Pawan Kapur

Unsupervised Hyperspectral Image Clustering with Interval Type-2 Uncertainty Fuzzy Modelling Approaches

School of Technology and Innovations
Master's Thesis
Industrial Systems Analytics

Vaasa 2025

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: Pawan Kapur
Title of the thesis: Unsupervised Hyperspectral Image Clustering with Interval Type-2 Uncertainty Fuzzy Modelling Approaches
Degree: Master's of Industrial Management
Discipline: Industrial Systems Analytics
Supervisor: Associate Prof. Amit K. Shukla
Year: 2025 **Pages:** 62

ABSTRACT:

In recent years, hyperspectral imaging (HSI) has proved to be immensely helpful in material identification and environmental monitoring, due to its ability to capture spectral signatures across different wavelength bands. However, due to high dimensionality and variability of spectral signatures, it is challenging to effectively implement unsupervised learning models – especially clustering methods which are primarily used for defining HSI class boundaries.

The study begins by examining various methods that are currently used to process hyperspectral images and label their pixels. Although many techniques such as support vector machines, neural networks, K-means – mitigate issues like noise, redundant bands, limited labels, and some are even able to handle high dimensionality, mixed-pixel ambiguity and spectral variability at class boundaries still pose significant challenges and leave much to be desired. With this thesis research, we attempted to address these limitations, utilizing fuzzy uncertainty-based approaches such as traditional fuzzy sets, type-2 fuzzy sets (T2 FSs) and related Fuzzy C-Means (FCM) clustering. With T2 FSs, type-reduction is performed using two currently known methods: Nie-Tan and Karnik-Mendel, yielding crisp values of the final-cluster centers. Advancing from available unsupervised clustering approach, we introduce an interval type-2 fuzzy c-multiple means (IT2-FCMM). The proposed approach is particularly relevant for applications where class boundaries are in the "grey area"—such as agricultural crop mapping (e.g., distinguishing mixed-crop types, detecting signs of disease in specific patches), mineral exploration (e.g., identifying mixed assemblages where two or more minerals grow together) and environmental change detection (e.g., monitoring deforestation over time, tracking shrinking of wetlands)—since it provides both accurate cluster assignments and a measure of confidence for each pixel. The algorithm has been implemented in Python and applied to publicly available benchmark hyperspectral datasets. Comparative experiments with other methods demonstrate that the proposed method achieves greater robustness to spectral variability and noise than currently known FCM and other centroid-based clustering approaches. Computation has been accelerated via GPU implementation.

KEYWORDS: Hyperspectral Imaging, Fuzzy Sets, Interval Type-2 Fuzzy Sets, Fuzzy C-Multiple-Means, Clustering, Uncertainty modelling.

Contents

Figures

Tables

Algorithms

Abbreviations

AA	Average Accuracy
CNN	Convolutional Neural Network
FCM	Fuzzy C-Means
FCMM	Fuzzy C-Multiple Means
FECBG	Fuzzy Embedded Clustering Based on Bipartite Graph
FOU	Footprint of Uncertainty
FS	Fuzzy Sets
GCN	Graph Convolutional Networks
HSI	Hyperspectral Imaging
HTD	Hyperspectral Target Detection
IT2	Interval Type-2
IT2 FS	Interval Type-2 Fuzzy Sets
IT2 FCM	Interval Type-2 Fuzzy C-Means
IT2-FCMM	Interval Type-2 Fuzzy C-Multiple Means
IT2FNN-GRM	Interval Type-2 Fuzzy Neural Network – Gaussian Regression Model
K-M, KM	Karnik Mendel Type Reduction
K-NN	K-Nearest Neighbors
LDA	Linear Discriminant Analysis
N-T, NT	Nie-Tan Type Reduction
OA	Overall Accuracy
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
RBF	Radial Basis Function
RF	Random Forests
RGB	Red-Green-Blue
S2GCN	Spectral-Spatial Graph Convolutional Networks
SVM	Support Vector Machines
T1	Type-1
T1 FS	Type-1 Fuzzy Sets
T2 FS	Type-2 Fuzzy Sets
ViT	Vision Transformer

1 Introduction

The thesis starts with this section which introduces the concept of Hyperspectral Imaging (HSI), the challenges associated with classification of its data and how uncertainty modelling is done using Fuzzy Sets (FS) in Section 1.1. Building on the need for more robust models, the research objective and questions are then described in Section 1.2. Finally, Section 1.3 describes the methodology that has been adopted to address the questions and the proposal for a new algorithm.

1.1 Background and Motivation

1.1.1 Hyperspectral Imaging

Regarded as one of the most significant discoveries in remote sensing imaging sciences and technological advancements, HSI comprises thousands of hypercubes and hence possesses a large resolution and an enormous amount of embedded information of all kinds such as spectral, spatial and temporal.

A hyperspectral image can be thought of as a three dimensional "data-cube", where two spatial dimensions (X-axis and Y-axis) correspond to the image plane and a third dimension (λ) is comprised of tens or even hundreds of spectral bands. Some of these bands may be narrow or wide. In a standard optical, RGB (Red-Green-Blue) photograph, each pixel carries just three values – red, green and blue. By contrast, an HSI pixel can contain 100-300 contiguous measurements (for example, spanning 400nm to 2,500nm at 5-10 nm intervals). Such a fine resolution enables the user to observe subtle absorption and reflectance features that are unique to every mineral or element – such as chlorophyll absorption near 680nm or water absorption around 970nm – that would otherwise be impossible to detect with only the three RGB bands. Figure 1.1 illustrates a typical visualization of an HSI data cube (FavPNG, n.d., used under license). The $\mathrm{X} - \lambda$ and $\mathrm{Y} - \lambda$ faces show how intensity varies across narrow wavelength bands. Each pixel in the cube carries a full spectrum (the black curve), rather than just the three-color channels.

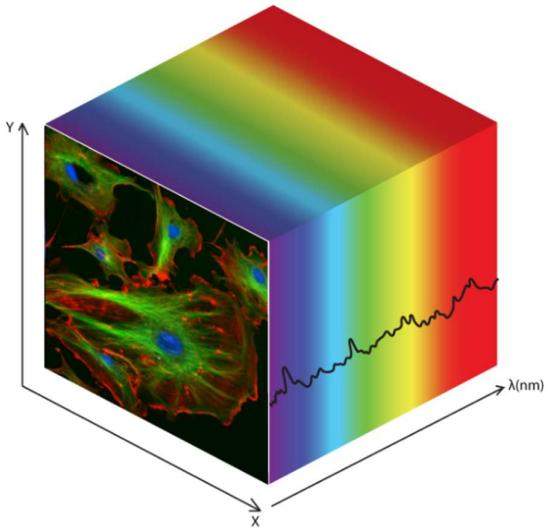

Figure 1.1 An HSI data cube

Now, because each pixel itself is a high-dimensional vector, those vectors can be analyzed to distinguish materials that look identical in the RGB imagery (for instance, two different minerals under identical visible-light conditions). In applications such as precision agriculture (Cetin et al., 2005), environmental monitoring (Huang et al., 2025) and mineral mapping (Li et al., 2018), the ability to "see" these narrow spectral features becomes crucial.

The sheer volume of data makes it infeasible to label every pixel manually. Hence, AI's machine learning methods are often applied to automate extraction of patterns and group similar spectra together. Among such methods is clustering – an unsupervised machine learning method that does not require any labels or trained data and organizes pixels into clusters or groups solely based on their spectral signatures. Because most hyperspectral images do not come with extensive ground-truth labels as acquiring labeled examples in every band cannot be feasible or at the very least, prohibitively costly, the unsupervised methods have proven to be particularly useful in this domain. Unsupervised learning approaches such as K-Means (Ren et al., 2020) and Fuzzy C-Means (FCM) (Mai et al., 2021) are often used as a first step. However, clustering hyperspectral pixels has many challenges:

- **Curse of Dimensionality:** The very large number of spectral features can lead to high estimation error and overfitting in classifiers. Many bands are often redundant or noisy, requiring careful feature selection or reduction (Fırat et al., 2022).
- **Overlapping Class Spectral Variability:** The same land-cover class or material can exhibit significant spectral variation across space and time. Factors like illumination differences, or instrument noise cause spectral variability, i.e., a single class's spectral signature may not be consistent, but rather show diversity and variations within that class. Also, different classes can have very similar spectral signatures in certain bands. For example, two distinct materials may produce overlapping spectral curves, making them hard to distinguish. At image boundaries or mixed pixels, spectral signature mixing occurs, further enhancing interclass similarity and causing classification ambiguities (Wu & Guo, 2021).
- HSIs often contain significant noise due to sensor limitations or atmospheric absorption (e.g., water-absorption bands). Entire bands can be corrupted by noise, and mis-labelled or uncertain ground truth can occur. These bands can further increase uncertainty in the data (Liu et al., 2024).
- This scarcity of training data exacerbates uncertainty, as models must generalize from a few examples.

1.1.2 Methods used for processing Hyperspectral Images

The first question that can be asked is:

Why is clustering needed in HSI?

