-learning (the model first learns on a small labeled sample, and then uses its predictions on unlabeled data as pseudo-labels for further training) [15].

2. Consistent learning (the idea is that the model should predict the same results for the same example, even if random augmentations are applied to it).
3. Pseudo-labelling (the model receives pseudo-labels on unlabelled data, but includes only those samples in further training for which it is most confident).
4. Graph models (the use of graph neural networks (GNN) in combination with semi-supervised learning allows information to be transferred between pixels that are spatially or spectrally close) [14].

For Ukraine, where monitoring agricultural land, detecting mines, pollution zones, etc. is extremely important, semi-supervised models pave the way for creating effective systems based on a limited amount of verified data. Instead of manually annotating thousands of pixels, you can use a small amount of labeled data combined with a large number of unlabeled images obtained from drones or satellites. This significantly reduces costs and speeds up the process of implementing HSI analysis in practice.

1.4 Problem Statement

To achieve high-performance UAV computer vision and navigation using HSI, the following tasks must be addressed:

1) conduct a review of existing approaches to hyperspectral image classification;
2) design a hybrid model architecture based on MDvT and Cross-Attention Fusion;
3) implement dual-branch processing of spectral and spatial features followed by their fusion;

4) perform training and evaluation of the model on standard hyperspectral datasets;
5) carry out a comparative analysis with classical methods (SVM, 3D-CNN, Vision Transformer);
6) evaluate the model's accuracy, inference time, and number of parameters.

The multi-objective optimization formula, which simultaneously minimizes classification error, inference time, and the number of model parameters, is defined as:

$$min_{\theta}(\lambda_1 \times E(\theta) + \lambda_2 \times \frac{T(\theta)}{T_{ref}} + \lambda_3 \times \frac{P(\theta)}{P_{ref}}) \ , \qquad (1.1)$$

where

θ – vector of model parameters (weights, biases),

$E(\theta) = 1 - A(\theta)$ – classification error,

$T(\theta)$ – average inference time in seconds,

$P(\theta)$ – number of model parameters in millions,

$\lambda_1, \lambda_2, \lambda_3 > 0, \lambda_1 + \lambda_2 + \lambda_3 = 1$ – weight coefficients defining the importance of each criterion,

T_{ref}, P_{ref} – reference values of inference time and parameters of the baseline model.

In this work, the main priority is optimization of inference time and model size, in order to make the architecture applicable on mobile devices under dynamic real-time conditions.

For classification quality evaluation, the following metrics will be used:

- Overall Accuracy (OA):

$$OA = \frac{1}{M}\sum_{k=1}^{M} 1(y_k = \hat{y}_k); \qquad (1.2)$$

- Average Accuracy (AA):

$$AA = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{\lceil M_c \rceil}\sum_{k\epsilon M_c} 1(y_k = \hat{y}_k); \quad (1.3)$$

- Kappa Coefficient:

$$\varkappa = \frac{p_o - p_e}{1 - p_e}; \quad (1.4)$$

- as well as average inference time and number of model parameters.

Conclusion to Section 1

This chapter reviewed the theoretical foundation necessary for developing a high-performance deep learning architecture for HSI classification. The core relevance of HSI lies in its ability to provide fine-grained material and object differentiation, which is crucial for UAV navigation and perception systems operating in complex environments.

Existing HSI deep learning methods were categorized into Convolutional Neural Networks (CNNs) for local feature extraction, Transformers for modeling global dependencies, and Hybrid Networks. The review highlights that hybrid approaches, such as the Mobile 3D Vision Transformer (MDvT), offer the most promising balance between high accuracy and computational efficiency, making them suitable for onboard UAV deployment.

The introduction of semi-supervised learning techniques was identified as a critical necessity to address the inherent challenge of scarce labeled data in the HSI domain.

The problem statement clearly outlines the research objectives: to achieve high-performance HSI computer vision by designing and evaluating a novel hybrid deep learning architecture. The core task is a multi-objective optimization focused on simultaneously minimizing classification error, inference time, and the number of model parameters. The primary goal is to ensure the final architecture is computationally lightweight, enabling its practical deployment on mobile devices and UAV onboard processing for dynamic, real-time operations. Classification quality will be assessed using metrics such as Overall Accuracy, Average Accuracy, and the Kappa Coefficient, in addition to the key efficiency metrics of inference time and parameter count.

CHAPTER 2 INTRODUCING SPECTRAL CROSS-ATTENTION

The theoretical analysis of existing deep learning methods for HSI classification, presented in Chapter 1, highlighted a critical trade-off between classification accuracy and computational efficiency. While 3D CNNs effectively capture local spectral-spatial features, they are often too computationally expensive for resource-constrained platforms. Conversely, ViTs excel at modeling global dependencies but require substantial memory and computational power, limiting their real-time applicability on UAVs. To address these limitations and answer the research question of how to effectively incorporate hyperspectral data into UAV navigation and detection pipelines, this chapter introduces a novel architectural component: the SpectralCA (Spectral Cross-Attention) module. The complete source code implementation of the proposed module is provided in Appendix A.

This chapter details the design and engineering of the SpectralCA module, a hybrid unit constructed to replace standard convolutional, transformer, or hybrid blocks within neural network architectures. The core innovation of the proposed module lies in its bi-directional cross-attention mechanism, which establishes a symmetric interaction between spatial and spectral feature branches. Unlike traditional methods that process these domains separately or sequentially, SpectralCA fuses them dynamically, allowing spatial context to guide spectral feature extraction and vice versa.

The chapter is organized to provide a comprehensive technical description of the proposed solution. Section 2.1 presents the topology of the SpectralCA block, illustrating the dual-branch architecture that combines 2D and 3D convolutions with attention mechanisms. Section 2.2 defines the mathematical model, formalizing the tensor operations, normalization, and attention calculations that govern the block's behavior. Finally, Section 2.3 analyzes the training parameters when incorporating the module into one of the most proficient models for HSI classification, namely MDvT. This analysis demonstrates how the proposed design reduces the overall parameter count and computational complexity compared to the baseline implementation, thereby optimizing the model for deployment on UAV onboard systems.

2.1 SpectralCA Module Topology

The research question addressed in this work is how hyperspectral data can be effectively incorporated into UAV navigation, object detection, material recognition, and terrain classification pipelines. To answer this, the SpectralCA block was designed as a hybrid computational module that synergistically combines the local feature extraction capabilities of 2D and 3D convolutions with the global context modeling of bidirectional cross-attention. This section presents the construction of a modified MDvT architecture, in which the baseline MobileViTBlock is replaced with a new module – SpectralCA (Spectral Cross-Attention) (Fig. 2.1).

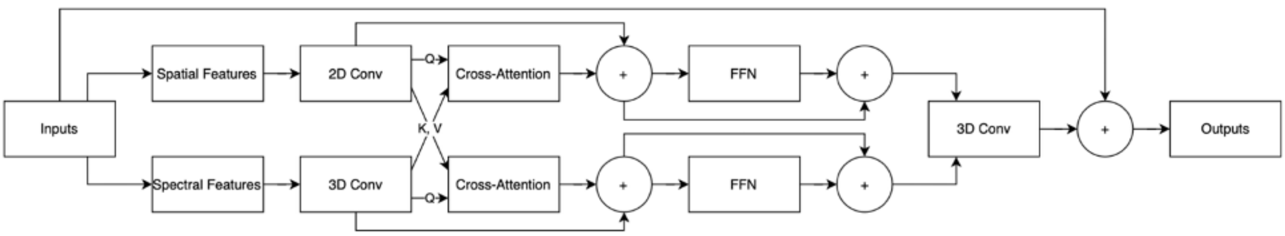

Figure 2.1 – Topology of the SpectralCA block

The purpose of the SpectralCA block is to improve the efficiency of hyperspectral image classification by enhancing the interaction between spatial and spectral features. Unlike standard fusion methods that often process these domains sequentially, the proposed design facilitates a dynamic exchange of information, allowing spatial structures to guide spectral analysis and vice versa. This approach aims to simultaneously reduce the number of parameters and inference time, addressing the critical hardware constraints typical of UAV onboard systems.

2.2 Mathematical Model

The formal mathematical definition of the SpectralCA block ensures reproducibility and clarity regarding the data flow. The input to the module is a 5-dimensional hyperspectral image tensor, highlighted in Figure 2.2 and described in Equation (2.1).

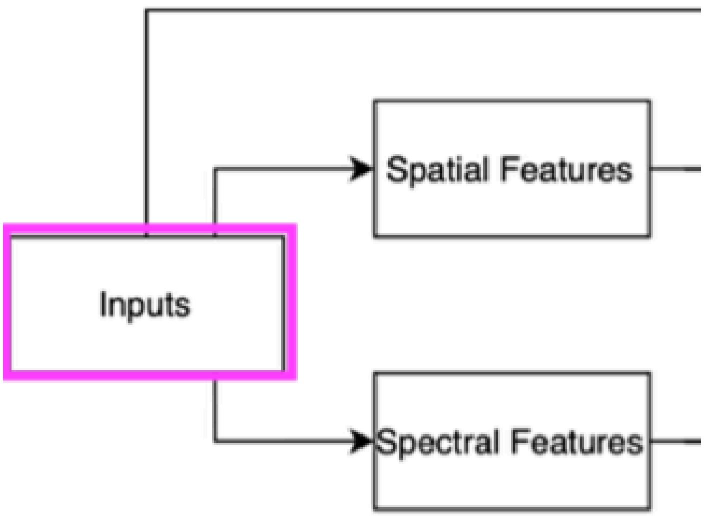

Figure 2.2 – The input to the SpectralCA module

$$X \in R^{\{B \times C \times H \times W \times D\}}, \tag{2.1}$$

where

B – batch size,

C – number of channels,

H×W – spatial coordinates (height and width),

D – spectral depth.

In this study, we employ hyperspectral data from the WHU-Hi-HongHu agricultural dataset, which contains high-precision spectral measurements of farmland [27]. This dataset serves as a rigorous testbed for evaluating the model's ability to distinguish between spectrally similar classes in complex environments.

The SpectralCA architecture processes features through two distinct parallel paths, followed by a cross-attention mechanism.

Spatial Feature Extraction. The first branch focuses on spatial information. By computing the mean value across the spectral axis, the input tensor is reduced to a 2D spatial feature map. It is highlighted in Figure 2.3 and showcased in Equations (2.2) and (2.3).

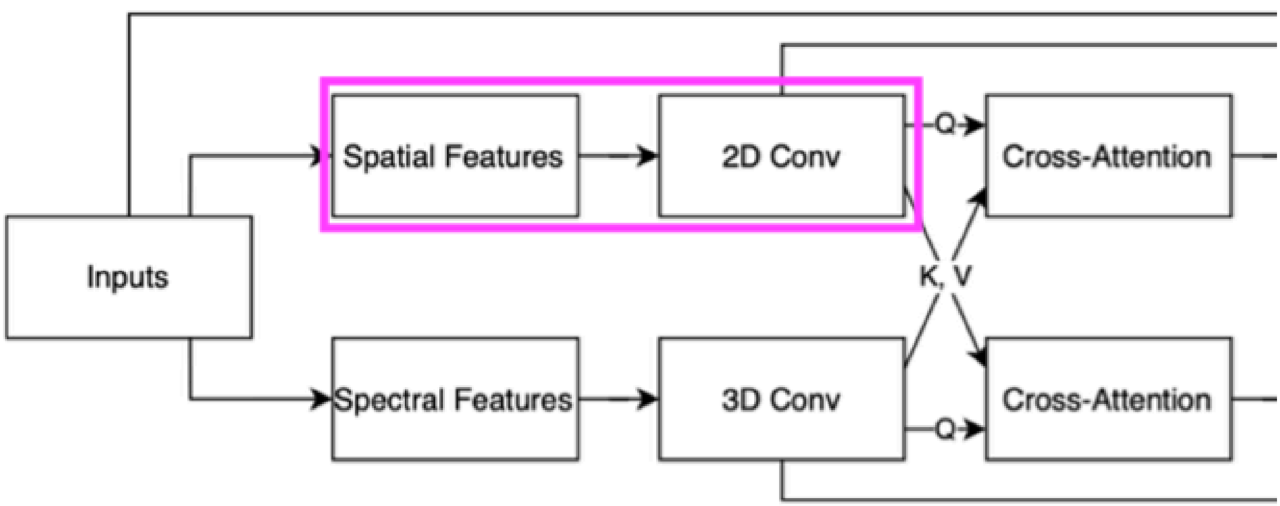

Figure 2.3 – Processing spatial features and applying convolutions

$$X_{spatial} \in \mathbb{R}^{B\times C\times H\times W}, \tag{2.2}$$

$$F_{spatial} = SiLU(BN(Conv2D^{3\times3}(X_{spatial}))), \tag{2.3}$$

where Conv2D detects local textures and spatial patterns, BatchNorm (BN) stabilizes the training process by normalizing layer inputs, and SiLU ensures nonlinear activation without the vanishing gradient problem often associated with deep networks. Following the convolution, Layer Normalization is applied to prepare the features for the attention mechanism, as defined in (2.4).

$$S = LayerNorm(F_{spatial}) \tag{2.4}$$

Spectral Feature Extraction. The second branch preserves the spectral integrity of the data. A 3D convolution with a 3×3×3 kernel is utilized to jointly model spatial–spectral correlations, capturing the interdependencies between adjacent spectral bands and spatial neighborhoods. This process is highlighted in Figure 2.4 and described in (2.5).

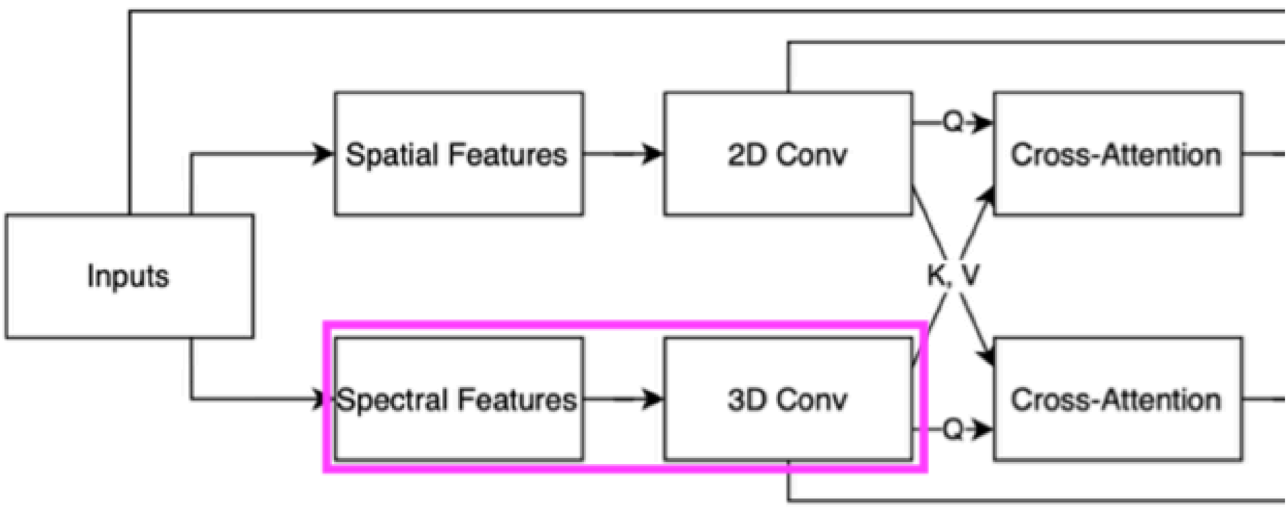

Figure 2.4 – Processing spectral features and applying convolutions

$$F_{spectral} = SiLU(BN(Conv3D^{3\times3\times3}(X))), \tag{2.5}$$

followed by normalization:

$$P = LayerNorm(F_{spectral}). \tag{2.6}$$

Cross-Attention Mechanism. The core of the module is the fusion of these two feature sets. Two attention branches are introduced to enable bi-directional information flow. The process is showcased in Figure 2.5 and described in (2.7) and (2.8).

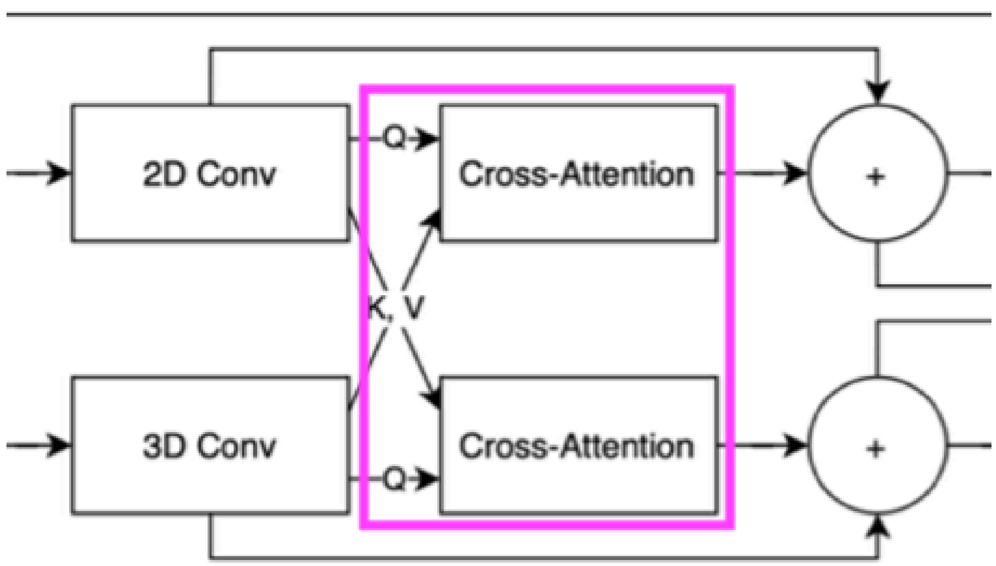

Figure 2.5 – Bi-directional cross-attention between spectral and spatial branches

$$Att_1 = softmax(\frac{Q_s K_p^T}{\sqrt{d_h}}) V_p, \tag{2.7}$$

$$Att_2 = softmax(\frac{Q_p K_s^T}{\sqrt{d_h}}) V_s, \tag{2.8}$$

where

Q, K, V are query, key, and value matrices,

$d_h = \frac{d}{h}$, with d being feature dimensionality and h the number of attention heads.

Att_1 allows the spatial context (from the 2D branch) to attend to the spectral features, effectively strengthening relevant spectral components (e.g., specific bands useful for crop type recognition) based on their spatial location.

Att_2 enables the spectral vectors to attend to the spatial features, allowing the model to adapt spectral signatures to their positional context, thereby improving spatial adaptability.

Following the attention calculation, residual connections, additional normalization, and Feed-Forward Networks (FFNs) are applied in both branches to refine the features and facilitate gradient propagation. This is shown in Figure 2.6 and Equations (2.9) to (2.12).