- Since manual labeling of every pixel in a large HSI is extremely expensive, and most of the time, the ground truth is not available for classification, unsupervised grouping (known as clustering) becomes necessary.
- Many pixels represent mixtures of multiple materials. Therefore, hard assignments cannot express that ambiguity.
- Demands for rapid monitoring (for instance in agriculture, environmental change) make it necessary to develop an unsupervised or minimally supervised technique to handle new datasets.

When working with HSI images, most traditional workflows begin by treating each pixel's full spectrum as a high-dimensional feature vector and applying either supervised classifiers or dimensionality-reduction models. Two types of classifiers are used to classify the pixels in HSI into various classes:

1. Supervised Classifiers

    a. Support Vector Machines (SVM)

        These map pixel spectra into high-dimensional spaces using kernels (e.g., RBF) to separate labeled examples. They are effective with enough labeled data but are prone to noise since they classify each pixel independently and scale poorly as the data size increases.

    b. Random Forests (RF) (Breiman, 2001)

        These are groups of decision trees that can handle high-dimensional inputs and can also rank every band's importance. They ignore spatial context unless it is explicitly added through other methods.

    c. Neural Networks

        Convolutional Neural Networks (CNN) are of two types: 3D-CNNs – these jointly convolve spatial and spectral dimensions, yielding high accuracy but at high memory and computation cost, and 2D-CNNs – these operate on Principal Component Analysis (PCA)-reduced bands (reduced dimensionality), capturing spatial features but with fewer resources. GCNs (Graph Convolutional Networks) treat pixels as graph nodes, summing the information from spectrally and/or spatially similar neighbors. Transformers use self-attention across patches to model long-range dependencies but require many labels or pretraining.

    All these supervised methods perform well when large, accurate label sets exist but struggle when labels are scarce or when mixed-pixel regions increase ambiguity.

2. Unsupervised Hard Clustering
    a. K-Means (Hartigan & Wong, 1979)

        As one of the oldest unsupervised techniques, this method assigns each pixel to the nearest centroid in the spectral space. It is fast but enforces hard assignments, does not use spatial information and finds it difficult to handle ambiguity or when a pixel belongs to two or more classes, resulting in it often misclassifying those mixed pixels and, hence producing noisy maps.

    b. PCA & Clustering

        PCA or similar transform methods are first applied to compress the hundreds of bands into a few principal components (for instance the top 10 or 20 bands), then a hard clustering algorithm (K-Means or affinity propagation) groups the reduced pixels. While this mitigates the “curse of dimensionality”, it still forces each pixel into exactly one cluster and cannot express spectral boundary uncertainty.

Because acquiring dense ground truth is often impossible and many pixels genuinely lie between classes, hard assignments become an insufficient solution. Methods are required that can: 1. Work with scarce labels and automatically discover natural groupings without labels and 2. Are able to handle mixed-pixel ambiguity.

Here, another question arises:

Why then choose Fuzzy Clustering for HSI?

Fuzzy clustering assigns each pixel a degree of membership to multiple clusters, reflecting real mixed-pixel conditions (Madhu et al., 2021) and spectral variability (Chen et al., 2023)

- A pixel whose spectrum lies between vegetation and soil can be “60% vegetation and 40% soil” rather than forced into one label.
- Even within a single class, factors like illumination or moisture can create variation. Fuzzy memberships capture that continuous spread rather than imposing sharp boundaries.

- By distributing a pixel's membership and hence its influence on multiple clusters, fuzzy methods reduce the impact of noise on estimation of the centroid.

Fuzzy clustering is therefore crucial to group hyperspectral image pixels without labels and to show the actual reality of mixed pixel while also allowing uncertainty in segmentation.

1.1.3 Uncertainty Modelling with Fuzzy Sets

Hyperspectral images inherently contain uncertainties stemming from mixed-pixel effects (where a single pixel may represent multiple materials), sensor noise, atmospheric interference, and natural spectral variability within a class. Rather than forcing each pixel into a single "hard" category, fuzzy-set theory allows pixels to belong to multiple classes with varying degree of memberships. In a broad sense, the progression of how fuzzy-set methods have been used for modeling HSI uncertainty can be summarized as follows:

1.1.3.1 Fuzzy Sets

Fuzzy Logic (Zadeh, 1965) is about quantifying vague concepts. In traditional logic, statements can only be true (1) or false (0). But concepts like human emotions or assumptions are rarely so black-and-white. For instance, in astronomy, for judging if a particular exoplanet is habitable or not, multiple parameters depend on it – temperature, atmosphere composition, distance from its host star, gravity and so on. From estimates the exoplanet may be judged "70% habitable". This is where fuzzy logic comes in. For fuzzy logic, the truth does not just lie in either 0 or 1, but also in between 0 and 1. So a particular concept or emotion may be 40% right and 60% wrong. In other words, it is not asked whether the assumption about the exoplanet being habitable is true or false, but rather how true it is.

To express the truth value of an assumption, membership functions are used. A fuzzy set A – a mathematical notation of fuzzy logic – (or Type-1 Fuzzy Set (T1 FS)) on universe U is characterized by a membership function as follows:

$$\mu_A : U \longrightarrow [0,1] \tag{1.1}$$

Where each element $x \in U$, $\mu_A(x) \in [0,1]$ denotes a crisp membership grade to which x belongs to A (Zadeh, 1965). The fuzzy set A is then written as:

$$A = \{ (x, \mu_A(x)) \mid x \in U \} \tag{1.2}$$

where $\mu_A : U \to [0,1]$.

$\mu_A(x) = 0$ corresponds to non-membership value, $\mu_A(x) = 1$ to full membership, and the intermediate values are crisp partial membership grades. (Zadeh, 1965).

In HSI image clustering, T1 FS allow each pixel to have a graded membership $\mu(x) \in [0,1]$ but still assign a crisp value that cannot capture the full uncertainty of that pixel arising from spectral variability, mixed-pixel effects and noise – particularly for pixels near class boundaries. As a result, T1 FS clustering methods may yield overconfident assignments at class boundaries or prove insufficient to fully model the Footprint of Uncertainty (FOU) inherent in mixed pixels. This motivates the move to Type-2 Fuzzy Sets (T2 FS) in which each membership grade $\mu(x)$ itself becomes an interval that can capture more ambiguity (Nie & Tan, 2008).

1.1.3.2 Type-2 Fuzzy Sets (T2 FS) and Interval Type-2 Fuzzy Sets (IT2 FS)

General Type-2 Fuzzy Sets

T2 FSs extend the concept of T1 FSs by allowing each pixel's membership to itself be 'fuzzy'. A T2 FS A over a universe X is characterized by a membership function as follows:

$$\mu_A : X \times [0,1] \to [0,1] \tag{1.3}$$

where,

- $x \in X$ is the primary variable (for instance a hyperspectral pixel).
- $u \in [0,1]$ is the secondary variable (a membership grade).
- $\mu_A(x,u)$ is the degree to which u is a possible membership value for the pixel x.

So instead of mapping x to a single $\mu(x)$, a T2 FS maps x to a FS having value between $[0,1]$. The uncertainty in the primary membership of a T2 FS, consists of the bounded region, the FOU. It is the union of all primary memberships (Hagras, 2005; Mendel & John, 2002) and is written as:

$$\mathrm{FOU(A)} = \bigcup_{x \in X} \{(x,u) | u \in J_x \subseteq [0,1]\} \tag{1.4}$$

where J_x is the support of the secondary membership function at x (Mendel & John, 2002).

In hyperspectral images, a pixel x's spectrum can be highly ambiguous due to subpixel mixtures, sensor noise, and atmospheric effects. A T2 FS makes it possible to model not only ***which classes*** x might belong to (first-order uncertainty) but also ***how uncertain*** that membership is (second-order uncertainty). For instance, a pixel x whose observed spectrum sits roughly halfway between two vegetation subtypes might have a primary membership "fuzzy slice" $\mu_A(x,u)$ that peaks around $u = 0.5$ but extends significantly across $[0.3,0.7]$, showing uncertainty in the confidence.

Consider three adjacent pixels x_1, x_2, x_3 in a hyperspectral image in a vegetation–soil boundary:

- Under T1 FS, each pixel might receive memberships $\mu(x_1) = 0.2$ (mostly soil), $\mu(x_2) = 0.5$ (ambiguous), and $\mu(x_3) = 0.8$ (mostly vegetation). These single values may hide uncertainty about whether $\mu(x_2)$ should be 0.4 or 0.6 if the sensor's noise shifts its observed spectrum slightly.
- Under T2 FS, pixel x_2 could have a secondary membership distribution $\mu_A(x_2,u)$ that peaks around $u = 0.5$ but has nonzero support over $[0.3,0.7]$. If subsequent processing detects unusual noise, $\mu_A(x_2,\cdot)$ could broaden to $[0.2,0.8]$ reflecting higher uncertainty.

Such flexibility is crucial when classification decisions must account for unpredictable noise or gradual transitions which is common in real-world remote sensing.