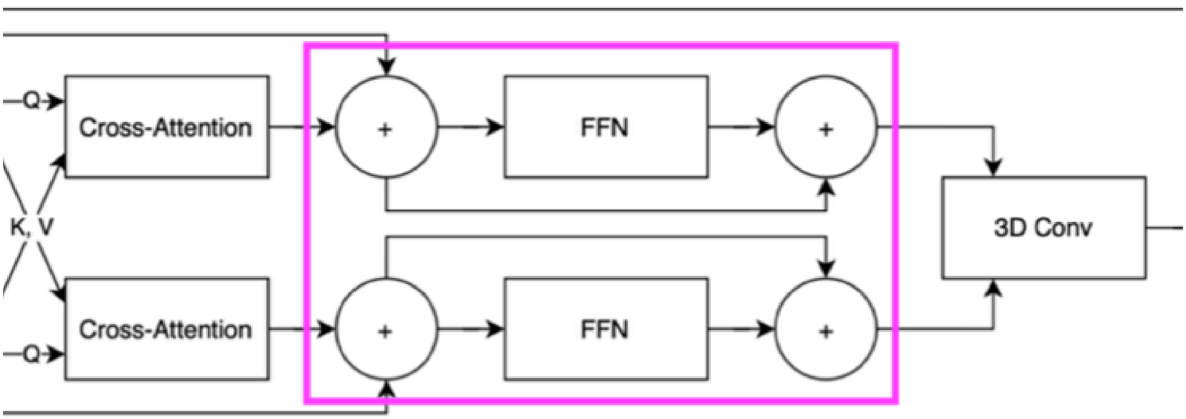

Figure 2.6 – FFN layers with residual connections

$$S' = S + Dropout(Att_1), \tag{2.9}$$

$$S'' = S' + FFN(LayerNorm(S')), \tag{2.10}$$

$$P' = P + Dropout(Att_2), \tag{2.11}$$

$$P'' = P' + FFN(LayerNorm(P')). \tag{2.12}$$

Output Projection. Finally, the refined features from both branches are concatenated and merged. A 1×1×1 3D convolution is applied to project the concatenated features back to the original channel dimensions, followed by a global residual connection to the original input tensor. This is illustrated in Figure 2.7 and Equations (2.13) – (2.14).

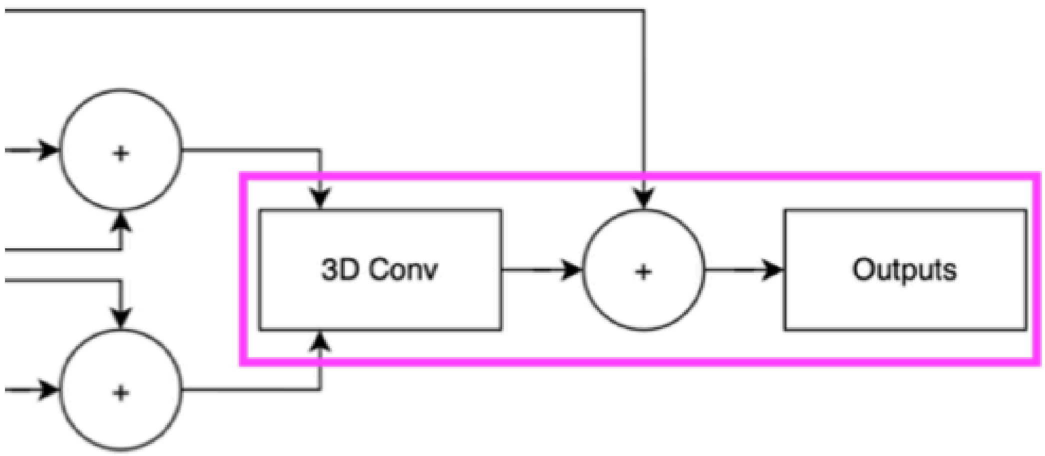

Figure 2.7– Fusion of spectral and spatial branches, followed by a global residual connection

$$Z = Concat(S'', P''), \tag{2.13}$$

$$Y = Conv3D^{1\times1\times1}(Z) + X. \tag{2.14}$$

This residual design ensures the preservation of initial information while enhancing gradient flow and reducing the risk of overfitting during training.

2.3 Training Parameters

The efficiency of the proposed architecture is quantified by analyzing the distribution of training parameters. The breakdown for the MDvT architecture integrating the new SpectralCA block is visually represented in Figure 2.8.

```
==========================================================================================
Layer (type:depth-idx)                                                  Param #
==========================================================================================
MDvT                                                                    256
├─Sequential: 1-1                                                       --
│    └─Conv3d: 2-1                                                      1,792
│    └─BatchNorm3d: 2-2                                                 128
│    └─ReLU: 2-3                                                        --
├─Sequential: 1-2                                                       --
│    └─SpectralCA: 2-4                                                  --
│    │    └─Sequential: 3-1                                             55,584
│    │    └─Sequential: 3-2                                             166,176
│    │    └─CrossAttention: 3-3                                         74,496
│    │    └─LayerNorm: 3-4                                              192
│    │    └─LayerNorm: 3-5                                              192
│    │    └─LayerNorm: 3-6                                              192
│    │    └─LayerNorm: 3-7                                              192
│    │    └─Sequential: 3-8                                             37,152
│    │    └─Sequential: 3-9                                             37,152
│    │    └─Conv3d: 3-10                                                12,352
├─Sequential: 1-3                                                       --
│    └─Conv3d: 2-5                                                      221,312
│    └─BatchNorm3d: 2-6                                                 256
│    └─ReLU: 2-7                                                        --
├─Sequential: 1-4                                                       --
│    └─SpectralCA: 2-8                                                  --
│    │    └─Sequential: 3-11                                            138,600
│    │    └─Sequential: 3-12                                            415,080
│    │    └─CrossAttention: 3-13                                        116,160
│    │    └─LayerNorm: 3-14                                             240
│    │    └─LayerNorm: 3-15                                             240
│    │    └─LayerNorm: 3-16                                             240
│    │    └─LayerNorm: 3-17                                             240
│    │    └─Sequential: 3-18                                            57,960
│    │    └─Sequential: 3-19                                            57,960
│    │    └─Conv3d: 3-20                                                30,848
├─InvertedResidual: 1-5                                                 --
│    └─Sequential: 2-9                                                  --
│    │    └─ConvBNReLU: 3-21                                            99,840
│    │    └─ConvBNReLU: 3-22                                            22,272
│    │    └─Conv3d: 3-23                                                196,608
│    │    └─BatchNorm3d: 3-24                                           512
├─Sequential: 1-6                                                       --
│    └─Rearrange: 2-10                                                  --
│    └─LayerNorm: 2-11                                                  512
├─Sequential: 1-7                                                       --
│    └─Transformer: 2-12                                                --
│    │    └─ModuleList: 3-25                                            4,876,800
├─Dropout: 1-8                                                          --
├─Sequential: 1-9                                                       --
│    └─LayerNorm: 2-13                                                  512
│    └─Linear: 2-14                                                     5,654
==========================================================================================
Total params: 6,627,702
Trainable params: 6,627,702
Non-trainable params: 0
==========================================================================================
```

Figure 2.8 – Distribution of MDvT training parameters with SpectralCA

For the first SpectralCA block (configuration: `in_channels=32, transformer_dim=64`), the module contains 383680 parameters. In an extended configuration for deeper layers (`in_channels=64, transformer_dim=96`), the parameter count increases to 816568. A comprehensive and detailed breakdown of the components is provided in Table 2.1.

Table 2.1 – Parameter distribution of SpectralCA components in the modified MDvT architecture

Component of SpectralCA	*Description*	*Parameter count (in_channels=32, transformer_dim=64)*	*Parameter count (in_channels=64, transformer_dim=96)*
Spatial Conv Block	Conv2D + Batch Normalization + SiLU, 3×3 kernel	55 584	138 600
Spectral Conv Block	Conv3D + Batch Normalization + SiLU, 3×3×3 kernel	166 176	415 080
Cross-Attention (2 directions)	6×Linear + 2×Linear out	74 496	116 160
LayerNorm ×4	Normalization for both paths	768	960
FFN for Spatial features	2×Linear, SiLU, Dropout	37 152	57 960
FFN for Spectral features	2×Linear, SiLU, Dropout	37 152	57 960
Output Projector	Conv3D (1×1×1) for channel restoration	12 352	30 848
Total		383 680	816 568

The SpectralCA block offers significant flexibility; it can be integrated into the MDvT architecture either as a direct replacement for the MobileViTBlock (the primary approach investigated in this work) or as an additional feature extraction component in early layers. Its specific design allows for adaptive weighting of spectral components with respect to the spatial context. Crucially, it maintains the geometric structure of the hyperspectral imagery throughout the processing pipeline, a property that is fundamental for high-precision UAV visual perception tasks such as terrain recognition, obstacle detection, and environmental monitoring.

Conclusion to Section 2

This chapter introduced the proposed SpectralCA module, the core architectural contribution of this work. The topology and design were presented, detailing its structure for the dual-branch processing and efficient fusion of spectral and spatial features. This was formalized through the mathematical model of the bi-directional cross-attention mechanism, which is designed to effectively integrate the complementary information from both domains. Furthermore, the chapter established the specific training parameters and methodologies necessary for implementing and optimizing the new block within the MDvT architecture. The development of SpectralCA directly addresses the efficiency and accuracy objectives, setting the stage for the comparative analysis and application evaluation in the subsequent chapter.

CHAPTER 3 EXPERIMENTAL EVALUATION AND PERFORMANCE ANALYSIS OF SPECTRALCA

The theoretical foundation and mathematical modeling of the Spectral Cross-Attention (SpectralCA) block, presented in the previous chapter, establish a basis for high-performance hyperspectral feature extraction. However, the efficacy of any deep learning architecture must be validated through rigorous empirical testing and comparative analysis. This chapter focuses on the practical implementation, training, and benchmarking of the proposed solution within a comprehensive experimental framework.

The architectural strategy adopted in this research is based on modular modification rather than the development of a neural network from scratch. The Mobile 3D Vision Transformer (MDvT) was selected as the baseline architecture due to its proven efficiency in balancing local 3D convolutional features with global transformer-based dependencies. By integrating the SpectralCA block specifically as a replacement for the standard MobileViTBlock, this study isolates the impact of the proposed bi-directional cross-attention mechanism. This approach allows for a direct assessment of how spectral-spatial fusion contributes to performance gains while maintaining the lightweight characteristics necessary for UAV deployment.

To ensure the reliability of the results, the experimental validation is conducted using the WHU-Hi-HongHu dataset. As detailed in Section 3.1, this dataset was chosen for its high spatial resolution and semantic complexity, which serve as a pre-training foundation and a scientifically valid proxy for the challenges inherent in landmine detection, such as distinguishing small, spectrally similar objects in cluttered environments.

Furthermore, acknowledging the critical constraint of data scarcity in real-world defense and agricultural applications, Section 3.3 investigates the integration of Semi-Supervised Learning (SSL). This section demonstrates how self-training paradigms can leverage unlabeled data to improve model robustness without the need for extensive manual annotation.

The chapter proceeds with a Comparative Analysis (Section 3.2), quantifying the improvements in accuracy, inference speed, and parameter efficiency against the baseline MDvT. Subsequently, Section 3.4 ranks the proposed SpectralCA architecture among other state-of-the-art hyperspectral classification methods to establish its position in the current research landscape. Finally, Section 3.5 provides a discussion of the findings and outlines the roadmap for future work, including transfer learning strategies for domain adaptation to real-world minefields.

Experimental Setup and Hardware. All experimental validations, including supervised and semi-supervised training phases, were conducted using the Google Colab cloud environment. The computational infrastructure relied on a single NVIDIA T4 Tensor Core GPU (16 GB VRAM). To ensure the reproducibility of the experiments and provide access to the full training pipeline, the project is hosted in a public Git repository, the link to which is provided in Appendix B.

3.1 HSI Dataset

For the experimental evaluation of the developed SpectralCA architecture, the WHU-Hi-HongHu dataset was selected. This dataset is part of the WHU-Hi (Wuhan UAV-borne Hyperspectral image) benchmark collection, acquired and shared by the RSIDEA research group of Wuhan University. The data was collected using a hyperspectral sensor mounted on a UAV platform.

The WHU-Hi dataset consists of three distinct hyperspectral scenes: HanChuan (Fig. 3.1), LongKou (Fig. 3.2), and HongHu (Fig. 3.3). All images were acquired in Hubei Province, China, using a Headwall Nano-Hyperspec sensor mounted on a DJI Matrice 600 Pro or Leica Aibot X6.

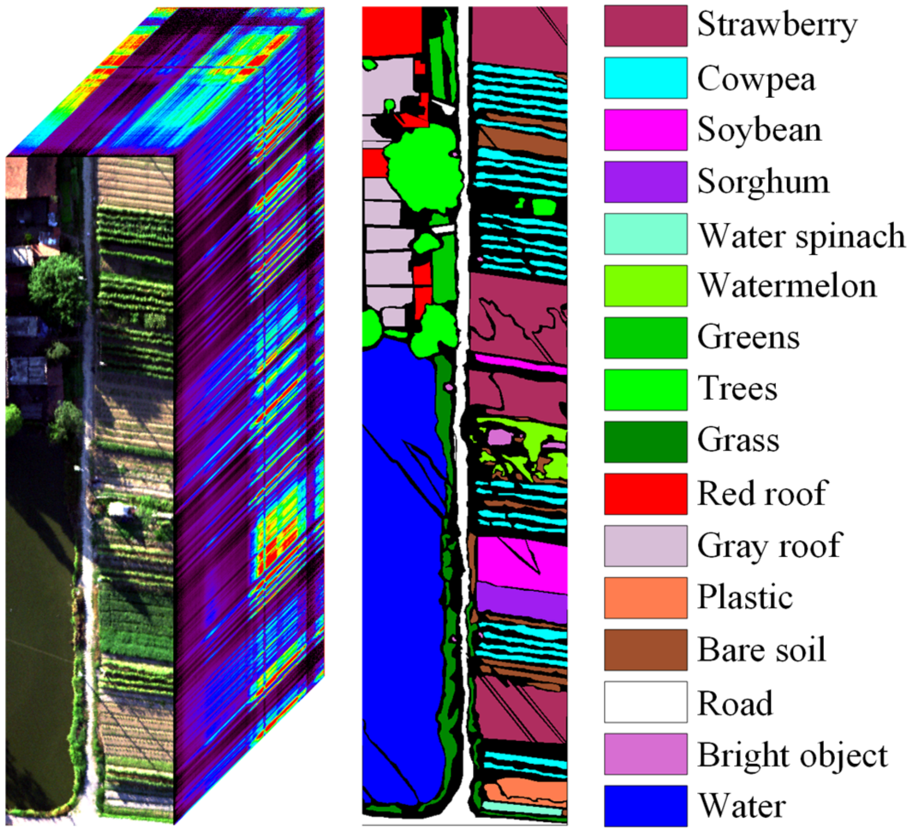

Figure 3.1 – The WHU-Hi-HanChuan dataset

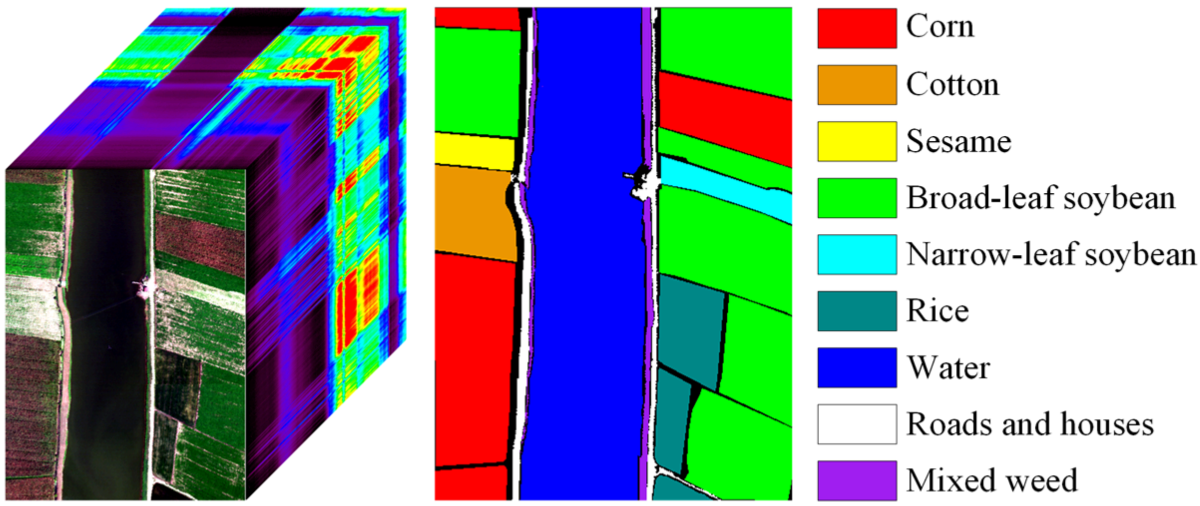

Figure 3.2 – The WHU-Hi-LongKou dataset

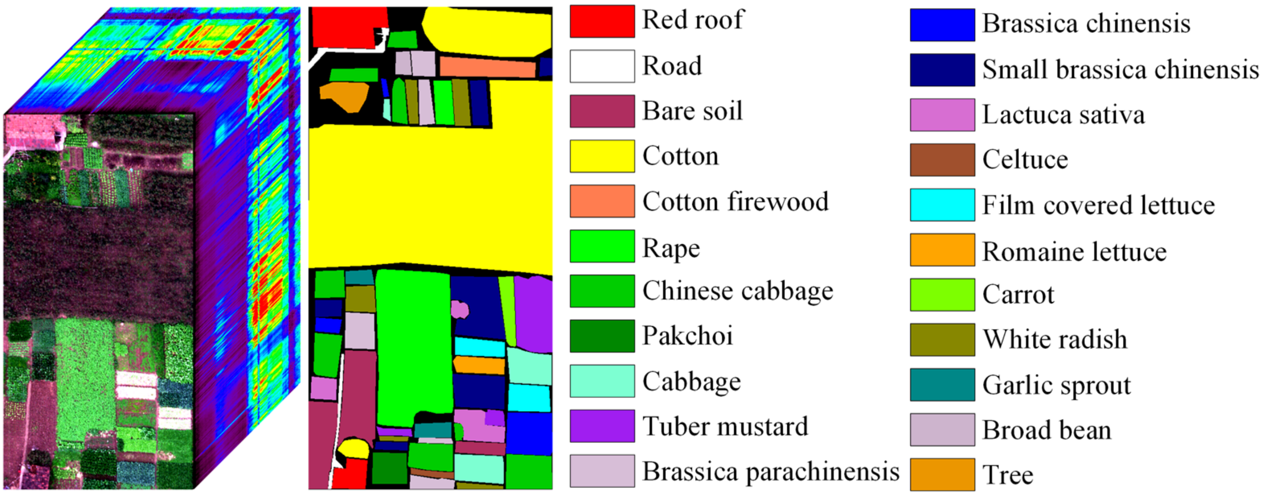

Figure 3.3 – The WHU-Hi-HongHu dataset

A key feature of this data is its high spatial resolution, which significantly exceeds the capabilities of spaceborne (e.g., Hyperion) or traditional airborne platforms. This high resolution enables the use of these images for precision agriculture tasks and the classification of small-scale objects.

The HongHu scene was captured on November 20th, 2017, in Honghu City, between 16:23 and 17:37. The acquisition was performed from an altitude of 100 meters using a sensor with a focal length of 17 mm, mounted on a DJI Matrice 600 Pro UAV. The technical parameters of the dataset are detailed in Table 3.1.

The HongHu dataset contains 22 object classes, making it the most diverse among all WHU-Hi sets. In addition to artificial objects (roads, roofs), it includes a wide range of agricultural crops [27].

A specific challenge of this dataset is the presence of different cultivars of the same crop, which exhibit very similar spectral signatures, creating a high degree of inter-class similarity. For example:

- *Brassica chinensis* vs. *Small Brassica chinensis*;
- *Lactuca sativa* vs. *Celtuce* vs. *Romaine lettuce*.

Table 3.1 – Technical Specifications of the WHU-Hi-HongHu Hyperspectral Cube

Parameter	*Value*
Platform	DJI Matrice 600 Pro
Sensor	Headwall Nano-Hyperspec (17 mm)
Image Size	940×475 pixels
Number of Spectral Bands	270
Spectral Range	400–1000 nm
Spatial Resolution	0.043 m/pixel
Number of Classes	22

The selection of the HongHu scene as the primary testbed for the SpectralCA model is driven by three critical factors.

1. *Highest Spatial Resolution*: HongHu offers a Ground Sample Distance of 0.043 m/pixel, which is significantly more detailed than LongKou (0.463 m/pixel) and HanChuan (0.109 m/pixel). For UAV tasks, particularly the detection of small objects (which serves as a proxy for landmine detection), high detail is a critical requirement. This allows for the verification of the Cross-Attention mechanism's ability to process fine spatial textures. Although spectral signatures differ from those of landmines, the morphological challenge of distinguishing fine-grained instances against a complex background presents a comparable segmentation challenge.
2. *Scene Complexity (Fine-Grained Classification)*: unlike the LongKou scene, described by the authors as a "simple agricultural scene" with 6

crop types, HongHu is a "complex scene" with 22 classes. The presence of spectrally similar subspecies (cultivars) of cabbage and lettuce requires the neural network to extract deep spectral features rather than relying solely on spatial shapes. This provides an ideal stress test for the spectral branch of the SpectralCA architecture.

3. Relevance to Hidden Object Detection: the acquisition conditions for HongHu (cloudy weather, low illumination, object similarity) better simulate real-world complex UAV operating environments compared to the ideal conditions of other datasets. Successful classification under these conditions confirms the robustness of the proposed method.

Thus, the use of WHU-Hi-HongHu allows for a comprehensive evaluation of the proposed architecture's efficiency in terms of both spatial and spectral resolution.

3.2 Comparative Analysis of SpectralCA and MobileViTBlock as Part of MDvT

In the modified MDvT model, the MobileViTBlock was replaced with the proposed SpectralCA block. Training was performed on the WHU-Hi-HongHu dataset, configured for object detection tasks. The training progress is shown in Figures 3.4 and 3.5.

The initial evaluation utilized a standard supervised training approach. As illustrated in the performance metrics, the base model achieves robust classification capability with a Test Accuracy and Overall Accuracy (OA) of 93.35%. The reliability of these predictions is further confirmed by a Kappa Accuracy of 0.9160, indicating strong agreement between predicted and true labels beyond chance. The model converged with a final Test Loss of 0.2256.

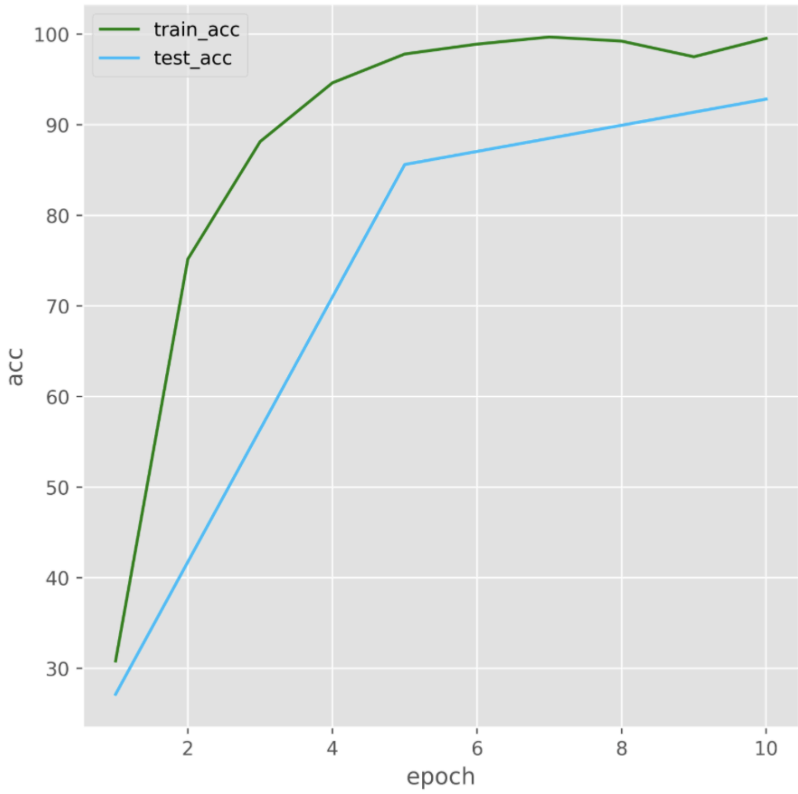

Figure 3.4 – Training results using the SpectralCA block: accuracy

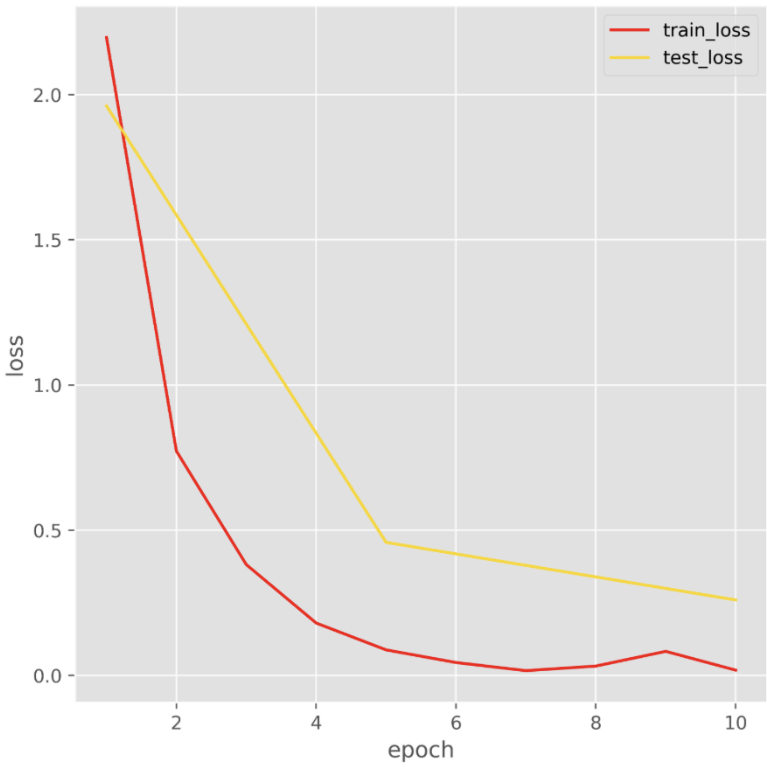

Figure 3.5 – Training results using the SpectralCA block: loss

A granular analysis of per-class performance (Fig. 3.6) reveals variability in the model's ability to distinguish specific spectral signatures.

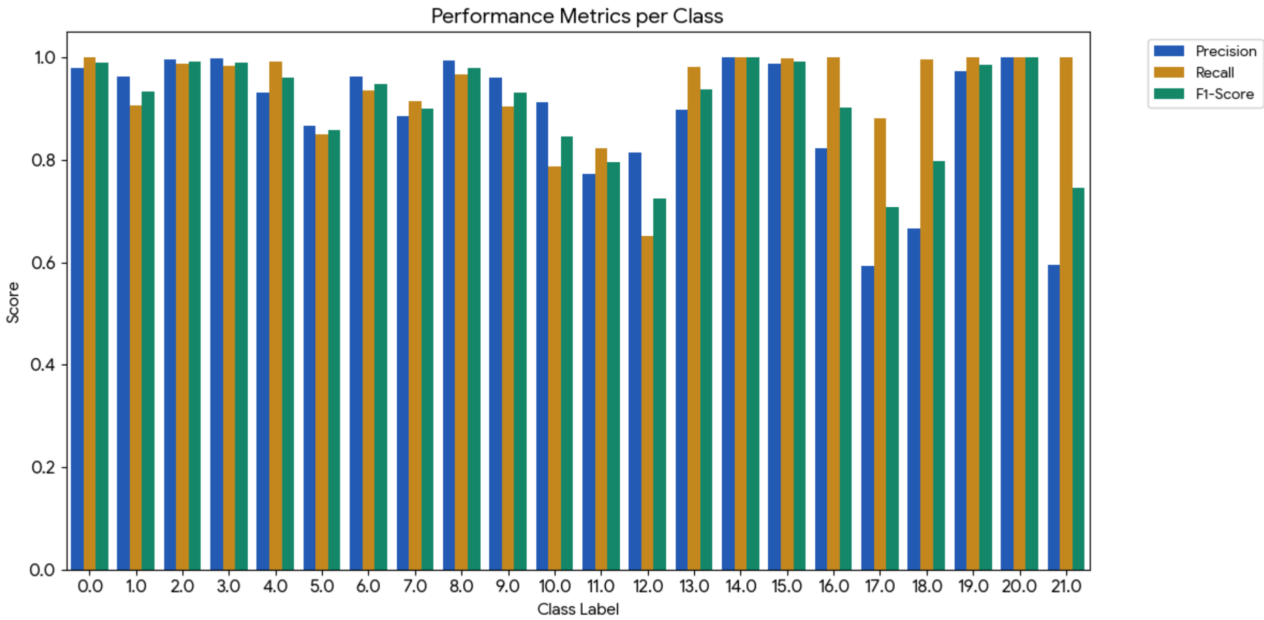

Figure 3.6 – Per-class performance of SpectralCA

The best performing class, *Lactuca sativa* (Class 14), achieved perfect classification with an F1-Score of 1.0. The lowest performing class, *Romaine lettuce* (Class 17), proved to be most challenging, yielding an F1-Score of approximately 0.71. The primary issue for this class was Precision (0.59), suggesting a higher rate of false positives.

The confusion matrix visualizes these results in Figure 3.7. A strong diagonal line indicates that the majority of samples were correctly classified. However, distinct misclassifications are visible as light squares off the diagonal. Specifically, the confusion matrix confirms the difficulties with Class 17, which shows frequent confusion with neighboring spectral classes due to spectral similarity.

The analysis shows that the modified MDvT architecture has approximately 1.1 million fewer parameters compared to the original version (Fig. 3.8).

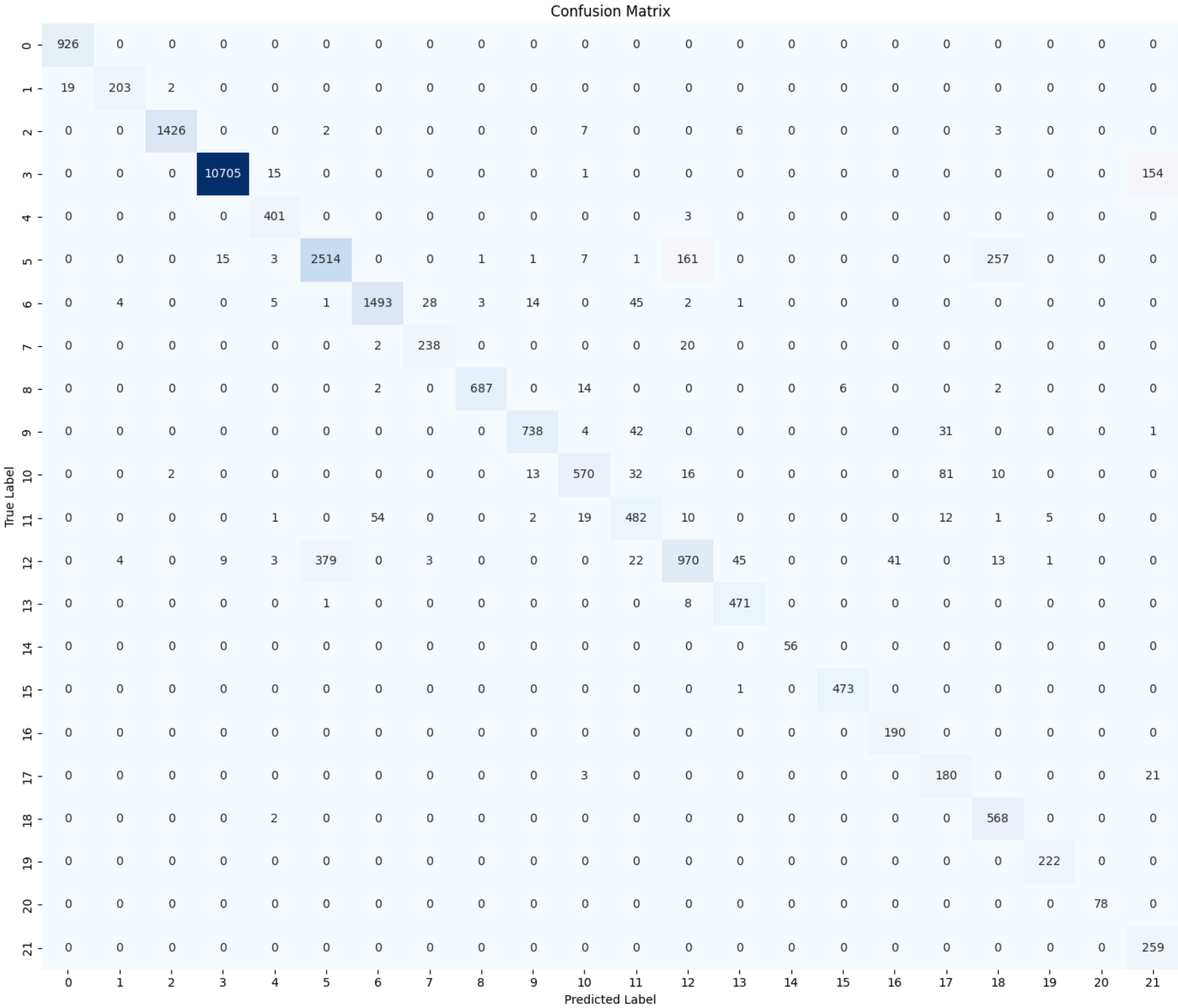

Figure 3.7 – Confusion matrix for the WHU-Hi-HongHu dataset

Experiments demonstrate that the SpectralCA architecture is almost twice as fast as the base MDvT model while maintaining only a ~4% drop in accuracy, reaching approximately 93%. This performance boost is especially important for real-time inference. The comparison is illustrated in Figures 3.9 and 3.10, with a detailed report provided in Table 3.2.

For UAV perception, this translates into more reliable detection of terrain classes, vegetation stress, and artificial structures. Improved perception capabilities directly enhance UAV motion control by reducing navigation errors and enabling safer path planning.

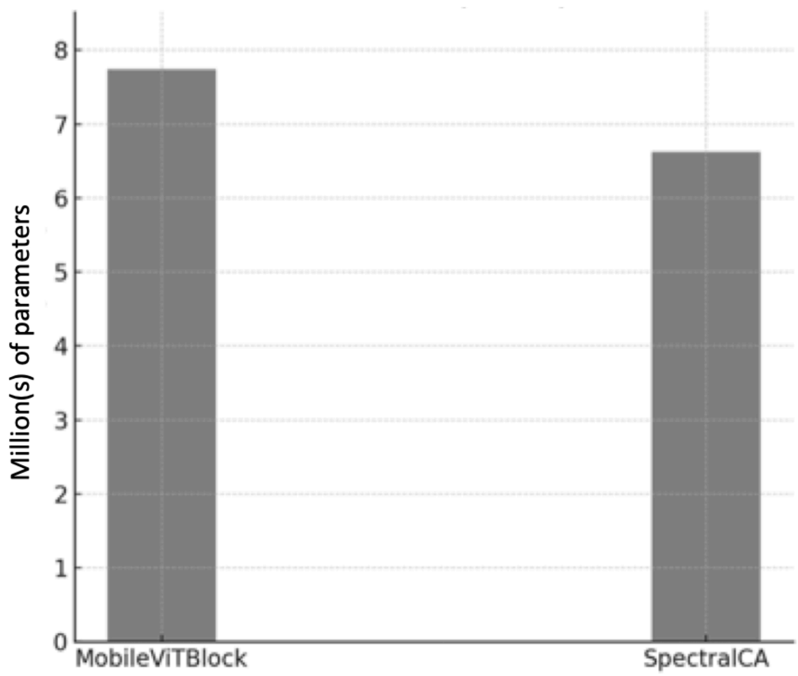

Figure 3.8 – Comparison of the model parameter counts (in millions) between the SpectralCA and MobileViTBlock variants

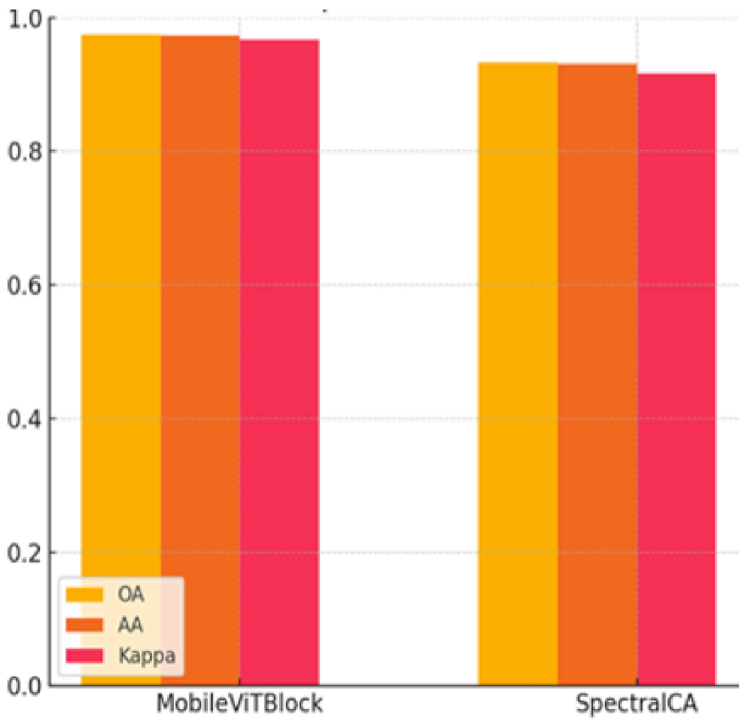

Figure 3.9 – Comparison between the proposed SpectralCA block and the base MobileViTBlock: OA, AA, and Kappa

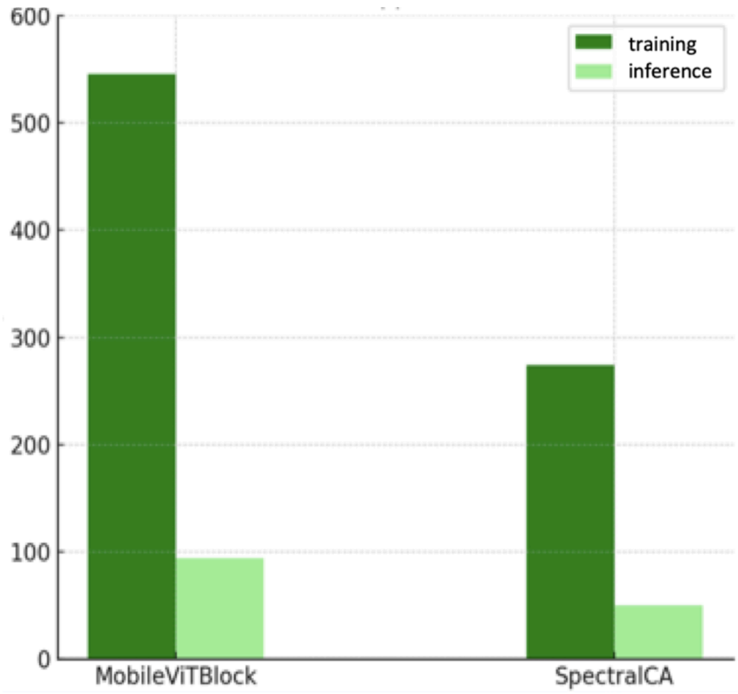

Figure 3.10 – Comparison between the proposed SpectralCA block and the base MobileViTBlock: training and inference times in seconds

Table 3.2 – Detailed report on the performance of the SpectralCA and MobileViTBlock variants

Model	*MobileViTBlock*	*SpectralCA*
OA	**0.974**	0.9334
AA	**0.9739**	0.9308
Kappa	**0.9671**	0.9162
Train time (sec)	545.64	**274.38**
Infer time (sec)	94	**50**
Param count (mil)	7.746	**6.628**

Furthermore, robustness tests show that SpectralCA maintains performance under variable illumination and partial occlusion.