Interval Type-2 Fuzzy Sets (IT2 FSs)

An IT2 FS model the second-order uncertainty – arising from spectral variability, mixed pixels at class boundaries and noise – by assigning each element x a range of membership values $\left[\underline{\mu}(x), \overline{\mu}(x)\right]$ rather than a single number. The union of these intervals across the image pixels constitutes the FOU and written as follows:

$$\mathrm{FOU(A)} = \bigcup_{x \in X} \left[\underline{\mu}_A(x), \overline{\mu}_A(x)\right] \tag{1.5}$$

which captures the variability of spectral signatures and sensor noise in HSIs. This interval representation is computationally efficient, as all secondary memberships within the interval are set to 1, and 0 outside (Mendel & John, 2002).

An IT2 FS A on universe X is written as:

$$A = \left\{\left(x,\ \left[\underline{\mu}_A(x), \overline{\mu}_A(x)\right]\right) \,\middle|\, x \in X\right\}, \quad \underline{\mu}_A(x) \leq \overline{\mu}_A(x) \tag{1.6}$$

where $\underline{\mu}_A(x)$ and $\overline{\mu}_A(x)$ are the lower and upper membership bounds for element x respectively.

The process to convert intervals to a single ("crisp") value is termed as Defuzzification or "type-reduction" and often obtained from the midpoint calculated as follows:

$$\frac{1}{2}\left(\underline{\mu}_A(x) + \overline{\mu}_A(x)\right) \tag{1.7}$$

or applying more sophisticated interval-aggregation rules, such as Nie-Tan (NT) (Nie & Tan, 2008) or Karnik-Mendel (KM) (Karnik & Mendel, 2001).

Key benefits of IT2 for HSI clustering include:

- Pixels deviating from class boundaries due to illumination or sensor drift can still be correctly grouped within their respective FOU (Mendel et al., 2002).
- In overlapping-class regions, IT2 assigns an interval that straddles multiple classes, reducing misclassification of mixed pixels (Wu & Guo, 2021).
- Outlier pixels receive a broad membership interval which reduces their effect on estimation of centroid, yielding more accurate clustering (Mendel et al., 2002).

Interval memberships $\left[\underline{\mu}_A(x), \overline{\mu}_A(x)\right]$ are then collapsed to crisp values using established type-reduction methods – specifically, the Nie-Tan (NT) (Nie & Tan, 2008) and Karnik-Mendel (KM) (Karnik & Mendel, 2001) algorithms – before centroid updates and final membership assignments.

1.2 Research Question and objectives

Although the existing HSI clustering approaches address many of the issues highlighted in this section, several challenges remain, such as mixed-pixel ambiguity and multimodal spectral variability. Every pixel in HSI usually covers large areas on the ground. Thus, it is very possible that a single pixel is a mixture of ground materials. These mixed pixels have their spectral bands display a mixture of signatures from those multiple materials rather than a single, pure material signature which would have made it easy to classify that pixel. Their existence reduces the accuracy of recognition and classification of ground objects and materials. In clustering, this ambiguity means that pixels lying at class boundaries may not belong to any one single cluster, as their spectra has characteristics of two or more materials. Traditional hard-clustering methods often misclassify such mixed pixels, producing noisy class boundaries unless an unmixing or fuzzy approach is used to capture partial memberships (Liu et al., 2009).

Coming to multimodal spectral variability, HSI clusters, due to mixed pixels, frequently exhibit multiple density modes within a single land-cover class due to:

- Variations in material composition (e.g., sunlit vs. shaded canopies) creating distinct spectral sub-categories.
- Illumination angles, atmospheric scattering, and seasonal changes cause nonlinear intensity shifts in reflectance spectra.
- Mixed Pixels that often contain mixtures of materials with varying abundances (e.g., soil-vegetation composites), leading to multimodal distributions in the feature space.

Using classical clustering algorithms (e.g., K-Means) which assume single separate clusters causes two key issues:

- Distinct subclasses get erroneously grouped into one cluster causing over-merging.
- Subtle transitions between the materials lack clear separability in Euclidean space causing ambiguity in boundaries.

The objective of this study is to propose a novel, robust and efficient approach that addresses these challenges through multistage assignments and demonstrates improvements over current T1 FSs and other clustering methods.

Through this objective the thesis attempts to answer the following questions:

1. How does the performance of the proposed method improve as compared to traditional FCM method?
2. Do type-reductions methods influence the proposed method's performance?

1.3 Research Methodology

The following steps outline the approach of the thesis. It follows a quantitative, experimental approach within the framework of a literature-based review:

i. Literature review of existing HSI clustering methods

A concise study is presented of traditional and recent HSI clustering approaches (e.g. T1, FCM, FCMM) highlighting their methodology, advantages and shortcomings in terms of handling mixed pixels or HSI images with high noise.

ii. Mathematical formulation of the proposed IT2-FCMM Model

The objective functions of IT2-FCMM are derived including:

- Pixel-to-subcluster membership and subcluster centroid update formulas.
- Subcluster-to-final cluster membership and final cluster centroid update formulas.

iii. Algorithm Design

An IT2-FCMM algorithm is designed, detailing:

- Initialization procedure.
- Update steps for memberships and centroids.
- Checking for convergence.

The algorithm is implemented using Python – using PyTorch for leveraging GPU-accelerated processing.

iv. Experimental evaluation

The proposed algorithm is applied to benchmark hyperspectral datasets (e.g. Indian Pines, Pavia University). For each dataset:

- Ground truth labels are used for unsupervised accuracy assessment.
- Overall Accuracy (OA), Average Accuracy (AA) and Kappa coefficient are computed.

v. Comparison and analysis

The performance metrics are then tabulated and compared against existing methods (T1 FCM). The results are analyzed statistically to determine the significance of the improvements. The results are also visualized along with the ground truth for comparison discussion.

2 Literature Review

In this section, theoretical foundations are explained that form the basis of a proposal in Section 3. Section 2.1 describes various non-fuzzy methods have been used for processing hyperspectral images. Section 2.2 then describes how interval type-2 fuzzy methods and its variants have been used to address mixed-pixel ambiguity in HSIs.

2.1 Non-Fuzzy HSI Processing Methods

1. **Support Vector Machines (SVM)**

   Support Vector Machines (SVMs) have always been crucial in hyperspectral image classification due to their effectiveness in high-dimensional feature spaces. An SVM seeks a maximum-margin hyperplane to separate labeled examples (x_i, y_i), where each $x_i \in \mathbb{R}^d$ is the pixel spectrum and y_i its class label. SVMs are robust to overfitting when using kernel functions (e.g., RBF) that implicitly map data to feature spaces of very high (even infinite) dimensionality.

   In terms of applications, Melgani & Bruzzone (2004) actually pioneered the SVM method for classification of hyperspectral images. They used a Gaussian (RBF) kernel to map the high-dimensional pixel spectra into a feature space and achieved greater than 90% overall accuracy on the Indian Pines HIS dataset.

   Although SVMs can handle high-dimensional data, are robust to overfitting while producing clear decision boundaries, they still have a few limitations. Achieving maximum performance from the SVMs requires careful tuning of hyperparameters C (penalty) and σ (the kernel width). Also, since they do not take into consideration the spatial context (the adjacent region of the pixel), they end up classifying each pixel independently which often leads to “salt-and-pepper” noise. Their computational cost also scales in the order of $O(n^3)$, hindering their usage in large scale HIS imagery.

2. Random Forests (RF)

Originally introduced by Breiman (2001), RFs construct an ensemble of decision trees, grown on random subsets of training data and features. Each tree votes on class assignment and the majority vote decides which label does the datapoint get assigned to. RF has been applied successfully to HSI classification due to its innate ability to handle high-dimensional data and to provide estimates of which features are of high importance.

Wang & Wang (2022) extended classical RFs by incorporating both spatial context and spectral-temporal information to tackle thick-cloud removal in HSIs. Their Fast Spatial-Spectral Random Forests algorithm first applies PCA to compress temporally neighboring HSIs into a small set of principal components. Then it trains RF models using local 3×3 spatial patches extracted from those components to predict missing reflectance in cloudy pixels. By combining multiple compressed bands with spatial neighborhoods, the algorithm achieved accuracy comparable to a full-band Spatial-Spectral Random Forest model but ran over six times faster, demonstrating RF's flexibility when augmented with PCA-driven dimensionality reduction and explicit spatial features (Wang & Wang, 2022).

RFs treat each pixel independently; spatial neighborhood is ignored without explicit spatial features. RF also suffers when the number of training labels is very small (< 50 per class). Depth and number of trees must be tuned; overly large forests increase inference time.

3. K-Nearest Neighbors (K-NN)

In K-NN each unlabeled pixel is assigned the most common class among its k closest labeled neighbors in spectral space. In K-NN no separate training model is computed except storing the labeled samples. However, because distance is measured in the high-dimensional spectral space, noisy or redundant bands can affect negatively on how similar a pixel is to its nearest neighbor.

Huang et al. (2015) used K-NN not as a standalone classifier but as a nonlocal filtering step to refine SVM-generated probability maps. After an initial pixel-wise

SVM classification produces class probability maps, they defined a joint feature vector for each pixel. Then, for each pixel they found its K-nearest neighbors in this combined spectral spatial space and averaged the SVM probabilities over those neighbors. By doing so, they transferred spatial structure from the first PCA component (used as the “guidance image”) into the probability maps, producing smoother, more accurate classification without costly segmentation or global optimization (Huang et al., 2015).