Key advantages of using SpectralCA compared to MobileViTBlock:

- 2x training and inference speed;
- 1.1 million fewer parameters in the final MDvT model;
- accuracy > 93% preserved.

These findings confirm that integrating hyperspectral imaging with bi-directional cross-attention provides a practical pathway to advancing UAV autonomy.

3.3 Semi-Supervised Learning

The task of processing hyperspectral images using AI is often accompanied by a serious limitation – a lack of labeled data. Each pixel of a hyperspectral cube corresponds to a spectral vector, which in many cases requires expert interpretation, especially when it comes to specific fields such as agronomy, geology, or military intelligence. Therefore, labeling hyperspectral data is a labor-intensive and expensive process.

At the same time, modern neural networks, especially transformer and hybrid architectures, have a large number of parameters and require a significant amount of data for effective training. This creates a contradiction between the need for a large number of labeled samples and the practical impossibility of obtaining them in real conditions.

One effective solution to this problem is semi-supervised learning, which allows a small number of annotated samples to be combined with a large amount of unlabeled data.

Within the scope of this work, a self-training approach using proxy labeling was applied. This method assumes that the model is first trained on a limited number of manually labeled pixels and then used to generate pseudo-labels for unlabeled samples. The model's most confident predictions are included in a new training set, after which the model is retrained on the expanded dataset. The procedure is performed iteratively, allowing for a gradual increase in the amount of effectively used data.

In the proposed implementation, a heuristic confidence threshold was used: only those pseudo-labels for which the classification probability exceeded a specified level (namely 90%) were included in further training. This avoids the accumulation of noise in the training set and improves the overall stability of the model.

The training progress is shown in Figures 3.11 and 3.12.

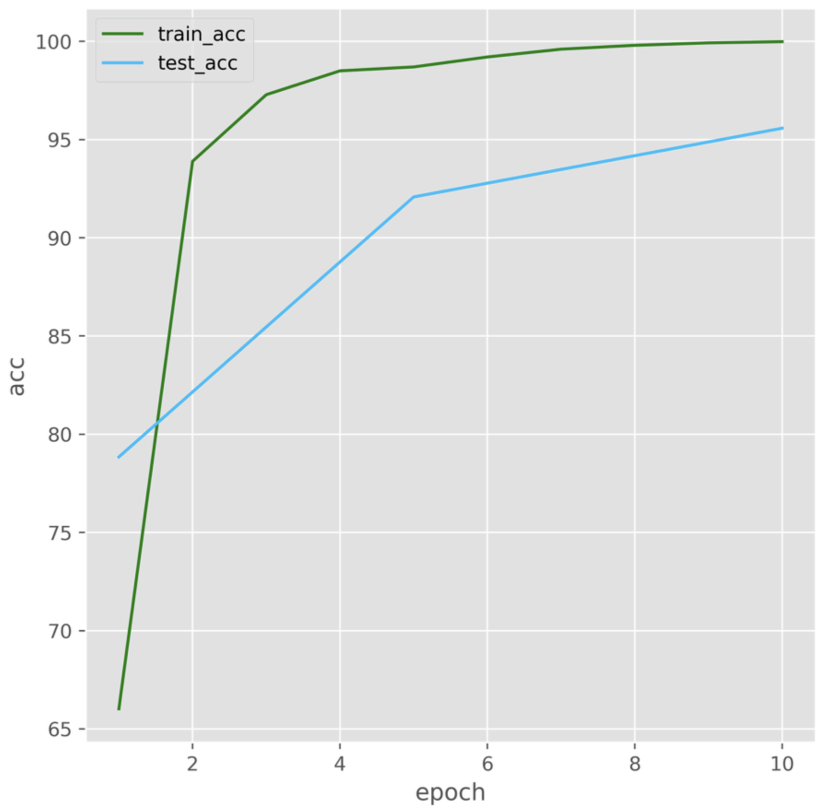

Figure 3.11 – Training results using the SpectralCA block with SSL: accuracy

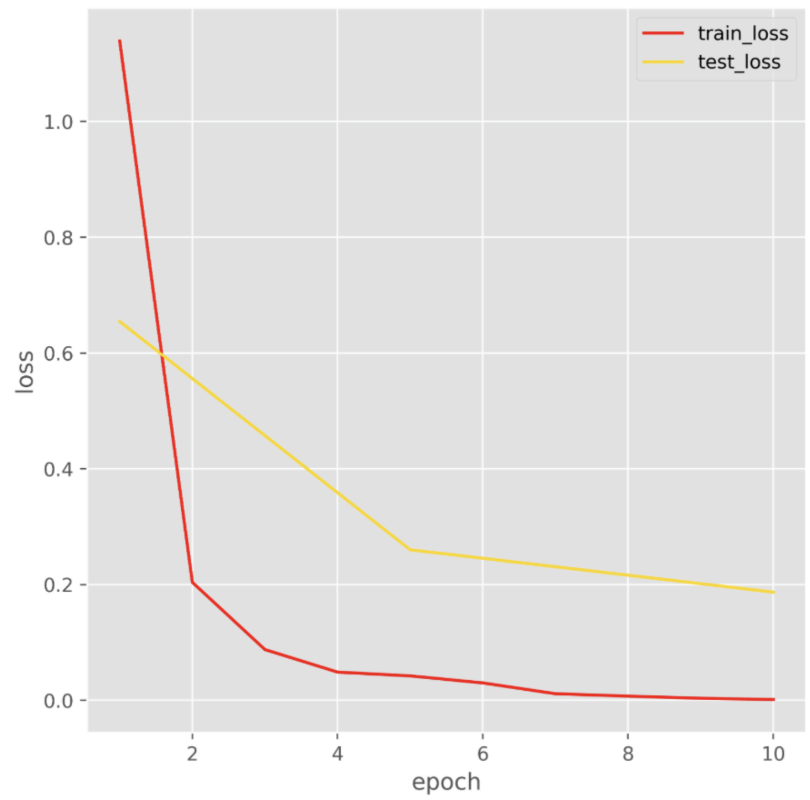

Figure 3.12 – Training results using the SpectralCA block with SSL: loss

Five thousand new images were labeled. The method allowed the model to be adapted to new image fragments that were not included in the initial training set and reduced dependence on fully annotated datasets.

The integration of semi-supervised learning techniques yielded a clear and significant improvement over the baseline performance. Test Accuracy increased to 95.57%, representing a gain of over 2 percentage points compared to the 93.35% of the base model. Kappa Accuracy improved to 0.9440, reflecting superior classification reliability. Test Loss decreased to 0.1906, indicating better generalization and model convergence.

The per-class performance metrics for the SSL model also show broad improvements (Fig. 3.13). While Brassica parachinensis (Class 11) is the most difficult with an F1-Score of 0.7865, this performance is significantly higher than the worst-performing class in the base model (~0.71). This demonstrates that the SSL approach effectively mitigated errors in the most challenging segments of the dataset.

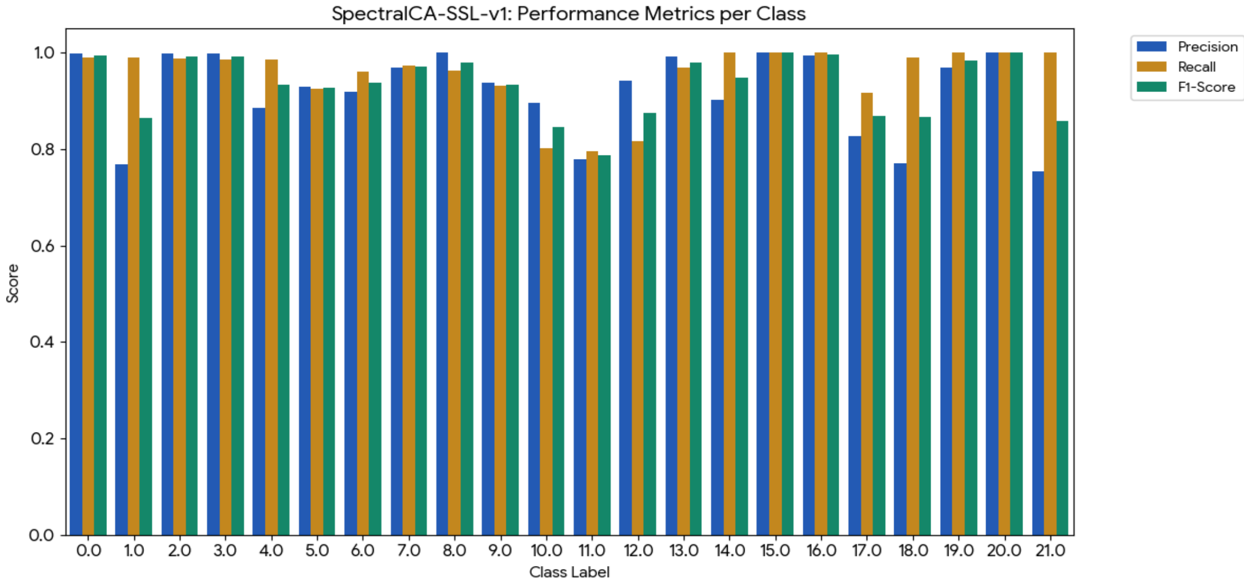

Figure 3.13 – Per-class performance of SpectralCA with SSL

The confusion matrix for the SpectralCA with SSL model displays a noticeably cleaner diagonal compared to the base model (Fig. 3.14). The reduction in off-diagonal errors confirms that misclassifications have been minimized across the board. The SSL approach likely facilitated the learning of more robust feature representations, directly boosting performance on spectrally similar and difficult-to-distinguish classes.

The comparative results of the model trained on a dataset with pseudo-labels are shown in Figures 3.15 and 3.16. The plot in Figure 3.17 shows that after semi-supervised learning, the accuracy of the model increased, approaching the initial version of MDvT. Figure 3.16 demonstrates the stable inference time of MDvT with the SpectralCA block. However, it is evident that although the inference time remained twice as fast as MDvT with MobileViTBlock, the training time increased due to SSL. Table 3.3 contains more detailed characteristics of SpectralCA usage in supervised and semi-supervised training. It shows that the accuracy of the model in SSL increased by ~2%, reaching ~95%.

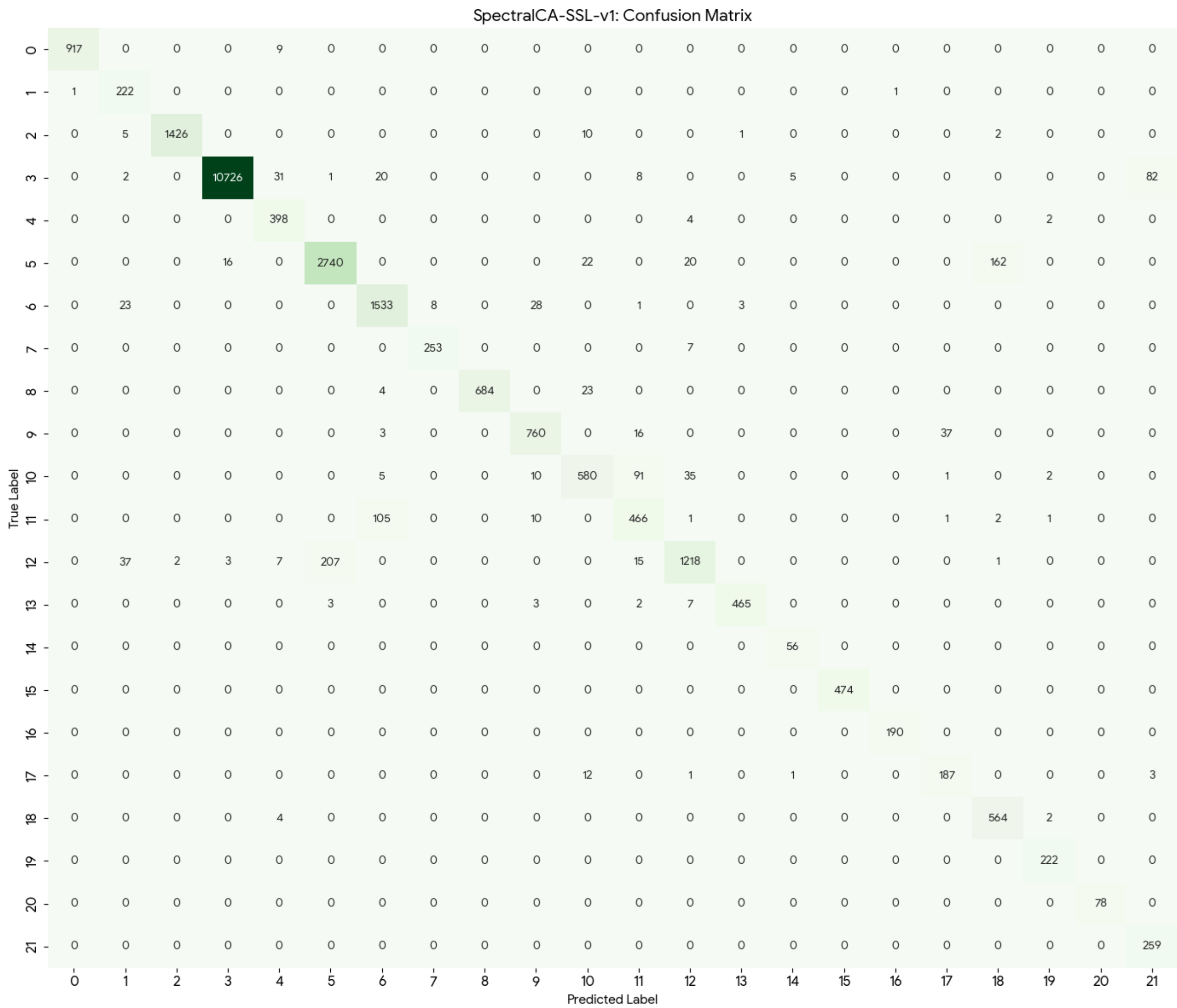

Figure 3.14 – Confusion matrix for the WHU-Hi-HongHu dataset after SSL

The experiments demonstrated that the use of self-training with proxy labeling improves model accuracy. This approach is particularly relevant for practical application in the field, where the number of verified labels is limited and the need for accurate object recognition is critically high.

Table 3.4 shows a detailed comparison of MobileViTBlock and SpectralCA + Self-training. The accuracy of the proposed model is only ~2% lower than the initial architecture with MobileViTBlock, while maintaining twice as fast inference and 1.1 million fewer parameters.

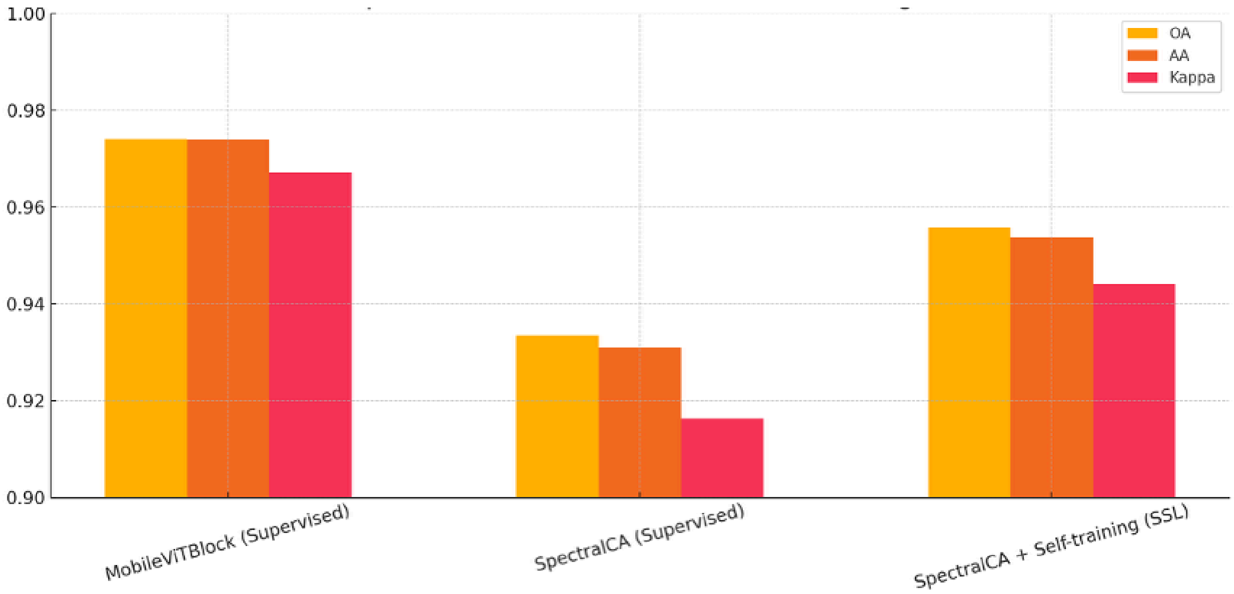

Figure 3.15 – Comparison of model accuracy

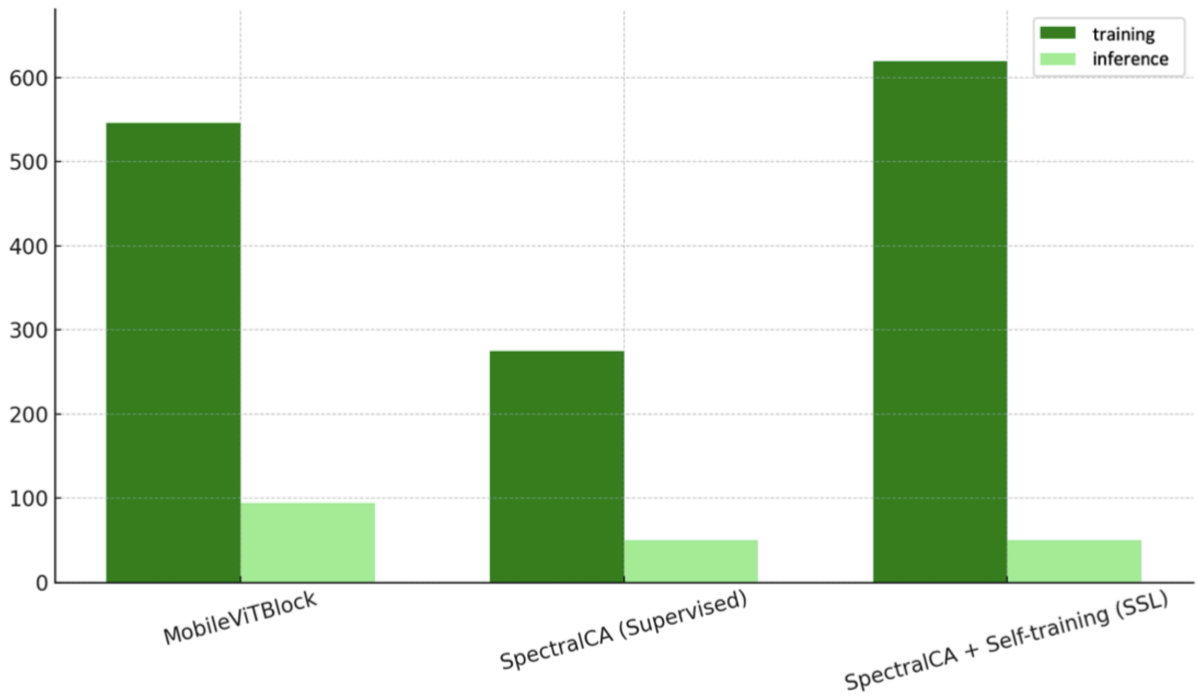

Figure 3.16 – Comparison of model operating times in seconds

Table 3.3 – SpectralCA metrics in supervised and semi-supervised training runs

Model	*SpectralCA (Supervised)*	*SpectralCA (SSL)*
OA	0.9334	**0.9557**
AA	0.9308	**0.9537**
Kappa	0.9162	**0.944**
Train time (sec)	**274.38**	619.05
Infer time (sec)	50	50
Param count (mil)	6.628	6.628

Table 3.4 – Finalized comparison between MobileViTBlock and SpectralCA with self-training

Model	*MobileViT Block*	*SpectralCA (SSL)*
OA	**0.974**	0.9557
AA	**0.9739**	0.9537
Kappa	**0.9671**	0.944
Train time (sec)	**545.64**	619.05
Infer time (sec)	94	**50**
Param count (mil)	7.746	**6.628**

Thus, integrating semi-supervised learning with the SpectralCA architecture is a promising direction for developing hyperspectral classification systems for real-world tasks.