4. Convolutional Neural Networks (CNNs)

In recent years, deep-learning models have been used extensively in many applications. In HSI image processing, many forms of neural networks are employed.

Spectral–spatial CNNs exploit both the detailed spectral signatures and the local spatial context inherent in hyperspectral images. In a 3D-CNN, each input sample is considered as a small cube (for instance a 15×15 pixels × 100 bands) so that convolutional filters span both spatial (width/height) and spectral (band) dimensions. This joint processing allows complex features to be learnt and capture how material spectra vary locally across neighboring pixels. Although this typically yields a very high accuracy, it comes at a cost of large memory and computation time. In contrast, a 2D-CNN operates on dimensionally reduced images (which have undergone PCA or other band reduction methods), capturing spatial patterns across those principal components. While 2D-CNN is less resource intensive than a full-fledged 3D-CNN, a 2D-CNN can still capture valuable spatial information once the most informative spectral information has been filtered.

Fırat et al. (2022) show that combining a suitable dimension reduction method (e.g., PCA) with a lightweight hybrid 3D/2D architecture, where the initial 3D convolutions learn spectral and spatial features and the subsequent 2D architecture and depth-wise separable convolutions refine the spatial detail resulting in almost perfect classification, $> 99.8\%$ OA on Indian Pines and Pavia University datasets, with far fewer parameters than a pure 3D model. Chen et al. (2024) extend this idea by adding a hyperspectral target-detection (HTD) branch

where one stream uses HTD plus a 2D-CNN (with deconvolution) to highlight class specific spectral signatures, while a parallel 3D-CNN processes PCA-compressed bands. Merging these streams further improves accuracy of the small classes, thereby increasing the OA approximately to 98.4% in Indian Pines and 99.9 % in Pavia University.

5. Graph Convolutional Networks (GCNs)

GCNs extend standard convolution to arbitrarily structured data by treating pixels as nodes in a graph, where edges encode similarity (e.g., spectral distance) or spatial adjacency. Instead of sliding a fixed shape kernel over a grid, a GCN sums up all the information from each node's neighbors. That is at each layer, a node's new feature is computed by taking a weighted combination (often a normalized sum) of its own feature and those of adjacent nodes, followed by linear transformation. This allows GCNs to capture both long-range spectral relationships (nodes far apart in the image but similar in spectrum) and local spatial structure (information on adjacent pixels) in one unified method.

Qin et al. (2019) proposed a Spectral-Spatial Graph Convolutional Networks (S2GCN) that fuses spectral signatures and local pixel neighborhoods into a single graph-convolution step. The authors built a graph whose nodes are pixels, the edges encode spectral similarity, and then they augment each edge weight by a Gaussian-based spatial distance to ensure that only physically adjacent pixels remain strongly connected. In each graph-convolutional layer, a node's updated feature is the sum of its own and its neighbor's features (weighted by both spectral and spatial affinities), passed through a learnable linear projection and ReLU. By combining spectral adjacency with spatial context, S2GCN markedly outperforms a conventional GCN, boosting overall accuracy in Indian Pines from approximately 66.8% (pure GCN) to approximately 91.6%. And it shows similar large gains on Botswana Hyperion and Kennedy Space Center hyperspectral image data (Qin et al., 2019).

Liu et al. (2024) developed a Spectral-Spatial GCN with Dynamic-Synchronized Multiscale Features for few-shot HSI classification. First, they generated multiscale

patches (for instance 9 × 9 up to 25 × 25 pixels) around each labeled pixel. Within each scale, a small CNN branch “weighted spectral optimization” was applied to re-weight the bands by their discriminative power. Then two parallel branches extract the local spectral-spatial features via standard convolutions and an adaptive dynamic GCN, where adjacency is defined by the Mahalanobis distance rather than a fixed threshold, thus allowing the graph to be relearned for each patch. To join the information across different scales, an LSTM processes the multiscale feature sequence, and auxiliary classifiers at each scale regularize the training data. On Indian Pines, Pavia University and Salinas, this dynamic GCN achieved 87.25 %, 92.72 %, and 93.36 % OA respectively under extreme few-shot settings, demonstrating that dynamically learned graphs plus multiscale spectral-spatial features give the most robust outputs even with very limited labels (Liu et al., 2024).

6. Transformers

Transformers rely entirely on self-attention to model long range dependencies across input tokens. In a typical Vision Transformer (ViT) for HSI classification, the image cube is first split into nonoverlapping patches (for instance, 4 × 4 pixels across all spectral bands). Each patch is flattened and projected onto an embedding vector, then augmented with a positional encoding so the model knows where it came from. Stacked layers of self-attention compute pairwise affinities among all patch embeddings, enabling the network to learn which spectral-spatial regions influence one another the most. This expansive field helps distinguishing materials whose spectral signatures may be separated by many pixels but share small similarities. Compared to 3D CNNs, which use fixed size kernels over small cubes, transformers dynamically learn attention weights, making them particularly adept at modeling nonlocal spectral-spatial patterns. Their main limitation is about how self-attention scales quadratically with the number of patches and they often require either large amounts of labeled data or effective training to avoid overfitting.

Wang et al. (2022) proposed a Spectral-Spatial-Temporal Transformer (SST-Former) for change detection. SST-Former first processes each HSI cube through a Spectral-Spatial Transformer (SS-Former) branch, two spectral transformer encoders extract the band sequence features per patch, followed by three spatial transformer encoders that capture neighborhood textures. A learnable “class token” sums up these spectral–spatial features. Parallely, a second SS-Former (with shared weights) processes the co-registered HSI at the next time step. Their outputs (class tokens plus patch tokens) then feed a Temporal Transformer (T-Former) to learn change related features between the two class tokens and their patch embeddings. By jointly modeling spectral sequences, spatial textures, and temporal evolution, SST-Former achieved a state-of-the-art change detection (approximately 91.6% OA in Indian Pines) with very limited training samples (Wang et al., 2022).

Wang et al. (2025) introduced a Masked Vision Transformer (MViT) for fast HSI classification. MViT first applies PCA to reduce bands to a fixed number (for instance 30 bands) and then extracts a 15 × 15 spatial cube around each pixel. During training, it uses a small 3D convolution layer followed by a 2D convolution layer to produce spectral-spatial embeddings, then randomly masks 75 % of these embeddings, while at the same time, always preserving the central pixel’s embedding to force the model to learn robust “patterned features” rather than just discriminative ones. These retained embeddings (plus a learnable class token) are fed into three ViT encoder blocks. During the test time, no masking occurs and all embeddings are used for final classification. By combining masking-driven robustness with the self-attention mechanism, MViT achieved high accuracy on multiple benchmark datasets while remaining extremely lightweight and training much faster than conventional ViTs (Wang et al., 2025).

2.2 Fuzzy HSI Processing Methods

2.2.1 Interval Type-2 Fuzzy Methods

1. **Foundational IT2 FCM (Hwang & Rhee, 2007)**

   Hwang and Rhee (2007) extend classical FCM (Bezdel et al., 1984) by modelling uncertainty in the fuzzifier parameter via an Interval Type-2 (IT2) fuzzy set. Instead of using a single fuzzifier m, they introduce two fuzzifiers $m_1 < m_2$ which generate a Footprint of Uncertainty (FOU) for each pixel's membership to a cluster. Then, for each pixel x_i and cluster j, they compute a lower membership $\underline{u}_{ij}$ using m_2(a bigger fuzzifier value means more diffuse the boundary is, and more diffused the assignments will be) and an upper membership $\overline{u}_{ij}$ using m_1 (a smaller fuzzifier value means, sharper the boundary is and sharper the assignments will be). The resulting interval $\left[\underline{u}_{ij}, \overline{u}_{ij}\right]$ forms an IT2 fuzzy membership. To update the cluster centers, they apply the Karnik–Mendel (KM) (Karnik & Mendel, 2001) iterative type-reduction to each interval set, sorting distances and iteratively computing the exact lower and upper centroids. Then take the midpoint as the crisp centroid. In the end, hard assignments are obtained by defuzzifying the IT2 memberships before assigning each pixel to the cluster with the highest reduced membership. By managing fuzzifier uncertainty explicitly, IT2 FCM has proved its ability to adaptively widen or narrow each cluster's "maximum fuzzy region," yielding more robust clustering when clusters differ in size or density. (Hwang & Rhee, 2007).

2. **Single-Fuzzifier IT2-FLICM for Robust Land-Cover Segmentation**

   Wu & Guo (2021) extend Interval Type-2 FCM by first representing each pixel not as a single point but as a small interval that reflects local spectral variation and noise. Then they introduce the concept of "local information" – at each iteration, a pixel's membership is also influenced by its neighbors, encouraging spatially coherent clusters. By using only one fuzzy exponent, they simplify the usual two-parameter IT2 approach and are still able to capture the interval uncertainty

associated with each class assignment. This Single-Fuzzifier IT2 Fuzzy Local C-Means method produces noticeably smoother and more accurate land-cover maps across several high-resolution datasets (e.g., Salinas, PaviaU) compared to classic FCM or standard IT2 FCM. Noise and mixed-pixel errors are significantly reduced, while the runtime remains practical for large images.

However, introducing local information as weight and interval width parameter requires a lot of tuning as incorrect values can lead to over or under smoothing. Computing intervals for each pixel based on the neighborhood and performing type-reduction—even via a single fuzzifier—adds runtime compared to classic FCM. A strong neighborhood influence can blur fine spatial details and potentially merge small but distinct land-cover features. Using a fixed 3×3 window may not capture larger spatial context or adapt to varying spatial resolutions as found in real-world data, reducing flexibility of the method and its ability to be adapted across different sensors or datasets.

3. Hybrid IT2 Semi-Supervised Possibilistic FCM with PSO for Satellite Imagery (IT2SPFCM-PSO)

Mai et al. (2021) propose a clustering framework that integrates multiple fuzzy concepts and optimization techniques to tackle noise, uncertainty, and scarce labels in satellite images. First, they use Interval Type-2 fuzzy sets to assign each pixel an interval membership, capturing uncertain boundaries more flexibly than Type-1 approaches. Second, a possibilistic C-Means term is introduced which is a “typicality” measure to reduce the influence outlier pixels exert on centroids. Third, a semi-supervised constraint incorporates a small set of labeled pixels, guiding the cluster centroids towards known land covers without requiring full supervision. Finally, Particle Swarm Optimization (PSO) automatically tunes key parameters – cluster centers, and fuzzifier exponents and avoids manual initialization that often plagues fuzzy methods. By combining interval uncertainty, possibilistic robustness, sparse labels, and automatic parameter search, the method delivers reliable land-cover maps even under challenging conditions in which the imagery was acquired.

However, including PSO adds significant runtime since many particles must fully evaluate the fuzzy objective repeatedly and moreover the PSO swarm settings require tuning to achieve the desired results. So even though parameter search became automated, the swarm setting is manual and hence needs careful tuning. By introducing semi-supervised constraint, dependency is increased upon accurately labelled samples as mislabeled pixels can cause incorrect clustering. Also, this approach does not consider spatial context which also affects the class boundary smoothness.

4. Interval Type-2 Fuzzy Neural Network – Gaussian Regression Model (IT2FNN-GRM) for High Resolution Land-Cover Classification

Wang et al. (2022) propose a two-stage classification framework - called IT2FNN-GRM - that combines a Gaussian Regression Model (GRM) with an Interval Type-2 Fuzzy Neural Network (IT2FNN) to address uncertainty and complex spatial patterns in high-resolution land cover imagery. First, they fit a Gaussian regression model to each adjacent same region by using local pixel intensities (and their 3×3 neighborhood) to capture how grey-level distributions deviate from a single-peak assumption. This GRM serves as a base of “fuzzy membership” function that reflects each pixel’s likelihood of belonging to a given land-cover class. In the second stage, they feed the resulting upper/lower interval memberships (derived from the GRM) into an IT2FNN, which further refines class assignments by processing both interval boundaries and neighborhood relationships via a fuzzy-neural architecture. As a result, IT2FNN-GRM can express both pixel-level uncertainty (through interval memberships) and local spatial coherence (by incorporating neighbors’ membership degrees) before making a final classification decision.

In terms of drawbacks, the model’s complexity has grown and includes many hyperparamters (e.g., interval widths, network weights, membership exponents) which require careful tuning. Although faster than a full-fledged IT2 type-reduction at scale, the dual stages (GRM fitting + IT2FNN training) still post higher runtimes

than simpler FCM or classic CNN approaches. Here too relying on a fixed 3x3 window may not generalize well across different resolutions or scales.

Despite the broad array of non-fuzzy and fuzzy approaches reviewed in this section – from SVMs, RFs, and CNNs to foundational and advanced IT2 based clustering – none simultaneously address intra-class variability, mixed-pixel uncertainty, and spatial coherence in a scalable manner. Type-1 methods either ignore second-order ambiguity or require post-hoc smoothing. While full KM-based (Karnik & Mendel, 2001) IT2FCM captures uncertainty with high accuracy, it is extremely computationally expensive on large scale datasets and single-stage IT2 variants lack subcluster modeling to handle multimodal spectral distributions.

Therefore, there remains a clear need for a two-stage model that captures intra-class heterogeneity by introducing subclusters and embeds interval uncertainty with efficient NT (Nie & Tan, 2008) type-reduction to maintain runtime feasibility. Section 3 proposes a solution that attempts to achieve robust, uncertainty-aware scalable clustering.

3 Proposed Unsupervised Hyperspectral Image Clustering

This section builds upon the concepts discussed in the previous sections and tries to address the shortcomings by proposing a novel approach for unsupervised clustering in hyperspectral images. Section 3.1 outlines the parameter notations used by the methods discussed in this section as well as the proposed algorithm. Section 3.2 introduces the concept of clustering in HSI with the Fuzzy C-Means (FCM) method. Section 3.3 builds on the FCM method and describes Fuzzy C-Multiple-Means (FCMM) in detail as first proposed by Yang et al. (2023). In the subsequent sections of Section 3.3 – 3.5, mathematical formulation of the proposed algorithm's objective functions, membership updates, centroid updates and convergence criteria are shown. In Sections 3.6 and 3.7 the proposed algorithm's pseudocode is shown with expected inputs and output along with a detailed explanation of how it works as well as a visual flow of its implementation.

3.1 Notations

Table 3.1 outlines the parameters that have been used throughout this section. For every parameter, it provides its corresponding details and in which approaches (FCM, FCMM and IT2-FCMM) these parameters are used.

3.2 Fuzzy C-Means (FCM) – A Type-1 Clustering Method

Proposed by Dunn (1973), FCM is a soft clustering algorithm that assigns each datapoint x_i a membership $u_{ij} \in [0,1]$ to each of the c clusters, provided the summation of the datapoint's membership to all clusters is 1. i.e., $\sum_{j=1}^{c} u_{ij} = 1$. The algorithm in FCM attempts to minimize the objective function written as:

$$J_{FCM} = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{r} \left\| x_i - v_j \right\|^2 \tag{3.1}$$

where v_j is the centroid of cluster j, and $r > 1$ is the fuzzifier exponent controlling the cluster's softness: larger r $(where, r > 1)$ yields more fuzziness of the cluster boundary

while $r \to 1$ results in boundary becoming sharper. $\left\|x_i - v_j\right\|^2$ is the square of the Euclidean distance of the datapoint from the cluster center (centroid of the cluster).

Table 3.1 Parameters and their description

Parameter	Detail	Used In Approaches
$X = \{x_i \in \mathbb{R}^d\}_{i=1}^n$	Set of n hyperspectral pixels	IT2-FCMM
n	Number of pixels (treated as rows)	FCM, FCMM, IT2-FCMM
d	Number of spectral bands (treated as columns)	FCMM, IT2-FCMM
c	Number of final clusters	FCM, FCMM, IT2-FCMM
q	Number of subclusters	FCMM, IT2-FCMM
r	Fuzzifier parameter	FCM, FCMM
r_1, r_2	Lower and Upper fuzzifier respectively, where $1 < r_1 < r_2$	IT2-FCMM
α	The weighing parameter between the two objective terms where $\alpha > 0$	FCMM, IT2-FCMM
u	Subcluster memberships	FCM, FCMM, IT2-FCMM
$\underline{u} \in \mathbb{R}^{n \times q}$	Pixel → Subcluster lower memberships with entries $\underline{u}_{if}$	IT2-FCMM
$\overline{u} \in \mathbb{R}^{n \times q}$	Pixel → Subcluster upper memberships with entries $\overline{u}_{if}$	IT2-FCMM
z	Final cluster memberships	FCMM, IT2-FCMM
$\underline{z} \in \mathbb{R}^{q \times c}$	Subcluster → final cluster Lower memberships with entries $\underline{z}_{fj}$	IT2-FCMM
$\overline{z} \in \mathbb{R}^{q \times c}$	Subcluster → final cluster Upper memberships with entries $\overline{z}_{fj}$	IT2-FCMM
M	$[m_1; \dots; m_q] \in \mathbb{R}^{q \times d}$ the subcluster centroids	FCMM, IT2-FCMM
V	$[v_1; \dots; v_c] \in \mathbb{R}^{c \times d}$ the final cluster centroids	FCM, FCMM, IT2-FCMM

The Membership value, u_{ij} of the datapoint is updated as follows:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{(r-1)}}} \tag{3.2}$$

where $\|x_i - v_j\|$ is the Euclidean distance of the datapoint x_i from v_j cluster and $\|x_i - v_k\|$ is the Euclidean distance of the datapoint x_i from rest of the v_k clusters. By updating the memberships, the centroid location is updated as follows:

$$v_j = \frac{\sum_{i=1}^{n} u_{ij}^{r} \, x_i}{\sum_{i=1}^{n} u_{ij}^{r}} \tag{3.3}$$

By alternating between membership update and centroid update, iterations are done so that "convergence" is reached. This convergence is defined by a threshold (ε). For instance, if the threshold is set at 0.0005, and the centroid update results in the movement of the centroid by a distance less than the threshold, it is said that convergence has been reached. It is written mathematically as follows:

$$\left\| v_j^t - v_j^{(t-1)} \right\| < \varepsilon \tag{3.4}$$

where $v_j^{(t-1)}$ is centroid's position at previous iteration $(t-1)$ and v_j^t is the centroid's position at current iteration t.