3.4 SpectralCA Ranking Among Other Advanced Hyperspectral Classification Methods

Standard benchmarks for the WHU-Hi-HongHu dataset typically report results on training splits ranging from 25 to 100 samples per class [27]. However, deep learning architectures incorporating attention mechanisms, such as the proposed SpectralCA, often require a higher density of training examples to stabilize the learning of attention weights and prevent overfitting on complex features.

Therefore, this study adopts a 150-sample per class configuration. While this represents a larger labeled dataset than the standard 100-sample benchmark, it ensures that the comparative analysis reflects the architecture's true capacity rather than its stability under data starvation.

Table 3.5 presents a comparative ranking of the proposed SpectralCA architecture (utilizing semi-supervised learning) against established benchmarks and advanced deep learning methods on the WHU-Hi-HongHu dataset. Metrics for other models trained on 100 samples per class are cited from [27]. The results indicate that SpectralCA achieves a highly competitive Overall Accuracy of 95.57%, securing the second-highest position among the evaluated methods. The proposed model demonstrates a substantial performance advantage over traditional machine learning approaches, outperforming SVM (73.55%) by over 22 percentage points, and significantly surpassing earlier deep learning frameworks such as R-PCA CNN (85.43%) and SSRN (91.29%). Furthermore, SpectralCA outperforms complex state-of-the-art architectures, including CNN-CRF (93.74%) and PresNet (95.32%). While it trails slightly behind the top-performing FPGA framework (97.45%), the marginal difference of approximately 1.9% is offset by the architectural efficiency and reduced parameter count of SpectralCA, as detailed in Chapter 2. This positioning confirms that the proposed solution effectively balances near-SOTA accuracy with the computational constraints required for practical UAV deployment.

Table 3.5 – Comparison of SpectralCA with other hyperspectral classification methods

Model	*Overall Accuracy*
SVM	73.55
FNEA-OO	88.83
SVRFMC	89.86
R-PCA CNN	85.43
SAE-LR	78.68
SSAN	87.34
SSRN	91.29
PresNet	**95.32**
SSFCN-CRF	94.26
FPGA	**97.45**
CNNCRF	93.74
SpectralCA	**95.57**

3.5 Discussion and Future Work

The experimental results presented in this chapter confirm the validity of the SpectralCA architecture as a robust and efficient solution for hyperspectral image classification. The comparative analysis demonstrated that the proposed model achieves an optimal balance between computational efficiency and classification accuracy. While the Overall Accuracy of 95.57% (using SSL) is marginally lower than the top-performing FPGA framework (97.45%), the SpectralCA model offers a distinct advantage in terms of parameter efficiency and inference speed, which are the primary constraints for the target application of UAV-based landmine detection.

Furthermore, the successful application of SSL addressed one of the most critical challenges in the domain: the scarcity of labeled data. The ability to improve model performance by leveraging unlabeled samples validates the practical viability of this approach for real-world scenarios where manual annotation is costly or dangerous.

However, the transition from agricultural proxy data to real-world demining operations requires further research and optimization. Based on the findings of this study, the following directions for future work have been identified.

1. *Hyperparameter Optimization via Grid Search*. While the current configuration of the SpectralCA block yielded high performance, the hyperparameters were selected based on heuristic trials. Future work will involve conducting a systematic Grid Search to identify the optimal combination of learning rates, attention head counts, and patch sizes. This process aims to squeeze the maximum possible accuracy out of the architecture without increasing its computational footprint.
2. *Transfer Learning Strategy*. Pre-training and fine-tuning to bridge the domain gap between agricultural data and landmine detection, a Transfer Learning strategy will be implemented:
   - *pre-training*: the model will first be pre-trained on large-scale, general-purpose hyperspectral datasets (e.g., Pavias, Salinas, or aggregated satellite imagery); this will allow the feature extractors to learn robust, generalized spectral representations and spatial textures;
   - *fine-tuning on target data*: the pre-trained weights will then be fine-tuned on specific, smaller datasets containing spectral signatures of landmines (both plastic and metal) in various soil conditions; this two-step approach is expected to significantly

accelerate convergence and improve detection rates on the target objects.

3. *Advanced Semi-Supervised Learning on Target Domains*. Real-world demining datasets typically contain a vast amount of background data (soil, vegetation) and very few labeled mine targets. Therefore, the Semi-Supervised Learning pipeline validated in Section 3.3 will be reiterated on the target landmine datasets. By utilizing the abundant unlabeled data from UAV surveys, the model can further refine its decision boundaries, ensuring robust detection of concealed objects even with limited ground truth annotations.

Conclusion to Section 3

This chapter presented the practical implementation and empirical evaluation of the SpectralCA architecture within the MDvT framework. The experimental design was grounded in the selection of the WHU-Hi-HongHu dataset, which was established in Section 3.1 as a pre-training foundation and a scientifically valid proxy for landmine detection due to its high spatial resolution and fine-grained spectral complexity.

The Comparative Analysis (Section 3.2) quantitatively confirmed the efficiency of the proposed solution. The SpectralCA-based model demonstrated a 2x increase in inference speed and a reduction of 1.1 million parameters compared to the baseline MobileViTBlock, while maintaining high classification accuracy. These metrics validate the architecture’s suitability for resource-constrained UAV onboard systems.

Furthermore, Section 3.3 successfully demonstrated the efficacy of SSL. By leveraging proxy labeling on unlabeled data, the model achieved an Overall Accuracy of 95.57%, bridging the gap between data-scarce environments and high-performance requirements.

The benchmarking results in Section 3.4 positioned SpectralCA as a top-tier classification method, outperforming established deep learning frameworks like CNN-CRF and PresNet, and competing closely with state-of-the-art global learning frameworks.

Finally, Section 3.5 outlined a clear roadmap for bridging the domain gap to real-world operations, identifying hyperparameter optimization and transfer learning as key next steps.

Collectively, these findings confirm that the SpectralCA module is not only a theoretical advancement but a robust, deployable solution for autonomous hyperspectral perception.

CHAPTER 4 STARTUP IMPLEMENTATION

The implementation of the developed technology is realized through the establishment of a full-fledged startup entity named SpectralCA. The company is structured as a dual-vector enterprise: the core business focuses on developing proprietary software solutions for Hyperspectral Image (HSI) analysis using the neural network architecture described in previous chapters. Simultaneously, a dedicated hardware department manages the integration of Unmanned Aerial Vehicles (UAVs) with hyperspectral sensors to execute field operations.

To ensure rapid growth and minimize initial capital expenditure, SpectralCA operates a specialized business development department. This unit focuses on securing technical equipment on a pro-bono basis by leveraging the humanitarian significance of the project. The primary mission is humanitarian demining, a critical task for global security. By positioning the technology as a "Tech for Good" solution, the company aims to secure hardware (UAVs and sensors) from manufacturers and partners as charitable technical assistance, which significantly optimizes the tax burden during the early stages.

4.1 Problem Description

The war in Ukraine has resulted in the contamination of vast territories with explosive ordnance, making it the most heavily mined country in the world. According to reports from the Ministry of Economy of Ukraine, approximately 30% of the country’s territory is potentially dangerous [1]. The scale of this problem highlights critical deficiencies in existing demining methodologies.

Current demining operations primarily rely on manual teams using metal detectors and prodders. This process is:

1) extremely slow – a sapper can clear only a few square meters per day;
2) dangerous – human operators are in direct proximity to the hazard;
3) inefficient – metal detectors generate a high rate of false positives due to soil mineralization and shrapnel, and they often fail to detect modern plastic mines (e.g., PFM-1 "Butterfly").

While industrial hyperspectral solutions exist, they are designed for large manned aircraft or satellites. According to distributor price lists, such systems often exceed $50,000 [32], lack the spatial resolution to detect small anti-personnel mines, and do not offer real-time processing capabilities. There is a significant gap in the market for a lightweight, AI-driven solution capable of identifying both metallic and plastic mines remotely using low-altitude UAVs.

4.2 Market Analysis

To assess the viability of SpectralCA, an analysis of the demining and remote sensing market was conducted. The market currently consists of traditional manual demining operators, providers of heavy mechanical demining machines, and emerging sensor tech companies.

The potential market is divided into three key segments:

1) defense and humanitarian (B2G) – the State Emergency Service (SES), NGOs (HALO Trust), and Private Military Contractors requiring high-reliability detection of plastic mines;
2) precision agriculture (B2B) – agro-holdings requiring early detection of pests and water stress;
3) environmental monitoring – ecological services monitoring soil contamination.

A comparative analysis of the solutions is given in Table 4.1. It covers the approach to detection, cost efficiency, safety, and scalability.

Table 4.1 – Comparative analysis of demining and scanning solutions

Criteria	*Manual Demining*	*Industrial HSI (Satellite/Manned)*	*SpectralCA (This Project)*
Technology	Metal detectors, prodders, dogs	Heavy hyperspectral cameras, post-processing on ground	AI-driven Hyperspectral Analysis on UAVs
Safety	Low (Direct human contact)	High (Remote)	High (Remote)
Detection Capability	Detects metal; struggles with plastic	Broad spectral analysis; low spatial resolution	High resolution; detects plastic & metal via spectral signature
Speed	Very slow ($<10 m^2$/day)	Fast scanning, slow processing	Fast scanning + Real-time AI inference
Cost	High operational costs (salaries, insurance)	Very High (Equipment >$50k + flight costs)	Low to Medium (Pro-bono hardware + Low-cost AI inference)
Accessibility	Limited by trained personnel	Limited by budget and airspace control	High (Portable, deployable by small teams)

To provide a comprehensive evaluation of the project's strategic position, a SWOT analysis was conducted. This method identifies internal Strengths and Weaknesses, as well as external Opportunities and Threats (Table 4.2).

The analysis indicates that SpectralCA's primary strategy should be "Niche Domination". By focusing on the specific "AI for Humanitarian Demining" niche, the company leverages its strengths (accuracy, cost) to secure grants while mitigating the threat of large competitors who view this niche as too small or risky.

Table 4.2 – SWOT Matrix of the SpectralCA Project

Strengths (Internal)	*Weaknesses (Internal)*
1. Technological Superiority: The developed SpectralCA architecture achieves 95.57% accuracy, outperforming standard CNN models by ~2%. 2. Cost Efficiency: The software is optimized for mid-range GPUs (NVIDIA T4), eliminating the need for industrial supercomputers. 3. Dual-Use Capability: The underlying technology is applicable to both humanitarian demining and commercial agriculture, diversifying revenue streams. 4. Tax Advantages: Operating as a Private Entrepreneur of Group 3 with pro-bono hardware acquisition significantly reduces the burn rate.	1. Hardware Dependency: The business model relies on securing third-party sensors and UAVs, creating a supply chain risk. 2. Lack of Certification: As a new entrant, the company lacks the ISO/IMAS certification required for independent demining operations. 3. Data Volume: While the company possesses initial samples of minefield data, creating a massive, annotated dataset covers huge operational costs. Current real-world data volumes are sufficient for testing but limited for full-scale model training. 4. Team Size: A small core team (4 people) limits the ability to handle multiple large contracts simultaneously.

End of Table 4.2

Opportunities (External)	*Threats (External)*
1. Reconstruction of Ukraine: The government estimates a market potential of $37 billion for demining operations over the next decade. 2. Grant Ecosystem: Access to defense-tech accelerators (Brave1, NATO DIANA) provides non-dilutive R&D funding. 3. Global Food Crisis: Increasing demand for precision agriculture tools to maximize crop yields in unstable climates. 4. Partnership Potential: High willingness of hardware manufacturers (Headwall, DJI) to test equipment in real combat zones for marketing purposes.	1. Regulatory Bottlenecks: Delays in obtaining flight permits or demining licenses from the State Emergency Service. 2. Fiscal Vulnerability: High sensitivity to potential changes in taxation legislation for Private Entrepreneurs. 3. Currency Devaluation: Risk of Hryvnia devaluation affecting local purchasing power. 4. Geopolitical Instability: Changes in grant funding priorities if the conflict freezes or de-escalates rapidly.

4.3 Solution

The SpectralCA Ecosystem is designed not merely as a piece of software, but as a holistic hardware-software complex capable of autonomous operation in hazardous environments. The solution consists of three integrated modules: the Aerial Acquisition Unit (Hardware), the AI Processing Core (Software), and the Mission Control Interface (UI).

The hardware vector operates as a separate department within the company, responsible for the maintenance, integration, and deployment of UAVs. Given the humanitarian focus, the equipment strategy relies on retrofitting commercial platforms with specialized sensors obtained through partnership agreements.

Technical Specifications of the Integrated Unit:

- UAV Platform: Commercial heavy-lift quadcopter (e.g., DJI Matrice 300 RTK). According to official specifications, this platform offers a flight time of 55 minutes and a payload capacity of 2.7 kg [33], making it suitable for carrying hyperspectral loads:
  - Navigation: RTK-GNSS for centimeter-level positioning accuracy;
- Hyperspectral Sensor: Push-broom or snapshot spectral camera (e.g., Headwall Nano-Hyperspec):
  - Spectral Range: VNIR (400–1000 nm);
  - Bands: 270 spectral channels [32];
  - Data Rate: High-speed onboard storage (NVMe SSD);
- Onboard Compute: NVIDIA Jetson Orin NX for edge pre-processing (optional, for real-time downlinking).

AI Processing Core (Software Vector). This is the intellectual property (IP) backbone of the startup. The software is built upon the SpectralCA neural network architecture developed in Chapter 2.

- Input: Raw hyperspectral data cubes (ENVI/TIFF format);
- Processing Pipeline:
  1) Radiometric calibration (Dark/White reference correction);
  2) Geometric correction (Georeferencing);
  3) Spectral-Spatial feature extraction via Bi-Directional Cross-Attention;
  4) Classification mask generation;
- Output: A geo-referenced "Hazard Map" where red zones indicate high probability of explosive ordnance presence (P>90%).

Operational Interface. The end-product delivered to the client (SES or NGOs) is accessed via a secure web dashboard.

- Mission Planning: Operators define the flight path polygon;
- Live Telemetry: Monitoring battery and sensor status;
- Result Visualization: Interactive 2D/3D map layers overlaying the mine probability heatmaps onto standard satellite imagery.

Computational Infrastructure Selection. A critical technical and financial decision for the startup was selecting the optimal computational infrastructure for training and inference. The choice involved a trade-off between Cloud Hosting (OpEx) and On-Premise Hardware (CapEx).

Analysis of Hosting Providers. For the deployment of the web dashboard and light inference tasks, a comparative analysis of hosting providers was conducted to minimize monthly recurring costs (Table 4.3).

Cloud hosting (AWS/GCP) becomes prohibitively expensive for continuous R&D training, costing over $3,000 annually per instance. Therefore, a decision was made to:

1) purchase a dedicated Deep Learning Workstation for heavy model training (R&D), ensuring data privacy and zero marginal cost per experiment;
2) use Google Colab Pro+ ($50/month) for rapid prototyping and sharing notebooks;
3) use Low-Cost Web Hosting (Hostinger) ($5/month) for the client-facing dashboard, which does not require GPU power (results are pre-rendered).

Justification for Hardware Selection (NVIDIA Ecosystem). The decision to base the infrastructure specifically on NVIDIA hardware was driven by the software architecture requirements (Table 4.4).

Table 4.3 – Comparative Analysis of Hosting and Compute Providers [34]

Provider	*AWS (EC2 G4dn)*	*Google Cloud (Vertex AI)*	*Hetzner (Dedicated GPU)*	*SpectralCA Choice: Hybrid*
Model	Pay-as-you-go	Pay-as-you-go	Monthly Rental	On-Premise + Colab
GPU Spec	NVIDIA T4	NVIDIA T4	GTX 1080 / RTX 3060	RTX 3090 / 4090
Cost per Hour	~$0.52/hour	~$0.40/hour	N/A (Monthly flat rate)	$0 (after purchase)
Monthly Cost (Full load)	~$374	~$288	~$180	~$20 (Electricity)
Scalability	High	High	Low	Medium
Data Privacy	High	High	Medium	Maximum

Investing $3,500 upfront in a custom workstation with an NVIDIA RTX 3090/4090 yields a Return on Investment (ROI) within 9 months compared to renting equivalent AWS instances. Furthermore, owning the hardware ensures that sensitive demining data never leaves the company's local, air-gapped network, satisfying security requirements for defense contracts.

Table 4.4 – Rationale for Purchasing NVIDIA RTX Workstation [35]

Criteria	*NVIDIA RTX 3090/4090 (Chosen)*	*AMD Radeon VII*	*Apple M2 Ultra*	*Google TPU (Cloud)*
Framework Support	Native PyTorch/CUDA support. Industry standard for Transformers.	ROCm support is experimental and unstable for 3D convolutions.	MPS support is limited; slower training for large batches.	Excellent for TensorFlow, harder to debug PyTorch custom layers.
VRAM Capacity	24 GB GDDR6X. Sufficient for 3D Hyperspectral Cubes.	16 GB HBM2.	Unified Memory (up to 128GB) but slower bandwidth.	Variable.
Cost Efficiency	High. Consumer-grade card with near-server performance.	Medium.	Low (High hardware cost).	Low (Recurring rental cost).
Ecosystem	TensorRT optimization allows easy deployment to Jetson edge devices.	No direct edge deployment path.	No edge deployment path.	Cloud-only.