Although FCM provides smooth, overlapping cluster boundaries, it has a few shortcomings when applied to HSI, such as:

- Each cluster is represented by a single centroid v_j, which cannot capture the often complex, multimodal distributions of spectral signatures within a land-cover class.
- Even though $u_{ij} \in [0,1]$, FCM assigns a single membership value per pixel hence not covering second order uncertainty (i.e. uncertainty about the membership itself). This leads to misclassifications in mixed-pixel regions, bands with too much noise or at class boundaries resulting in FCM being not sufficient to describe the high-order fuzzy uncertainties and cannot achieve accurate segmentation (Wu & Guo, 2021).
- Outlier pixels can disproportionately still influence the centroid positions and consequently membership updates, reducing segmentation accuracy.
- The single fuzzifier r controls the fuzziness of all cluster boundaries uniformly. This means it cannot adapt locally to varying degrees of cluster overlap.

These shortcomings motivate the move to explore more methods related to FCM while also leveraging IT2's ability to model uncertainty more effectively.

3.3 Fuzzy C-Multiple Means (FCMM) Clustering

FCMM augments FCM by introducing multiple sub-clusters to better capture intra-class variability and mitigate outlier effects. Originally proposed by Yang et al. (2023), FCMM's objective function has two parts which are executed simultaneously:

1. Subcluster formation

   Each data point x_i is assigned memberships u_{if} to q subclusters with centroids $\{m_f\}_{f=1}^{q}$ by minimizing

   $$J_1 = \sum_{i=1}^{n}\sum_{f=1}^{q} u_{if}^{r} \left\|x_i - m_f\right\|^2 \tag{3.5}$$

   where $\sum_{f=1}^{q} u_{if} = 1$ and r is again the fuzzifier exponent and $r > 1$ (Yang et al., 2023)

2. Final cluster formation

   The q clusters are now assigned to c final clusters via memberships z_{fj} by minimizing

   $$J_2 = \sum_{f=1}^{q}\sum_{j=1}^{c} z_{fj}^{r} \left\|m_f - v_j\right\|^2 \tag{3.6}$$

   where $\sum_{j=1}^{c} z_{fj} = 1$

A weighing parameter $\alpha > 0$ balances the effect of the second objective function and a unified objective function is obtained, which is as follows:

$$J_{FCMM} = \sum_{i=1}^{n}\sum_{f=1}^{q} u_{if}^{r} \left\|x_i - m_f\right\|^2 + \alpha \sum_{f=1}^{q}\sum_{j=1}^{c} z_{fj}^{r} \left\|m_f - v_j\right\|^2 \tag{3.7}$$

Like FCM, the algorithm here also alternates between membership updates and centroid updates until convergence is reached. The memberships and centroids of the subclusters and final clusters are calculated as follows:

- Subcluster memberships:

$$u_{if} = \frac{1}{\sum_{g=1}^{q} \left(\frac{\|x_i - m_f\|}{\|x_i - m_g\|} \right)^{\frac{2}{(r-1)}}} \tag{3.8}$$

- Subcluster centroids:

$$m_f = \frac{\sum_{i=1}^{n} u_{if}^r \, x_i}{\sum_{i=1}^{n} u_{if}^r} \tag{3.9}$$

- Final cluster memberships

$$z_{fj} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|m_f - v_j\|}{\|m_f - v_k\|} \right)^{\frac{2}{(r-1)}}} \tag{3.10}$$

- Final cluster centroid

$$v_j = \frac{\sum_{f=1}^{q} z_{fj}^r \, m_f}{\sum_{f=1}^{q} z_{fj}^r} \tag{3.11}$$

Overall, FCMM first treats the q subcluster prototypes $\{m_f\}$ as initial centroids and applies the FCM update rules to assign pixels to these subclusters. It then treats those same q subclusters as ”data points” and reapplies FCM to group them into the final clusters, c. This two-stage use of FCM enables FCMM to model complex, variable spectral distributions within each class – achieving better cluster boundaries while also reducing the effect of outlier pixels on the memberships and centroids. Empirical evaluation results on standard hyperspectral datasets also show that FCMM has

consistently outperformed standard FCM methods in terms of overall accuracy and shown robustness to spectral variability and noise (Yang et al., 2023).

3.4 Extension of FCMM to IT2-FCMM Objective Functions

In this subsection, the mathematical framework of FCMM is extended to incorporate IT2 FSs, introducing the IT2-FCMM approach.

The objective function now has an upper bound and is written as (Yang et al., 2023)

$$J_{UB} = \sum_{i=1}^{n}\sum_{f=1}^{q} \overline{u}_{if}^{r_2} \left\|x_i - m_f\right\|^2 + \alpha \sum_{f=1}^{q}\sum_{j=1}^{c} \overline{z}_{fj}^{r_2} \left\|m_f - v_j\right\|^2 \tag{3.12}$$

Similarly, the lower bound of the objective function is written as (Yang et al., 2023)

$$J_{LB} = \sum_{i=1}^{n}\sum_{f=1}^{q} \underline{u}_{if}^{r_1} \left\|x_i - m_f\right\|^2 + \alpha \sum_{f=1}^{q}\sum_{j=1}^{c} \underline{z}_{fj}^{r_1} \left\|m_f - v_j\right\|^2 \tag{3.13}$$

3.5 Membership and Centroid Updates

Like the objective function, the membership functions are also now split into lower and upper bounds for both: the pixel → subcluster memberships and subcluster → final cluster memberships. The final cluster centroids now too have upper and lower bounds. Calculating their final position is done by using type-reduction methods.

1. Lower and upper pixel → subcluster memberships

$$\underline{u}_{if} = \frac{1}{\sum_{g=1}^{q}\left(\frac{\left\|x_i - m_f\right\|}{\left\|x_i - m_g\right\|}\right)^{\frac{2}{(r_1-1)}}}, \qquad \overline{u}_{if} = \frac{1}{\sum_{g=1}^{q}\left(\frac{\left\|x_i - m_f\right\|}{\left\|x_i - m_g\right\|}\right)^{\frac{2}{(r_2-1)}}} \tag{3.14}$$

2. Midpoint pixel → subcluster membership calculation

$$u_{if} = \frac{1}{2}\left(\underline{u}_{if} + \overline{u}_{if}\right) \tag{3.15}$$

3. Subcluster centroid calculation

$$m_f = \frac{\sum_{i=1}^{n} u_{if}^{r_{mid}}\, x_i}{\sum_{i=1}^{n} u_{if}^{r_{mid}}}, \qquad r_{mid} = \frac{1}{2}(r_1 + r_2) \tag{3.16}$$

4. Lower and upper subcluster → final cluster memberships

$$\underline{z}_{fj} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\|m_f - v_j\|}{\|m_f - v_k\|}\right)^{\frac{2}{(r_1-1)}}}, \qquad \overline{z}_{fj} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\|m_f - v_j\|}{\|m_f - v_k\|}\right)^{\frac{2}{(r_2-1)}}} \tag{3.17}$$

5. Midpoint subcluster → final cluster membership calculation

$$z_{fj} = \frac{1}{2}\left(\underline{z}_{fj} + \overline{z}_{fj}\right) \tag{3.18}$$

6. Final cluster centroid calculation

$$v_j = \frac{\sum_{f=1}^{q} z_{fj}^{r_{mid}}\, m_f}{\sum_{f=1}^{q} z_{fj}^{r_{mid}}}, \qquad r_{mid} = \frac{1}{2}(r_1 + r_2) \tag{3.19}$$

3.6 Convergence Criteria

Iterations are stopped when the maximum absolute change in the final centroids' position falls below a threshold ε

$$\left\|V^{(t+1)} - V^{(t)}\right\| < \varepsilon \tag{3.20}$$

3.7 Proposed IT2-FCMM Algorithm

3.7.1 Algorithm Pseudocode

Input: Data matrix $X \in \mathbb{R}^{n \times d}$; subclusters q; final clusters c; fuzzifiers $r_1 < r_2$; weight α.

Initialize:

1. Random crisp membership values for subclusters and final clusters ($u_{mid}\ and\ z_{mid}$)
2. Subcluster centroids M according to Eq. 3.16
3. Final cluster centroids V according to Eq. 3.19 from subcluster centroids calculated in Step 2 and final cluster memberships initialized in Step 1.

Iterate until convergence (Eq 3.20) is satisfied or maximum number of iterations are reached:

1. Pixel → Subcluster:
   a. Compute Lower and Upper memberships $\left(\underline{u}_{if}, \overline{u}_{if}\right)$ according to Eq. 3.14
   b. Midpoint of membership $\left(u_{if}\right)$ calculation according to Eq. 3.15
2. Subcluster centroids:
   a. Update each subcluster centroid $\left(m_f\right)$ according to Eq. 3.16
      i. Lower and Upper centroids are calculated. Type reduction is done via Nie-NT (Nie & Tan, 2008) or KM (Karnik & Mendel, 2001).
3. Subcluster → Final cluster:
   a. Lower and Upper memberships $\left(\underline{z}_{fj}, \overline{z}_{fj}\right)$ according to Eq. 3.17
   b. Midpoint of membership $\left(z_{fj}\right)$ calculation according to Eq. 3.18
4. Final-cluster centroids:
   a. Update each final cluster centroids $\left(v_j\right)$ according to Eq. 3.19
      i. Lower and Upper centroids are calculated. Type reduction is done via NT (Nie & Tan, 2008) or KM (Karnik & Mendel, 2001).
5. Check for convergence according to Eq. 3.20

Output:

---

7. Final prediction labels of pixel membership to final clusters

---

3.7.2 Algorithm Workflow and Rationale

As previously mentioned in Section 1.1.1, a hyperspectral cube has three dimensions – X, Y and Z-axis. The X and Y axis contain pixels while Z-axis contains the spectral bands. In order to work on a hyperspectral image, it is converted into two-dimensional data by multiplying X and Y axis, treating them as rows and treating the bands in Z-axis as columns. In the mentioned algorithm, this forms our data matrix with n number of pixels as rows and d number of bands as columns. Since the benchmark datasets already have known number of final clusters, the variable c remains known and fixed. What can change are the amount of subclusters (q), the fuzzifier parameters (r_1, r_2) and the weighing parameter (α). This weighing parameter decides the influence that the subcluster → final cluster objective function has on both the lower and upper bounds of the overall objective functions as seen in Eqs. 3.12 and 3.13.

In the Initialization stage, first using random generators available in the Python library, membership values are generated that the pixels will have, to the subclusters. Such values are also randomly generated for the memberships that the subclusters will have to the final clusters. Using the memberships values, the centroids of the subclusters using midpoint of the memberships are calculated (Eqs. 3.15, 3.16). Similarly, centroids of the final clusters are computed using s.

When the iteration phase begins, in the first step, using the current location of the subcluster centroids, the lower and upper bounds of the pixel → subclusters memberships $\left(\underline{u}_{if}, \overline{u}_{if}\right)$ are calculated using Eq. 3.14 and a midpoint of the memberships is computed as per Eq. 3.15. In the second step, the position of the subcluster centroids – the lower and upper bounds – is updated using Eq. 3.16. Type reduction or collapsing of the intervals is done using NT (Nie & Tan, 2008) or KM (Karnik & Mendel, 2001) methods. In the third step, lower and upper bounds of memberships $\left(\underline{z}_{fj}, \overline{z}_{fj}\right)$ for subclusters → final clusters are updated using Eq. 3.17. A midpoint of the memberships is calculated using Eq. 3.18. In the fourth step, the lower and upper bounds of the centroids of the final clusters are updated using Eq. 3.19 and type-reduction methods are again used to collapse the intervals to one single value. Finally in the fifth step, it is checked whether the distance by which the final cluster centroids have moved

from their previous is greater than the assigned threshold (Eq. 3.20). If the distance is greater than the threshold (ε), the code loops back to the first step of the iteration phase and iteration starts again from the first step. If the distance is less than the set threshold, it means that further iterations are not going to move the centroids much and that the centroids of the final clusters have reached their final position.

Through this position then, as output, the subclusters get assigned to which final cluster they belong to and by implication, the pixels get assigned to which final clusters they belong to and are assigned that cluster number as a label. The algorithm has thus predicted which pixels belong to which cluster. An array of such predicted labels is given as the final output.

3.8 Visual representation of IT2-FCMM Algorithm

Figure 3.1 provides a visual insight as to how the proposed algorithm works. (a) shows two randomly generated clusters of points that can be considered as pixels. Although a visual glance is enough to conclude that the pixels can be divided into two clusters, challenges arise when it comes to classifying the ones highlighted in red – the mixed-pixel region or pixels on the borders of the clusters. To resolve this, the proposed algorithm first randomly assigns subclusters to cover the entire space so that every pixel can associate itself belonging to those subclusters. As shown in (b) the centroids of these subclusters are then considered and at this Pixel → Subcluster stage, the Interval Type-2 FCM method is used iteratively to label or associate every pixel with a subcluster to which its Euclidean distance is the shortest. The method alternates between computing Pixel → Subcluster memberships $\left(\underline{u}, \overline{u}\right)$ from which u_{mid} is calc. (Eqs. 3.14 & 3.15) and using the latest information of the memberships, updating the centroids of the subclusters using Eq. 3.16. The lower and upper boundary of the centroids is calculated, implying that the actual position of the centroid is somewhere between these bounds. The final location is calculated by using type-reduction methods - NT (Nie & Tan, 2008) or KM (Karnik & Mendel, 2001).

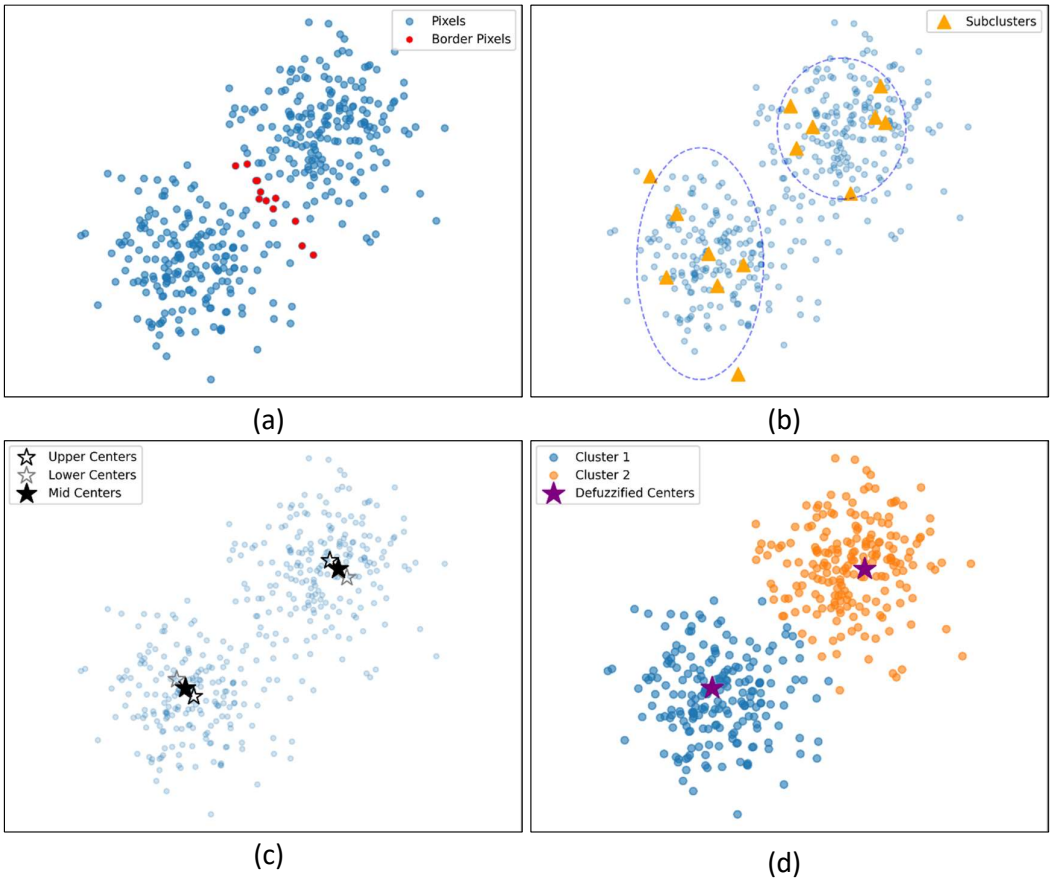

Figure 3.1 Visual representation of proposed algorithm

The subclusters are then considered as datapoints and since the final clusters are supposed to be only two, Type-2 FCM method is used again to associate the subclusters' membership to the final clusters. In this Subcluster → Final cluster stage, subcluster memberships $(\underline{z}, \overline{z})$ are calculated from which z_{mid} is calc. (Eqs. 3.17 & 3.18). Similar to the previous stage, a lower and upper bound is calculated for the centroid position of the final clusters using Eq. 3.19. (c) shows the lower and upper bound of the final cluster's centroid's position. The final position of the final cluster centroid is again calculated using type-reduction methods. (d) shows the final position of the centroid of the final clusters and since every subcluster has been assigned to one of the final clusters, by extension the pixels belonging to those subclusters also get assigned to the final clusters. Through this way it becomes easier to solve mixed-pixel regions and as output which pixel belongs to which class/label is obtained in the form of a predicted label data array.

4 Experimental Results and Evaluation

This section presents the empirical study of the proposed IT2-FCMM algorithm on three benchmark hyperspectral datasets. Section 4.1 provides the details of the datasets, the preprocessing methods employed, the metrics that are used for evaluating the proposed algorithm's performance and the configuration of the environment in which the evaluation is done. Section 4.2 presents the results of the evaluation in a tabular format along with a side-by-side visual comparison of the algorithm's proposed variants and existing Type-1 methods.

4.1 Setup and Evaluation metrics

4.1.1 Datasets

This section explains characteristics of the benchmark datasets used for the experimentation. Table 4.1 is a tabular compilation of all the characteristics such as size, number of bands, classes and labelled pixels.