4.4 Business Model

The business model of SpectralCA is structured to ensure sustainability during the R&D phase while preparing for rapid scaling in the commercial phase.

Legal and Organizational Structure To minimize administrative overhead and tax burden during the startup phase, the entity is registered as a Private Entrepreneur of the 3rd Group in Ukraine. This structure is optimal for IT and service-based businesses.

1. Taxation regime – simplified tax system paying 5% of total revenue + 1% Military Levy.
2. Social contribution – fixed quarterly payment (currently 1,760 UAH/month) [36].

The business model acknowledges the volatility of the current fiscal landscape. Ukrainian national revenue strategy outlines a roadmap for reforming the simplified tax system, which presents a significant risk of increased fiscal pressure. Specifically, the potential introduction of mandatory Value Added Tax (VAT) for Group 3 entrepreneurs would fundamentally change the administrative requirements. To mitigate this risk, the operational budget includes a contingency for transitioning to the general taxation system, specifically allocating funds for professional accounting services to ensure compliance with complex VAT reporting.

Grant & Aid Strategy (The "Pro-Bono" Lever): A critical component of the financial model is the non-monetary acquisition of fixed assets.

1. Equipment (UAVs, cameras) received from international donors or partners for humanitarian purposes is classified as charitable aid and is not subject to the 5% revenue tax.
2. Cash grants (e.g., Brave1, USF, NATO DIANA, Horizon Europe) are accounted for separately.

The strategic logic of the enterprise is summarized in the Lean Canvas (Table 4.5).

Table 4.5 – Lean Canvas for SpectralCA

Segment	*Description*
Problem	1. Slow manual demining speeds. 2. High risk to human life. 3. Inability to detect plastic mines with metal detectors.
Customer Segments	1. Gov: State Emergency Service (SES), Ministry of Defense. 2. NGO: HALO Trust, DRC, FSD. 3. Commercial: Agro-holdings (post-war land reclamation).
Unique Value Prop.	"Software-defined demining: 2x faster than manual teams, safer than dogs, cheaper than heavy machinery."
Solution	AI-driven hyperspectral analysis software + UAV deployment service.
Revenue Streams	1. Service Fee: Charge per hectare of scanned land. 2. Grant Funding: Non-dilutive capital for R&D. 3. SaaS Licensing: Future B2B sales to other drone operators.
Cost Structure	1. Payroll: Engineering & Pilot team. 2. Hardware Maint.: Batteries, propellers, sensor calibration. 3. Cloud Compute: GPU instance costs.
Key Metrics	1. Hectares scanned per day. 2. False Positive Rate (FPR). 3. Cost per hectare.

Marketing Strategy (4P):

1) product – high-precision hazard maps. The USP is "Safety through Remote Sensing";

2) price – market penetration strategy. Service is offered at cost-recovery rates for humanitarian partners to build a dataset and reputation. Commercial rates for agro-holdings will be market-based (~$50–100 per hectare);
3) place – direct B2G sales and partnerships with demining operators;
4) promotion – industry conferences, defense-tech hackathons (Brave1), and publication of scientific validation results.

The financial plan is calculated for the first 12 months of active operation.

Staffing and Payroll. The team consists of 4 core members. To optimize taxes, personnel are engaged as independent contractors. The projected salaries align with the current Ukrainian IT market [37]. In Q1 and Q2, while the product is in development, founders withdraw only subsistence-level stipends, scaling to market rates (Table 4.6) only after commercial revenue stabilizes in Q3.

Table 4.6 – Monthly Staffing Requirements and Salary Fund

Role	*Responsibilities*	*Monthly Rate (USD)*	*Annual Cost (USD)*
CEO / Team Lead	Strategy, Fundraising, Gov Relations	$1,500	$18,000
CTO / AI Engineer	Neural Network optimization, Cloud infra	$2,000	$24,000
UAV Operator	Flight planning, drone maintenance	$1,200	$14,400
BizDev Manager	Partner relations, securing pro-bono hardware	$1,000 + % success fee	$12,000
Total		**$5,700**	$68,400

Capital Expenditures (CAPEX) & Pro-Bono Savings (Table 4.7). This table highlights the efficiency of the BizDev strategy. By securing hardware as aid, the startup avoids massive initial debt.

Table 4.7 – Capital Expenditures and Pro-Bono Savings

Item	*Market Price (USD)*	*SpectralCA Cost (USD)*	*Source / Note*
UAV Platform (Matrice 300)	$12,000	$0	Grant / Partner Aid
Hyperspectral Sensor	$45,000	$0	Partner Loan / Aid
Workstation (GPU Server)	$3,500	$3,500	Own Investment / Grant
Field Equipment (Generator)	$1,500	$500	Partial Aid
Software Licenses	$1,000	$1,000	Annual subscription
Total CAPEX	**$63,000**	$5,000	Savings: ~$58,000

Operating Expenses (OPEX) (Table 4.8). Monthly fixed costs required to keep the business running.

Profitability analysis. The break-even point is calculated based on a projected service fee of $100 per hectare (commercial rate) or equivalent grant funding.

1. Total monthly cost (burn rate): salaries ($5,700) + OPEX ($1,700) = $7,400.
2. Break-even volume: $7,400 / $100 = 74 hectares per month.

Table 4.8 – Monthly Operating Expenses

Expense Category	*Description*	*Cost (USD)*
Office/Lab Rent	Co-working space or small lab	$300
Cloud Computing	Google Colab Pro+	$100
Logistics	Fuel for field missions	$400
Internet/Comms	Starlink subscription	$100
Hardware Maintenance	Spare props, batteries	$200
Taxes (5%)	Estimated on $10k revenue	$500
Legal/Accounting	Outsourced	$100
Total Monthly OPEX		**$1,700**

Given that a single UAV flight can cover ~10–20 hectares, and multiple flights can be conducted per day, reaching 74 hectares/month is operationally feasible (requires ~4-5 active field days per month).

A comprehensive risk analysis is vital for defense-tech startups, shown in Table 4.9.

Expansion Strategy (Stage 3). While landmine detection is the foundational mission, the core technology – Spectral Anomaly Detection – is sector-agnostic. Future revenue streams will diversify into:

1) agriculture – early detection of crop diseases (Fusarium, blight) invisible to RGB cameras;
2) ecology – identification of oil spills and illegal waste dumps;
3) infrastructure – detecting rust and metal fatigue on bridges/pipelines via spectral signatures.

Table 4.9 – Risk Register and Mitigation Strategies

Risk Category	*Risk Description*	*Probability*	*Impact*	*Mitigation Strategy*
Technical	UAV Crash / Loss due to EW (Electronic Warfare).	Medium	High	Operate in "safe" rear zones initially; use inertial navigation; insurance.
Legislative	Abolition of simplified tax system.	High	High	Budgeting for a senior accountant ($800/mo).
Financial	Grant Rejection.	Medium	High	Bootstrap mode (cut salaries); pivot to commercial agro-services earlier.
Legal	Certification Delays.	High	Medium	Partner with already certified operators (SES) as a sub-contractor.
Operational	Weather Conditions.	High	Low	Flexible mission planning; focus on data processing during downtime.

Operational Implementation Plan. To ensure the disciplined execution of the business strategy, a roadmap for the first 12 months of operation has been developed (Table 4.10). The timeline is divided into four strategic phases: Legal Setup, R&D Finalization, Pilot Testing, and Commercial Launch.

Table 4.10 – Project Implementation Schedule (Year 1)

Phase	*Month*	*Key Activities*	*Responsible*	*Deliverable*
I. Setup & Legal	M1	1. Registration of Private Entrepreneur of Group 3. 2. Opening multi-currency bank accounts. 3. Signing NDA/IP agreements.	CEO	Legal Entity Functional
II. Tech Acquisition	M2-M3	1. Negotiations with camera manufacturers (Headwall/Resonon) for pro-bono hardware. 2. Grant application submission (Brave1, USF). 3. Setup of Cloud GPU infrastructure.	BizDev / CTO	Hardware Secured
III. R&D & Integration	M4-M6	1. Physical mounting of sensor to UAV. 2. Integration of SpectralCA software with sensor SDK. 3. Field calibration flights.	CTO / Pilot	Integrated Prototype (MVP)

End of Table 4.10

Phase	*Month*	*Key Activities*	*Responsible*	*Deliverable*
IV. Pilot Operations	M7-M9	1. Joint field tests with SES (State Emergency Service) at certified test ranges. 2. Dataset expansion (real minefield data). 3. Model Fine-Tuning (Transfer Learning).	CEO / Pilot	Certified Tech Report
V. Commercialization	M10-M12	1. First commercial service contract (Humanitarian Demining NGO). 2. Marketing campaign for the Agricultural sector. 3. Signing first B2B agro-monitoring contract.	BizDev	First Revenue

Advanced Marketing and Sales Strategy. Given the dual-use nature of the technology, SpectralCA employs a bifurcated sales strategy targeting two distinct customer personas: the Government/NGO Partner and the Agricultural Client (Table 4.11).

Table 4.11 – Marketing Budget and Channels (Year 1)

Channel	*Activity*	*Estimated Cost (USD)*	*Expected Outcome*
Industry Conferences	Exhibiting at Defense forums.	$1,000	5-10 B2B Leads
Digital Content	Publishing case studies on LinkedIn/Website.	$200 (Ads)	Brand Authority
Direct Sales	Travel expenses for meetings with Agro-holding directors.	$800	2 Pilot Projects
Demo Days	Field demonstrations for SES representatives.	$500 (Logistics)	Gov Certification
Total		**$2,500**	

The B2G (Business-to-Government) Sales Funnel. The sales cycle for demining services is long (6–12 months) and trust-based:

- awareness – participation in closed defense-tech hackathons and industry roundtables;
- validation – providing "Proof of Concept" (PoC) reports where SpectralCA maps are compared against ground-truth manual demining logs;
- partnership – signing a Memorandum of Understanding (MoU) with certified operators (e.g., HALO Trust, DRC);
- contracting – sub-contracting for survey missions.

The B2B (Business-to-Business) Agricultural Funnel. The agricultural market requires a faster, cost-driven approach:

- channel – partnership with existing drone-spraying service providers;

- offer – *"Add hyperspectral analysis to your spraying service for +$5/hectare"*;
- value – SpectralCA provides the software layer, while the partner provides the drone and pilot, creating a scalable revenue share model.

Detailed Financial Projections. While the profitability analysis provided a snapshot, the cash flow statement offers a dynamic view of the company's liquidity. The model assumes that the first 6 months are funded via Grants ($25,000 total), and revenue operations commence in Month 7. The projected cash flow statement is in Table 4.12.

Table 4.12 – Projected Cash Flow Statement (USD)

Category	*Q1 (M1-3)*	*Q2 (M4-6)*	*Q3 (M7-9)*	*Q4 (M10-12)*	*Total Year 1*
Inflows (income)					
Grant Funding	$25,000	$5,000	$0	$0	**$30,000**
Service Revenue (Demining)	$0	$0	$15,000	$30,000	**$45,000**
Service Revenue (Agro)	$0	$2,000	$5,000	$8,000	**$15,000**
Total Inflow	**$25,000**	**$7,000**	**$20,000**	**$38,000**	**$90,000**
Outflows (expenses)					
Payroll (Bootstrap in Q1-Q2)	$6,000	$9,000	$17,100	$17,100	**$49,200**
CAPEX (Servers/Field Gear)	$3,500	$1,500	$0	$1,000	**$6,000**

End of Table 4.12

Category	*Q1 (M1-3)*	*Q2 (M4-6)*	*Q3 (M7-9)*	*Q4 (M10-12)*	*Total Year 1*
OPEX					
Rent, Cloud, Fuel	$3,000	$3,000	$3,000	$3,000	**$12,000**
Account.and Bank Fees	$690	$690	$690	$690	**$2,760**
Marketing & Conf.	$500	$1,000	$500	$500	**$2,500**
Taxes					
Taxes (5% + 1%)	$0	$120	$1,200	$2,280	**$3,600**
Social	$150	$150	$150	$150	**$600**
Total Outflow	**$13,840**	**$15,460**	**$22,640**	**$24,720**	**$76,660**
Net Cash Flow	**+$11,160**	**-$8,460**	**-$2,640**	**+$13,280**	**+$13,340**
Cumulative Cash Balance	**$11,160**	**$2,700**	**$60**	**$13,340**	

<u>*Financial Assessment*</u>. The strategy successfully avoids the "Valley of Death".

- Q1 Buffer: The $25,000 Brave1 grant creates an immediate safety cushion of ~$11,000.
- Q2-Q3 Stability: Although monthly burn exceeds revenue in Q2, the cumulative reserves (+$2,700 at end of Q2) prevent bankruptcy. The explicit reduction of founder salaries during this period is the key stabilizer.

- Q4 Growth: By Q4, commercial contracts fully cover operational costs, returning the company to strong profitability.

Sensitivity Analysis. Startups operate in high-uncertainty environments. A sensitivity analysis evaluates how changes in key variables affect the financial viability of the project (Table 4.13).

Table 4.13 – Scenario Analysis for Year 1

Scenario	*Assumptions*	*Revenue*	*Outcome*
Pessimistic	No Grants received. Certification delayed by 6 months. Only minimal B2B Agro sales.	\$15,000	Bankruptcy without external investor injection. Team works pro-bono.
Realistic	\$20k Grant secured. Demining pilot successful in M9. Agro sales moderate.	\$75,000	Break-even achieved by M11. Founders take partial salary.
Optimistic	\$50k Grant secured (Brave1). Hardware partner provides 2 drones. Large SES contract signed in M7.	\$150,000	Profitable. Full salaries paid. R&D budget doubles for Year 2.

Social and Economic Impact Assessment. Beyond financial metrics, SpectralCA generates significant non-monetary value. In the context of the "Tech for Good" paradigm, measuring this impact is essential for attracting donors and government support.

Humanitarian Impact:

- lives saved: every detected mine represents a potential life saved or amputation prevented;

- land release – returning demined land to communities allows displaced persons to return home;
- safety – by replacing human sappers with UAVs for the initial survey, the risk of injury to demining personnel is reduced by an estimated 80%.

Economic Impact (Agricultural Restoration). One hectare of mined arable land in Ukraine generates $0 in value. If SpectralCA facilitates the clearing of 1,000 hectares in Year 1:

- average wheat yield: 4 tons/ha;
- market price: ~$150/ton [38];
- economic value generated: $1{,}000 \times 4 \times 150 = \$600{,}000$ returned to the Ukrainian economy;
- ROI for the state: the cost of the service (~$100k) yields a 6x economic return in the first harvest alone.

Ecological Impact:

- precision demining: SpectralCA's high-precision maps allow for targeted neutralization rather than carpet-bombing or heavy mechanical tiling, preserving the topsoil structure.
- carbon footprint: using electric UAVs for scanning releases significantly less CO_2 than heavy diesel-powered flail machines used in mechanical demining.

4.5 Future R&D Roadmap

To maintain a competitive edge, SpectralCA plans a continuous cycle of innovation. The technical roadmap extends the research initiated in Chapter 3 (Table 4.14).

Table 4.14 – Technical Evolution Roadmap

Version	*Timeline*	*Key Features*	*Hardware Requirement*
v1.0 (MVP)	Year 1	Offline processing. SpectralCA Base Model. Manual flight planning.	Standard Laptop (Post-flight)
v2.0 (Edge)	Year 2	Real-time inference on UAV. Integration of Jetson Orin. Auto-generated hazard heatmaps.	Onboard Edge Computer
v3.0 (Fusion)	Year 3	Multi-sensor fusion: Hyperspectral + LiDAR + Magnetometer. 3D sub-surface modeling.	Custom UAV Payload
v4.0 (Swarm)	Year 4	Cooperative swarm scanning (3-5 drones simultaneously). Distributed neural network processing.	Drone Swarm Protocol

Conclusion to Section 4

The implementation of the SpectralCA startup represents a financially viable and socially impactful venture. The rigorous analysis conducted in this chapter demonstrates that:

1) Market Demand is Critical: The intersection of the Ukrainian demining crisis and the global need for precision agriculture creates a robust dual-market opportunity.
2) Financial Feasibility is Achievable: Through a lean operational model (FOP Group 3) and a strategic "Pro-Bono Hardware" acquisition plan, the startup

significantly lowers the barrier to entry. The cash flow analysis indicates that with a seed grant of $20,000, the company can sustain operations until commercial revenue stabilizes in Q3.

3) Impact is Scalable: The technology does not just generate profit; it acts as a force multiplier for humanitarian efforts, potentially unlocking millions of dollars in economic value by returning mined land to production.

By following the outlined implementation plan, SpectralCA is positioned to evolve from a university research project into a key technological player in Ukraine's recovery and a global innovator in remote sensing analytics.

CONCLUSIONS

A new hybrid neural network architecture for hyperspectral image classification was developed, based on a modification of the MDvT model with the introduction of a SpectralCA (Spectral Cross-Attention) block. The proposed module enables more effective modeling of the relationships between spectral and spatial features through a bidirectional cross-attention mechanism, which integrates information from both types of features in a symmetric and complementary way. This approach reduces the number of model parameters and halves both training and inference time compared to the baseline MobileViTBlock, while still maintaining high classification accuracy.

Experimental analysis showed that although the new architecture is slightly behind the baseline version by approximately 2% in terms of absolute metric values (OA, AA, Kappa), it provides significant improvements in speed (2 times faster inference) and resource efficiency (14.5% fewer parameters). This is particularly important for deployment in resource-constrained environments, such as mobile devices and unmanned aerial vehicles (UAVs) operating in real-time field conditions. The reduced parameter count also facilitates adaptation of the model to new tasks and simplifies integration with other modules within larger remote sensing or environmental monitoring systems.

The comparison between the baseline MDvT and the version utilizing the proposed module was conducted under identical experimental conditions (including the number of training epochs and dataset size). Consequently, the algorithm's accuracy has the potential to be further elevated by increasing the volume of training data and the number of iterations. The current accuracy limits are partially attributed to the restricted number of training epochs relative to the increased complexity of the SpectralCA connections. While extending training may increase the training duration, this is an acceptable trade-off, as the primary objective is optimizing inference speed, which remains consistently twice as fast.

A notable advantage of the proposed architecture is its flexibility: the SpectralCA module can be integrated either as a full replacement for individual blocks in the transformer structure or as an additional component in the early layers of the model. This flexibility opens opportunities for further refinement and adaptation of the network to specific data types or application domains. In particular, one promising direction is the application of the model to UAV computer vision tasks, where the combination of real-time inference speed and hyperspectral feature discrimination could significantly enhance environmental perception, terrain classification, and autonomous decision-making.

A particularly important component of the implemented system was the integration of a semi-supervised learning method based on self-training with proxy labeling. Thanks to the iterative expansion of the training set using highly probable pseudo-labels, it was possible to improve the accuracy of the model even in cases where the amount of manually annotated data was limited. This opens up prospects for using the model in real-world conditions where complete labeling is not available, and also lays the foundation for the future use of more advanced semi-supervised or active learning strategies.

The practical feasibility of the developed technology was validated through a comprehensive startup implementation analysis. The research demonstrated that the SpectralCA system can be deployed as a commercially viable dual-use product. The proposed business model, which combines proprietary AI software with partner-supplied UAV hardware, significantly lowers entry barriers. Financial projections indicate that the project allows for a rapid return on investment while generating substantial social impact by accelerating humanitarian demining operations and restoring agricultural land potential.

The strategic roadmap for field deployment involves a transfer learning approach: gathering a massive corpus of generic hyperspectral data for model pre-training, followed by fine-tuning on the specific, limited samples of real mines.

The main findings of the study were presented at conferences, highlighted in the author's publications, and won the competitions from the following list.

1. D.V. Brovko. SpectralCA: Bi-Directional Cross-Attention for Next-Generation UAV Hyperspectral Vision. arXiv, 2025. Pre-print [39].
2. D.V. Brovko. SpectralCA: Utilising Hyper-Spectral Imaging in UAV Navigation. IEEE 8th International Conference on Methods and Systems of Navigation and Motion Control, 2025. Already presented, the proceedings are in print [40].
3. D.V. Brovko. SpectralCA: Bi-Directional Cross-Attention for Next-Generation UAV Hyperspectral Vision. International Student Research Competition in Artificial Intelligence, 2025. First place in the competition [41].

In conclusion, SpectralCA is a high-performance neural network module designed for the extraction of high-quality hyperspectral features. While it introduces more intricate architectural connections, it achieves a reduction in both parameter count and computational complexity. The proposed model operates two times faster than the baseline MDvT with MobileViTBlock. It maintains an accuracy of approximately 95% while reducing the total number of parameters by 14.5%, which corresponds to a decrease of approximately 1.1 million parameters. The results of the study demonstrate the feasibility and effectiveness of using a modified MDvT hybrid architecture with a SpectralCA module and the application of self-training in hyperspectral image classification tasks. The proposed model demonstrates strong potential for real-world use in environmental monitoring, agriculture, UAV-based navigation, security applications, and other domains where accuracy, efficiency, and adaptability to limited resources are critical.

REFERENCES

1. Over the year, the scope of humanitarian demining in Ukraine has increased by 50% / Ministry of Defence of Ukraine. 2025. URL: https://mod.gov.ua/en/news/over-the-year-the-scope-of-humanitarian-demining-in-ukraine-has-increased-by-50 (date of access: 31st October 2025).
2. Ukraine's contaminated land: Clearing landmines with rakes, tractors and drones / Bellingcat. 2025. URL: https://www.bellingcat.com/news/2025/07/02/ukraines-contaminated-land-clearing-landmines-with-rakes-tractors-and-drones/ (date of access:31st October 2025).
3. Sineglazov V. M., Karabetsky D. P. Energy system design of solar aircraft. Proc. 2013 IEEE 2nd Int. Conf. Actual Problems of Unmanned Air Vehicles Developments (APUAVD). 2013. P. 9–11. URL: https://ieeexplore.ieee.org/document/6705267 (date of access: 3rd November 2025).
4. Sineglazov V. M., Dolgorukov S. O. Test bench of UAV navigation equipment. Proc. 2014 IEEE 3rd Int. Conf. Methods and Systems of Navigation and Motion Control (MSNMC). 2014. P. 108–111. DOI: 10.1109/MSNMC.2014.6979743.
5. Sineglazov V., Shildskyi S. Navigation systems based on GSM. Proc. 2014 IEEE 3rd Int. Conf. Methods and Systems of Navigation and Motion Control (MSNMC). 2014. P. 95–98. DOI: 10.1109/MSNMC.2014.6979740.
6. Sineglazov V. M. Multi-functional integrated complex of detection and identification of UAVs. Proc. 2015 IEEE 3rd Int. Conf. Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD). 2015. P. 320–323. URL: https://www.researchgate.net/publication/308829015_Multi-functional_integrat

ed_complex_of_detection_and_identification_of_UAVs
(date of access: 3rd November 2025).

7. 2014 IEEE 3rd International Conference on Methods and Systems of Navigation and Motion Control, MSNMC 2014 – Proceedings. 2014. P. 108–111. DOI: 10.1109/MSNMC.2014.6979743.
8. Lutsky M. G., Sineglazov V. M., Ishchenko V. S. Suppression of noise in visual navigation systems. Proc. 2021 IEEE 6th Int. Conf. Actual Problems of Unmanned Aerial Vehicles Development (APUAVD). 2021. P. 7–10.
9. Hyperspectral imaging and its applications: A review / A. Bhargava et al. Heliyon. 2024. Vol. 10. e33208. DOI: 10.1016/j.heliyon.2024.e33208. URL: https://www.sciencedirect.com/science/article/pii/S2405844024092399
(date of access: 4th November 2025).
10. Hyperspectral systems increase imaging capabilities / NASA Spinoff. 2010. URL: https://spinoff.nasa.gov/Spinoff2010/hm_4.html
(date of access: 4th November 2025).
11. Lu G., Fei B. Medical hyperspectral imaging: A review. J. Biomed. Opt. 2014. Vol. 19, no. 1. 010901. DOI: 10.1117/1.JBO.19.1.010901. URL: https://pubmed.ncbi.nlm.nih.gov/24441941/
(date of access: 5th November 2025).
12. Dimensionality reduction of hyperspectral images based on improved spatial-spectral weight manifold embedding / H. Liu et al. Sensors. 2020. Vol. 20, no. 16. 4413. DOI: 10.3390/s20164413. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7472477/
(date of access: 5th November 2025).
13. Deep supervised learning for hyperspectral data classification through convolutional neural networks / K. Makantasis et al. Proc. IGARSS. 2015. P. 4959–4962. DOI: 10.1109/IGARSS.2015.7326900.
14. Deep learning classifiers for hyperspectral imaging: A review / M. E. Paoletti et al. ISPRS J. Photogramm. Remote Sens. 2019. Vol. 158. P. 279–317. DOI: 10.1016/j.isprsjprs.2019.09.006.

15. Spatial information based support vector machine for hyperspectral image classification / B.-C. Kuo et al. Proc. 2010 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). 2010. P. 832–835. DOI: 10.1109/IGARSS.2010.5651433. URL: https://ieeexplore.ieee.org/document/5651433 (date of access: 6th November 2025).

16. Hyperspectral Image Classification Based on 3D–2D Hybrid Convolution and Graph Attention Mechanism / J. Li et al. Neural Process. Lett. 2024. DOI: 10.1007/s11063-024-11584-2. URL: https://link.springer.com/article/10.1007/s11063-024-11584-2 (date of access: 6th November 2025).

17. Khadse S., Shivakumar N., Ramisetty K. R. Deep learning techniques for hyperspectral image analysis in agriculture: A review. arXiv:2304.13880. 2023. URL: https://arxiv.org/abs/2304.13880 (date of access: 7th November 2025).

18. Deep learning in medical hyperspectral images: A review / R. Cui et al. Sensors. 2022. Vol. 22, no. 24. 9790. DOI: 10.3390/s22249790. URL: https://www.mdpi.com/1424-8220/22/24/9790 (date of access: 7th November 2025).

19. Cherifi M., Cherifi H., Cherifi C. Dimensionality reduction for hyperspectral image classification. EasyChair Preprint No. 11193. 2023. URL: https://easychair.org/publications/preprint/Kq75 (date of access: 10th November 2025).

20. HybridSN: Exploring 3D-2D CNN feature hierarchy for hyperspectral image classification / S. K. Roy et al. IEEE Geosci. Remote Sens. Lett. 2020. Vol. 17, no. 2. P. 277–281. DOI: 10.1109/LGRS.2019.2918719.

21. LiDAR-guided cross-attention fusion for hyperspectral band selection and image classification / J. X. Yang et al. arXiv:2404.03883. 2024. URL: https://arxiv.org/abs/2404.03883 (date of access: 10th November 2025).

22. MFSA-Net: Semantic segmentation with camera–LiDAR cross-attention fusion based on fast neighbor feature aggregation / Y. Duan et al. IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. 2024. P. 1-14. DOI: 10.1109/JSTARS.2024.3472751.

23. Dual Selective Fusion Transformer Network for Hyperspectral Image Classification / Y. Xu et al. arXiv:2410.03171. 2024. URL: https://arxiv.org/abs/2410.03171 (date of access: 11th November 2025).

24. SpectralFormer: Rethinking hyperspectral image classification with transformers / D. Hong et al. IEEE Trans. Geosci. Remote Sens. 2022. Vol. 60. 5527212. DOI: 10.1109/TGRS.2021.3130716. URL: https://arxiv.org/abs/2107.02988 (date of access: 11th November 2025).

25. MDvT: Introducing mobile three-dimensional convolution to a vision transformer for hyperspectral image classification / X. Zhou et al. Int. J. Digital Earth. 2023. Vol. 16, no. 1. P. 1469–1490. DOI: 10.1080/17538947.2023.2202423. URL: https://doi.org/10.1080/17538947.2023.2202423 (date of access: 12th November 2025).

26. Spectral-spatial residual network for hyperspectral image classification: A 3-D deep learning framework / P. Zhong et al. IEEE Trans. Geosci. Remote Sens. 2018. Vol. 56, no. 2. P. 847–858. DOI: 10.1109/TGRS.2017.2755542.

27. WHU-Hi dataset resource page / Wuhan University. URL: http://rsidea.whu.edu.cn/resource_WHUHi_sharing.htm (date of access: 12th November 2025).

28. 3D convolution / Papers With Code. URL: https://paperswithcode.com/method/3d-convolution (date of access: 13th November 2025).

29. Carranza-García M., García-Gutiérrez J., Riquelme J. C. A framework for evaluating land use and land cover classification using convolutional neural networks. Remote Sens. 2019. Vol. 11, no. 3. 274. DOI: 10.3390/rs11030274.

URL: https://www.mdpi.com/2072-4292/11/3/274 (date of access: 13th November 2025).

30. Classification of hyperspectral data using a multi-channel convolutional neural network / C. Chen et al. Intelligent Computing Methodologies. Cham: Springer, 2018. P. 81–92. DOI: 10.1007/978-3-319-95957-3_10. URL: https://link.springer.com/chapter/10.1007/978-3-319-95957-3_10 (date of access: 13th November 2025).

31. Classification of hyperspectral–LiDAR dual-view data with hybrid feature and trusted decision fusion / J. Liu et al. Remote Sens. 2024. Vol. 16, no. 23. 4381. DOI: 10.3390/rs16234381. URL: https://www.mdpi.com/2072-4292/16/23/4381 (date of access: 13th November 2025).

32. Nano HP (400-1000nm) Hyperspectral Imaging Package / Headwall Photonics. 2023. URL: https://headwallphotonics.com/products/remote-sensing/nano-hp-400-1000nm-hyperspectral-imaging-package/ (date of access: 28th November 2025).

33. Support for Matrice 300 RTK / DJI. URL: https://www.dji.com/global/support/product/matrice-300 (date of access: 28th November 2025).

34. 7 Cheapest Cloud GPU Providers in 2025 / Northflank. 2025. URL: https://northflank.com/blog/cheapest-cloud-gpu-providers (date of access: 1st December 2025).

35. Nvidia Blackwell vs. Competitors: A Comparative Analysis / Medium. 2024. URL: https://gunjanvi.medium.com/nvidia-blackwell-vs-competitors-a-comparative-analysis-49d0b344374a (date of access: 2nd December 2025).

36. Tax Code of Ukraine. Article 291. Simplified system of taxation, accounting and reporting / Verkhovna Rada of Ukraine. 2024. URL: https://zakon.rada.gov.ua/laws/show/2755-17 (date of access: 3rd December 2025).

37. Salaries of Ukrainian Developers – Summer 2025 / DOU.ua. 2025. URL: https://dou.ua/lenta/articles/salary-report-devs-summer-2025/ (date of access: 4th December 2025).

38. Average prices for grain crops in Ukraine (Wheat Class 3) / APK-Inform Agency. 2025. URL: https://www.apk-inform.com/en/prices (date of access: 4th December 2025).

39. Brovko D. V. SpectralCA: Bi-Directional Cross-Attention for Next-Generation UAV Hyperspectral Vision. arXiv preprint. 2025. URL: https://arxiv.org/pdf/2510.09912 (date of access: 5th December 2025).

40. Brovko D. V. SpectralCA: Utilising Hyper-Spectral Imaging in UAV Navigation. Proc. IEEE 8th International Conference on Methods and Systems of Navigation and Motion Control. 2025. URL: https://msnmc.ieee.org.ua/wp-content/uploads/2025/10/Program_MSNMC2025.pdf (date of access: 8th December 2025).

41. Brovko D. V. SpectralCA: Bi-Directional Cross-Attention for Next-Generation UAV Hyperspectral Vision. International Student Research Competition in Artificial Intelligence. 2025. URL: https://aic.kpi.ua/wp-content/uploads/2025/11/spectralca.pdf (date of access: 9th December 2025).

APPENDIX A SOURCE CODE OF SPECTRALCA AND SSL

This appendix provides the core implementation details of the SpectralCA module and the SSL pipeline developed in this study. The code is written in Python utilizing the PyTorch deep learning framework.

The provided snippets focus on the novel contributions of this research. The implementation of the baseline MDvT architecture, into which these modules are integrated, follows the methodology described in the original paper "MDvT: Introducing mobile three-dimensional convolution to a vision transformer for hyperspectral image classification" by Zhou et al. [25].

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class ConvLayer(nn.Module):
    def __init__(self, in_channels, out_channels,
kernel_size, stride=1, groups=1,
                 bias=False, use_norm=True,
use_act=True, norm_type='group'):
    super().__init__()
    if isinstance(kernel_size, int):
        kernel_size = (kernel_size,) * 3
    if isinstance(stride, int):
        stride = (stride,) * 3
    padding = tuple((k - 1) // 2 for k in kernel_size)

    layers = [nn.Conv3d(in_channels, out_channels,
kernel_size, stride,
                        padding, groups=groups,
bias=bias)]
```

```
    if use_norm:
        if norm_type == 'group':
            layers.append(nn.GroupNorm(num_groups=8,
num_channels=out_channels))
        else:
            layers.append(nn.BatchNorm3d(out_channels))
    if use_act:
        layers.append(nn.SiLU())
    self.block = nn.Sequential(*layers)

    def forward(self, x):
    return self.block(x)


class FeedForward(nn.Module):
    def __init__(self, dim, hidden_dim, dropout=0.):
    super().__init__()
    self.net = nn.Sequential(
        nn.Linear(dim, hidden_dim),
        nn.SiLU(),
        nn.Dropout(dropout),
        nn.Linear(hidden_dim, dim),
        nn.Dropout(dropout)
    )

    def forward(self, x):
    return self.net(x)

class CrossAttention(nn.Module):
    def __init__(self, dim, num_heads=4, attn_drop=0.,
proj_drop=0.):
    super().__init__()
    self.num_heads = num_heads
    self.dim = dim
```

```
    assert dim % num_heads == 0, f"dim {dim} not
divisible by num_heads {num_heads}"
    self.head_dim = dim // num_heads
    self.scale = self.head_dim ** -0.5

    self.q_spatial = nn.Linear(dim, dim)
    self.k_spectral = nn.Linear(dim, dim)
    self.v_spectral = nn.Linear(dim, dim)

    self.q_spectral = nn.Linear(dim, dim)
    self.k_spatial = nn.Linear(dim, dim)
    self.v_spatial = nn.Linear(dim, dim)

    self.proj_spatial = nn.Linear(dim, dim)
    self.proj_spectral = nn.Linear(dim, dim)
    self.attn_drop = nn.Dropout(attn_drop)
    self.proj_drop = nn.Dropout(proj_drop)

    def forward(self, spatial, spectral):
    # Get batch sizes and sequence lengths
    B_spatial, N_spatial, C = spatial.shape
    B_spectral, N_spectral, C = spectral.shape

    # Make sure batch sizes match
    assert B_spatial == B_spectral, f"Batch sizes must
match: {B_spatial} vs {B_spectral}"
    B = B_spatial

    # Ensure dimension matches
    assert C == self.dim, f"Input dimension {C} doesn't
match model dimension {self.dim}"
```

```
    # Define generalized attention function that handles
different sequence lengths
    def attend(q, k, v):
        H = self.num_heads
        D = C // H

        # Get sequence length from the queries
        N_q = q.size(1)
        N_kv = k.size(1)

        # Reshape to multi-head attention format
        q = q.reshape(B, N_q, H, D).permute(0, 2, 1, 3)
# [B, H, N_q, D]
        k = k.reshape(B, N_kv, H, D).permute(0, 2, 1,
3)  # [B, H, N_kv, D]
        v = v.reshape(B, N_kv, H, D).permute(0, 2, 1,
3)  # [B, H, N_kv, D]

        # Compute attention scores
        attn = (q @ k.transpose(-2, -1)) * self.scale
# [B, H, N_q, N_kv]
        attn = F.softmax(attn, dim=-1)
        attn = self.attn_drop(attn)

        # Apply attention to values
        out = (attn @ v).transpose(1, 2).reshape(B,
N_q, C)  # [B, N_q, C]
        return out

    # Spatial -> Spectral attention
    q_s = self.q_spatial(spatial)  # [B, N_spatial, C]
    k_sp = self.k_spectral(spectral)  # [B, N_spectral,
C]
```

```
    v_sp = self.v_spectral(spectral)  # [B, N_spectral,
C]
    sa = attend(q_s, k_sp, v_sp)  # [B, N_spatial, C]
    sa = self.proj_drop(self.proj_spatial(sa))

    # Spectral -> Spatial attention
    q_sp = self.q_spectral(spectral)  # [B, N_spectral,
C]
    k_s = self.k_spatial(spatial)  # [B, N_spatial, C]
    v_s = self.v_spatial(spatial)  # [B, N_spatial, C]
    spa = attend(q_sp, k_s, v_s)  # [B, N_spectral, C]
    spa = self.proj_drop(self.proj_spectral(spa))

    return sa, spa

class SpectralCA(nn.Module):
    def __init__(self, in_channels, transformer_dim,
patch_h=2, patch_w=2, patch_d=2,
                 num_heads=4, attn_dropout=0.,
dropout=0.1, ffn_dropout=0.1,
                 img_size=None, channel_depth=None,
depth=None, mode=None):
        super().__init__()

        self.inner_dim = transformer_dim
        self.patch_h, self.patch_w, self.patch_d = patch_h,
patch_w, patch_d

        self.spatial_conv = nn.Sequential(
            nn.Conv2d(in_channels, self.inner_dim,
kernel_size=3, padding=1),
            nn.BatchNorm2d(self.inner_dim),
            nn.SiLU()
```

```
    )

    self.spectral_conv = nn.Sequential(
        nn.Conv3d(in_channels, self.inner_dim,
kernel_size=3, padding=1),
        nn.BatchNorm3d(self.inner_dim),
        nn.SiLU()
    )

    self.cross_attn = CrossAttention(self.inner_dim,
num_heads, attn_drop=attn_dropout, proj_drop=dropout)

    self.norm1_spatial = nn.LayerNorm(self.inner_dim)
    self.norm1_spectral = nn.LayerNorm(self.inner_dim)
    self.norm2_spatial = nn.LayerNorm(self.inner_dim)
    self.norm2_spectral = nn.LayerNorm(self.inner_dim)

    self.ffn_spatial = nn.Sequential(
        nn.Linear(self.inner_dim, self.inner_dim * 2),
        nn.SiLU(),
        nn.Dropout(ffn_dropout),
        nn.Linear(self.inner_dim * 2, self.inner_dim),
        nn.Dropout(ffn_dropout)
    )

    self.ffn_spectral = nn.Sequential(
        nn.Linear(self.inner_dim, self.inner_dim * 2),
        nn.SiLU(),
        nn.Dropout(ffn_dropout),
        nn.Linear(self.inner_dim * 2, self.inner_dim),
        nn.Dropout(ffn_dropout)
    )
```

```
    self.out_proj = nn.Conv3d(self.inner_dim * 2,
in_channels, kernel_size=1)

    def forward(self, x):
    B, C, H, W, D = x.shape
    res = x

    # Spatial: mean across D -> 2D input
    spatial_input = x.mean(dim=4)  # [B, C, H, W]
    spatial = self.spatial_conv(spatial_input)  # [B,
inner_dim, H, W]

    # Flatten spatial to sequence for attention
    spatial = spatial.permute(0, 2, 3, 1).reshape(B, H *
W, self.inner_dim)  # [B, H*W, inner_dim]

    # Spectral: keep full 3D
    spectral = self.spectral_conv(x)  # [B, inner_dim,
H, W, D]
    # Flatten spectral to sequence for attention
    spectral = spectral.reshape(B, self.inner_dim, H * W
* D).transpose(1, 2)  # [B, H*W*D, inner_dim]

    # Apply normalization before attention
    spatial_norm = self.norm1_spatial(spatial)  # [B,
H*W, inner_dim]
    spectral_norm = self.norm1_spectral(spectral)  # [B,
H*W*D, inner_dim]

    # Cross attention between spatial and spectral
    sa, spa = self.cross_attn(spatial_norm,
spectral_norm)
```

```
        # Skip connection + norm
        spatial = spatial + sa  # [B, H*W, inner_dim]
        spectral = spectral + spa  # [B, H*W*D, inner_dim]

        # Second normalization
        spatial_norm2 = self.norm2_spatial(spatial)
        spectral_norm2 = self.norm2_spectral(spectral)

        # FFN
        spatial = spatial + self.ffn_spatial(spatial_norm2)
# [B, H*W, inner_dim]
        spectral = spectral +
self.ffn_spectral(spectral_norm2)  # [B, H*W*D,
inner_dim]

        # Reshape back
        spatial = spatial.reshape(B, H, W,
self.inner_dim).permute(0, 3, 1, 2)  # [B, inner_dim, H,
W]
        spectral = spectral.transpose(1, 2).reshape(B,
self.inner_dim, H, W, D)  # [B, inner_dim, H, W, D]

        # Concatenate spatial and spectral features
        # Expand spatial to match 3D dimensions
        spatial_expanded = spatial.unsqueeze(-1).expand(-1,
-1, -1, -1, D)  # [B, inner_dim, H, W, D]
        combined = torch.cat([spatial_expanded, spectral],
dim=1)  # [B, 2*inner_dim, H, W, D]

        # Project back to original channel dimensions
        out = self.out_proj(combined)  # [B, C, H, W, D]

        # Ensure output has same shape as residual
```

```
    if out.shape != res.shape:
        out = F.interpolate(out, size=res.shape[2:],
mode="trilinear", align_corners=False)

    # Residual connection
    return out + res

class SpectralCA_MDvT(nn.Module):
    def __init__(self, image_size, channels,
num_classes, dim=64, kernels=[3, 3, 3], strides=[2, 2,
2],
                 heads=[2, 4, 4], depth=[2, 4, 6],
pool='cls', dropout=0.1, emb_dropout=0.1, scale_dim=4,
mode="ViT"):
    super().__init__()
    assert pool in {'cls', 'mean'}, 'pool type must be
either cls (cls token) or mean (mean pooling)'
    self.pool = pool
    self.dim = dim
    self.heads = heads
    ##### Stage 1 #######
    # n is spectral dimention
    self.conv1 = nn.Sequential(
        nn.Conv3d(in_channels=1, out_channels=dim,
kernel_size=kernels[0], stride=strides[0], padding=1),
        nn.BatchNorm3d(dim),
        nn.ReLU(inplace=True),
    )
    self.stage1_transformer = nn.Sequential(
        SpectralCA(in_channels=dim, transformer_dim=96,
img_size=(image_size + 1) // 2,
                        channel_depth=(channels + 1) //
2,
```

```
                            depth=depth[0],
num_heads=heads[0], mode=mode),
    )

    ##### Stage 2 #######
    in_channels = dim
    scale = 2
    channels = (channels + 1) // 2
    image_size = (image_size + 1) // 2
    dim = scale * dim
    self.conv2 = nn.Sequential(
        nn.Conv3d(in_channels, dim, kernels[1],
strides[1], 1),
        nn.BatchNorm3d(dim),
        nn.ReLU(inplace=True),
    )
    self.stage2_transformer = nn.Sequential(
        SpectralCA(in_channels=dim,
transformer_dim=120, img_size=(image_size + 1) // 2,
                            channel_depth=(channels + 1) //
2, depth=depth[1], num_heads=heads[1], mode=mode),
    )

    ##### Stage 3 #######
    in_channels = dim
    # //((heads[1]//heads[0]))
    scale = 2
    channels = (channels + 1) // 2
    image_size = (image_size + 1) // 2
    dim = (scale * dim)
    self.conv3 = InvertedResidual(in_channels, dim, 2,
1, 6)
    self.stage3_conv_embed = nn.Sequential(
```

```
            Rearrange('b c h w n-> b (h w n) c',
h=(image_size + 1) // 2, w=(image_size + 1) // 2),
            nn.LayerNorm(dim)
        )
        self.stage3_transformer = nn.Sequential(
            Transformer(dim=dim, img_size=(image_size + 1)
// 2, depth=depth[2],
                        heads=heads[2],
channel_depth=(channels + 1) // 2, dim_head=self.dim,
                        dropout=dropout, last_stage=True,
mode=mode),
        )

        self.cls_token = nn.Parameter(torch.randn(1, 1,
dim))
        self.dropout_large = nn.Dropout(emb_dropout)

        self.mlp_head = nn.Sequential(
            nn.LayerNorm(dim),
            nn.Linear(dim, num_classes)
        )

    def forward(self, img):
        conv1 = self.conv1(img)
        xs_trans1 = self.stage1_transformer(conv1)

        conv2 = self.conv2(xs_trans1)
        xs_trans2 = self.stage2_transformer(conv2)

        conv3 = self.conv3(xs_trans2)
        xs_conv3 = self.stage3_conv_embed(conv3)
        b, n, _ = xs_conv3.shape
```

```
    cls_tokens = repeat(self.cls_token, '() n d -> b n
d', b=b)
    xs_cat = torch.cat((cls_tokens, xs_conv3), dim=1)
    xs_drop = self.dropout_large(xs_cat)
    xs_trans3 = self.stage3_transformer(xs_drop)
    xs = xs_trans3.mean(dim=1) if self.pool == 'mean'
else xs_trans3[:, 0]

    xs = self.mlp_head(xs)
    return xs

print("--------------------dataset={0}---model={1}----m
ode={2}------------------".format(args.dataset,
args.model, args.mode))

label_train_loader = Data.DataLoader(Label_train,
batch_size=32, shuffle=True)
label_test_loader = Data.DataLoader(Label_test,
batch_size=32, shuffle=True)

# img = torch.ones([1, 3, 224, 224]).to(device)
#
-------------------------------------------------------
--------------
model = SpectralCA_MDvT(image_size=args.patches,
channels=band, num_classes=args.num_classes,
mode=args.mode).cuda()

parameters = filter(lambda p: p.requires_grad,
model.parameters())
parameters = sum([np.prod(p.size()) for p in
parameters]) / 1_000_000
print('Trainable Parameters: %.3fM' % parameters)
```

```
criterion = nn.CrossEntropyLoss().cuda()
# optimizer
optimizer = torch.optim.AdamW(model.parameters(),
lr=args.learning_rate)
# optimizer = torch.optim.Adam(model.parameters(),
lr=args.learning_rate, weight_decay=args.weight_decay)
scheduler = torch.optim.lr_scheduler.StepLR(optimizer,
step_size=args.epoches // 10, gamma=args.gamma)
tic = time.time()
# include train and test
if args.flag_test == 'train and test':
    tic = time.time()

    train_acc_list = []
    valida_acc_list = []
    train_loss_list = []
    valida_loss_list = []

    valida_epochs_list = []

    for epoch in range(args.epoches):

    # train modeltrain
    model.train()
    train_acc, train_obj, tar_t, pre_t =
train_epoch(model, label_train_loader, criterion,
optimizer)
    scheduler.step()

    print("Epoch: {:03d} train_loss: {:.4f} train_acc:
{:.4f}"
            .format(epoch + 1, train_obj, train_acc))
```

```
    if ((epoch + 1) % args.test_freq == 0) or (epoch ==
0):
        model.eval()
        with torch.no_grad():
            test_acc2, test_obj2, tar_v, pre_v =
test_epoch(model, label_test_loader, criterion,
optimizer)
            OA2, AA_mean2, Kappa2, AA2, matrix,
classification = output_metric(tar_v, pre_v)
            print("Epoch: {:03d} test_loss: {:.4f}
test_acc: {:.4f} OA2: {:.4f} AA_mean2: {:.4f} Kappa2:
{:.4f}"
                  .format(epoch + 1, test_obj2,
test_acc2, OA2, AA_mean2, Kappa2))

            valida_acc_list.append(test_acc2)
            valida_loss_list.append(test_obj2)

            valida_epochs_list.append(epoch + 1)

    train_acc_list.append(train_acc)
    train_loss_list.append(train_obj)

    toc = time.time()
    torch.save(model.state_dict(), model_path)
    print("Running Time: {:.2f}".format(toc - tic))

print("*************************************************
*")
    print("Final result:")
    print("OA: {:.4f} | AA: {:.4f} | Kappa:
{:.4f}".format(OA2, AA_mean2, Kappa2))
```

```
    print(AA2)
    print()

    with open(file_name, 'w') as x_file:
    x_file.write('{:.4f} Test
loss(%)'.format(test_obj2.cpu().numpy()))
    x_file.write('\n')
    x_file.write('{:.4f} Test accuracy
(%)'.format(test_acc2.cpu().numpy()))
    x_file.write('\n')
    x_file.write('\n')
    x_file.write('{:.4f} Kappa accuracy
(%)'.format(Kappa2))
    x_file.write('\n')
    x_file.write('{:.4f} Overall accuracy
(%)'.format(OA2))
    x_file.write('\n')
    x_file.write('{:.4f} Average accuracy
(%)'.format(AA_mean2))
    x_file.write('\n')
    x_file.write('\n')
    x_file.write('{} classification
(%)'.format(classification))
    x_file.write('\n')
    x_file.write('{} matrix (%)'.format(matrix))
    x_file.write('\n')
    x_file.write('\n')
    x_file.write('{:.4f} time (%)'.format(toc - tic))
    print("-------------txt finish------------------")
else :
    print("Shape of out :")  # [B, num_classes]
```

```
train_acc_list = [x.cpu().item() if torch.is_tensor(x)
else x for x in train_acc_list]
train_loss_list = [x.cpu().item() if torch.is_tensor(x)
else x for x in train_loss_list]
valida_acc_list = [x.cpu().item() if torch.is_tensor(x)
else x for x in valida_acc_list]
valida_loss_list = [x.cpu().item() if torch.is_tensor(x)
else x for x in valida_loss_list]

draw_acc_loss(train_acc_list, train_loss_list,
valida_acc_list, valida_loss_list, valida_epochs_list,
args.epoches, "content")


model = SpectralCA_MDvT(image_size=args.patches,
channels=band, num_classes=args.num_classes,
mode=args.mode).to(device)
# model.load_state_dict(torch.load(model_path))
model.load_state_dict(torch.load("/content/SpectralCA-ba
se.pt"))


test300_meta =
gdal.Open("/content/drive/MyDrive/WHU-Hi-HongHu/Training
samples and test samples/Test300.tif").ReadAsArray()

test300_coords = np.argwhere(test300_meta > 0)  # (N, 2)
train_coord_set = set(map(tuple, total_train))
filtered_unlabeled = [tuple(coord) for coord in
test300_coords if tuple(coord) not in train_coord_set]
filtered_unlabeled = np.array(filtered_unlabeled)

import random

subset_size = 5000  # tune this as RAM allows
```

```
subset_coords = random.sample(list(map(tuple,
filtered_unlabeled)), subset_size)
filtered_unlabeled = np.array(subset_coords)

print(f"Pseudo-label candidate count:
{len(filtered_unlabeled)}")

def extract_in_batches(coords, batch_size=10000):
    all_x = []
    for i in range(0, len(coords), batch_size):
    batch = coords[i:i+batch_size]
    x_batch = train_and_test_data(mirror_image,
data_hsi.shape[2], batch, patch=args.patches)
    all_x.append(x_batch)
    return np.concatenate(all_x, axis=0)

x_unlabeled_band =
extract_in_batches(filtered_unlabeled, batch_size=5000)

x_unlabeled =
torch.from_numpy(x_unlabeled_band).type(torch.FloatTenso
r).unsqueeze(1).to(device)

unlabeled_dataset =
Data.TensorDataset(x_unlabeled.cpu())
unlabeled_loader = Data.DataLoader(unlabeled_dataset,
batch_size=128, shuffle=False)

model.eval()
all_preds, all_probs = [], []

with torch.no_grad():
    for (xb,) in unlabeled_loader:
```

```
    xb = xb.to(device)
    logits = model(xb)
    probs = F.softmax(logits, dim=1)
    max_probs, preds = torch.max(probs, dim=1)
    all_preds.append(preds.cpu())
    all_probs.append(max_probs.cpu())

y_unlabeled_pred = torch.cat(all_preds)
y_unlabeled_conf = torch.cat(all_probs)

confident_mask = y_unlabeled_conf > 0.9
x_confident = x_unlabeled[confident_mask]
y_confident = y_unlabeled_pred[confident_mask]

Label_proxy = Data.TensorDataset(x_confident.cpu(),
y_confident.cpu())
Label_combined = Data.ConcatDataset([Label_train,
Label_proxy])

print("---------------------dataset={0}---model={1}----m
ode={2}------------------".format(args.dataset,
args.model, args.mode))

label_train_loader = Data.DataLoader(Label_combined,
batch_size=32, shuffle=True)
label_test_loader = Data.DataLoader(Label_test,
batch_size=32, shuffle=True)

# img = torch.ones([1, 3, 224, 224]).to(device)
#
---------------------------------------------------------
--------------
```

```
model = SpectralCA_MDvT(image_size=args.patches,
channels=band, num_classes=args.num_classes,
mode=args.mode).cuda()

parameters = filter(lambda p: p.requires_grad,
model.parameters())
parameters = sum([np.prod(p.size()) for p in
parameters]) / 1_000_000
print('Trainable Parameters: %.3fM' % parameters)

criterion = nn.CrossEntropyLoss().cuda()
# optimizer
optimizer = torch.optim.AdamW(model.parameters(),
lr=args.learning_rate)
# optimizer = torch.optim.Adam(model.parameters(),
lr=args.learning_rate, weight_decay=args.weight_decay)
scheduler = torch.optim.lr_scheduler.StepLR(optimizer,
step_size=args.epoches // 10, gamma=args.gamma)
tic = time.time()
# include train and test
if args.flag_test == 'train and test':
    tic = time.time()

    train_acc_list = []
    valida_acc_list = []
    train_loss_list = []
    valida_loss_list = []
    valida_epochs_list = []

    for epoch in range(args.epoches):

    model.train()
```

```
    train_acc, train_obj, tar_t, pre_t =
train_epoch(model, label_train_loader, criterion,
optimizer)
    scheduler.step()

    print("Epoch: {:03d} train_loss: {:.4f} train_acc:
{:.4f}"
                .format(epoch + 1, train_obj, train_acc))

    if ((epoch + 1) % args.test_freq == 0) or (epoch ==
0):
        model.eval()
        with torch.no_grad():
            test_acc2, test_obj2, tar_v, pre_v =
test_epoch(model, label_test_loader, criterion,
optimizer)
            OA2, AA_mean2, Kappa2, AA2, matrix,
classification = output_metric(tar_v, pre_v)
            print("Epoch: {:03d} test_loss: {:.4f}
test_acc: {:.4f} OA2: {:.4f} AA_mean2: {:.4f} Kappa2:
{:.4f}"
                        .format(epoch + 1, test_obj2,
test_acc2, OA2, AA_mean2, Kappa2))

            valida_acc_list.append(test_acc2)
            valida_loss_list.append(test_obj2)

            valida_epochs_list.append(epoch + 1)

    train_acc_list.append(train_acc)
    train_loss_list.append(train_obj)

    toc = time.time()
```

```
    torch.save(model.state_dict(), model_path + "_ssl")
    print("Running Time: {:.2f}".format(toc - tic))

print("*************************************************
*")
    print("Final result:")
    print("OA: {:.4f} | AA: {:.4f} | Kappa:
{:.4f}".format(OA2, AA_mean2, Kappa2))
    print(AA2)
    print()

    with open(file_name, 'w') as x_file:
    x_file.write('{:.4f} Test
loss(%)'.format(test_obj2.cpu().numpy()))
    x_file.write('\n')
    x_file.write('{:.4f} Test accuracy
(%)'.format(test_acc2.cpu().numpy()))
    x_file.write('\n')
    x_file.write('\n')
    x_file.write('{:.4f} Kappa accuracy
(%)'.format(Kappa2))
    x_file.write('\n')
    x_file.write('{:.4f} Overall accuracy
(%)'.format(OA2))
    x_file.write('\n')
    x_file.write('{:.4f} Average accuracy
(%)'.format(AA_mean2))
    x_file.write('\n')
    x_file.write('\n')
    x_file.write('{} classification
(%)'.format(classification))
    x_file.write('\n')
    x_file.write('{} matrix (%)'.format(matrix))
```

```
    x_file.write('\n')
    x_file.write('\n')
    x_file.write('{:.4f} time (%)'.format(toc - tic))
    print("-------------txt finish-------------------")
else :
    print("Shape of out :")  # [B, num_classes]

train_acc_list = [x.cpu().item() if torch.is_tensor(x)
else x for x in train_acc_list]
train_loss_list = [x.cpu().item() if torch.is_tensor(x)
else x for x in train_loss_list]
valida_acc_list = [x.cpu().item() if torch.is_tensor(x)
else x for x in valida_acc_list]
valida_loss_list = [x.cpu().item() if torch.is_tensor(x)
else x for x in valida_loss_list]
draw_acc_loss(train_acc_list, train_loss_list,
valida_acc_list, valida_loss_list, valida_epochs_list,
args.epoches, "content")

from torchinfo import summary

model = SpectralCA_MDvT(image_size=args.patches,
channels=band, num_classes=args.num_classes,
mode=args.mode).to(device)
model.load_state_dict(torch.load("/content/SpectralCA-ba
se.pt", map_location=torch.device('cpu')))
summary(model)
```

APPENDIX B GIT REPOSITORY OF SPECTRALCA AND SSL

To ensure the reproducibility of the experimental results and provide access to the complete training pipeline, the full project codebase is hosted in a public repository. This includes data preprocessing scripts, the full model architecture, training configurations, and evaluation metrics.

URL: https://github.com/BrovkoD/spectral-cross-attention.