4.1.1.1 Pavia University

The Pavia University scene was acquired in 2003 over the Pavia campus in northern Italy by the ROSIS-03 (Reflective Optics Systems Imaging Spectrometer) sensor. The image has a size of 610 x 340 pixels (total 207,400 pixels), with each pixel containing 103 spectral bands. The ground-truth map distinguishes nine land-cover classes—Asphalt, Meadows, Gravel, Trees, Painted Metal Sheets, Bare Soil, Bitumen, Self-Blocking Bricks, and Shadows — with a total of 42,776 labeled pixels (the remainder are unlabeled background).

4.1.1.2 Salinas

The Salinas scene was acquired by AVIRIS over California's Salinas Valley. It comprises of 512 x 217 pixels (total 111,104 pixels), with each pixel containing 200 spectral bands.

Sixteen classes – such as Vegetables, Vineyard, Soil types and Bare Earth – are labeled in a total of 54,129 pixels with the remainder being as background pixels.

4.1.1.3 Indian Pines

Collected by the AVIRIS sensor over an agricultural test site in northwest Indiana, Indian Pines scene is 145 x 145 pixels (total 21,025 pixels), with each pixel containing 200 spectral bands. Here the ground truth map distinguishes between sixteen classes (eg: various crops – Alfalfa, Corn, Soybean, plus Woods and Buildings) that span 10,249 labeled pixels and the rest being unlabeled background.

Table 4.1 Details of considered benchmark datasets

Dataset	Size (Pixels)	No. of Bands	Classes	Labeled Pixels
Pavia University	610 x 340	103	9	42,776
Salinas	512 x 217	200	16	54,129
Indian Pines	145 x 145	200	16	10,249

4.1.2 Preprocessing

Before implementing the proposed clustering algorithm, each dataset undergoes two main preprocessing steps:

1. **Mix-Max Normalization (0-1 Scaling)**

   Hyperspectral bands often differ in dynamic range, sensor response, and noise characteristics. Without normalization, some of the high variance bands may dominate the distance calculations in clustering and therefore skewing the membership updates. To avoid this, each of band's value is made to lie between $[0,1]$.

   1. For each pixel $x_i = [b_{i,1}, b_{i,2}, \dots, b_{i,D}]$ in the original D-dimensional spectral space, per band's minima and maxima is computed:

$$m_j = \min(b_{i,j})$$

$$M_j = \max(b_{i,j})$$

for $j = 1,2, \dots, D$

2. Each band's value is then scaled to $[0,1]$ via:

$$\tilde{b}_{i,j} = \frac{b_{i,j} - m_j}{M_j - m_j} \quad (4.1)$$

Outcome: The normalized pixel becomes $\tilde{x}_i = [\tilde{b}_{i,1}, \dots, \tilde{b}_{i,D}]$ with each band's value now lying between $[0,1]$.

2. Principal Component Analysis (PCA):

Hyperspectral images often have bands (D) on the order of 100 – 200, many of which are highly correlated or contain primarily noise (e.g., water - absorption regions). Directly clustering such high-dimensional data can significantly increase runtime and cause overfitting of the model due to sensitivity it must maintain for capturing so much of variance most of which is due to noise. PCA reduces dimensionality by projecting the data onto orthogonal directions of greatest variance. Rather than specifying a fixed number of components, we retain whichever set of principal components cumulatively explains 95 % of the total variance. In this study applying PCA to the three datasets results in retaining roughly 30 % of the original bands.

1. If X is the data matrix with n pixels (normalized) and D bands such that $X \in \mathbb{R}^{n \times D}$
2. First, compute the covariance matrix:

$$C = \frac{1}{n} X^{\mathrm{T}} X \quad (4.2)$$

3. Perform the eigen decomposition of C:

$$C\, v_j = \lambda_j v_j, \qquad j = 1, 2, \dots, D \quad (4.3)$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$ are eigenvalues (variances) and v_j are the corresponding eigenvectors (principal components). Each λ_j is the variance explained by the principal component j

4. Calculate the cumulative explained variance ratio:

$$\rho(k) = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{D} \lambda_D}, \qquad k = 1, 2, \dots, D \tag{4.4}$$

5. The smallest integer K is selected such that $\rho(K) \geq 0.95$. That is, the first K principal components whose eigenvalues account for 95% of the total variance are selected. In this study, K amounts to roughly 30% of the bands (D) for all three datasets.

6. A project matrix is formed

$$V_{(1:K)} = [v_1 | v_2 | \dots | v_K] \tag{4.5}$$

where each v_j is the j-th eigenvector of the covariance matrix C. The normalized data is then projected onto this reduced subspace

$$X_{\mathrm{PCA}} = XV_{(1:K)} \in \mathbb{R}^{n \times K} \tag{4.6}$$

Outcome:

- The original D-dimensional data is now reduced to a K-dimensional data, with K chosen in such a way that the entire data's 95% variance is preserved. In this way 95% of the spectral information is retained while achieving an almost 70% reduction in dimensionality.
- This reduces computational cost of clustering as well as removes those low variance and often noisy spectral bands.

4.1.3 Metrics used for performance evaluation:

To compare clustering results quantitatively, three standard metrics have been used. Overall Accuracy (OA), Average Accuracy (AA) and the Kappa Coefficient (κ). All metrics are computed on labeled pixels only – the background/unlabeled pixels are excluded. Since clustering assigns arbitrary numeric labels to clusters, a Hungarian-algorithm-based mapping is done to align predicted cluster IDs with the ground-truth labels to maximize the correct number of matches.

1. **Overall Accuracy (OA):**

OA is the ratio of those labeled pixels, whose predicted labels match the ground truth labels. It gives an overall measure of classification accuracy across all classes.

$$\begin{aligned} \text{OA} &= \frac{\text{Number of correctly classified labeled pixels}}{\text{Total number of labeled pixels}} \\ &= \frac{\sum_{i=1}^{n} 1(\hat{y}_i = y_i)}{N_{labeled}} \end{aligned} \tag{4.7}$$

where:

- $N_{labeled}$ is the total number of labeled pixels
- y_i is the true label of pixel i
- $\hat{y}_i$ is the predicted label
- $1(\cdot)$ is the indicator – 1 if the argument is true, 0 if it is false

2. Average Accuracy (AA):

AA is the arithmetic mean of per-class accuracies. Each class is given equal weightage, and each class's accuracy is the ratio of correctly classified pixels to the overall number of pixels in that class.

$$\text{AA} = \frac{1}{C}\sum_{j=1}^{C} \underbrace{\left(\frac{\sum_{i:y_i=j} 1(\hat{y}_i = j)}{\sum_{i:y_i=j} 1}\right)}_{\text{Accuracy of class j}} \tag{4.8}$$

where C is the number of classes and the fraction is the accuracy of class-j.

By assigning equal weightages, AA mitigates the effect of class-imbalance – where one class which may have greater number of pixels than other classes, may offset the overall performance.

3. Kappa Coefficient (κ):

Although OA tells the percentage of total correctly labeled pixels, it does not account for the agreement one might simply get by chance (for example, if one class is very large). This is where Kappa coefficient comes into the picture. It adjusts for this "chance agreement," so it more fairly reflects how much better the clustering has been rather than just random guessing. In Cohen's Kappa, two terms are involved: Observed agreement P_o and Expected agreement P_e. While P_o is just

OA, P_e is a bit different. If each pixel is randomly assigned to a class, but in such a way that the overall number of pixels per class stays the same as in the predictions and in the ground truth, then under such a random assignment, some pixels will match the true label purely by luck or chance. By subtracting out P_e only those extra agreements beyond what random chance would give are considered. If the clustering is just random guessing, then $P_o = P_e$ and $\kappa = 0$. If the clustering is perfect, then $\kappa = 1$

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{4.9}$$

where:

- P_o is Observed agreement $=$ OA itself
- P_e is expected chance agreement:

  A confusion matrix $\{n_{ij}\}$ is built, where n_{ij} is the number of pixels whose true class is i and whose predicted class is j

$$p_j^{(true)} = \frac{\sum_{i=1}^{C} n_{ij}}{N_{labeled}}, \quad p_j^{(pred)} = \frac{\sum_{i=1}^{C} n_{ji}}{N_{labeled}} \tag{4.10}$$

  Therefore,

$$P_e = \sum_{j=1}^{C} p_j^{(true)} \times p_j^{(pred)} \tag{4.11}$$

  This gives the probability two independent random draws (one from true labels, one from predicted labels) would coincide in class j.

If $\kappa = 1$, the clustering method's prediction matches the ground truth exactly.
If $\kappa = 0$, the clustering method's prediction is no better than a random guess.
Therefore, by reporting κ alongside OA and AA, it is ensured that high accuracy due to a dominant class does not mask the performance which may be poor in smaller classes, as κ penalizes any agreement that could be expected by chance. The results from both the variants NT and KM are compared along with the classic T1 FCM Method.

4.1.4 Environment Configuration and Considerations

The following are the environment configuration and parameters considered while implementing the algorithm:

- Computing Environment: CSC Puhti Supercomputer (HPC cluster), Finland
- Software Version: Python 3.12.8, PyTorch 2.6.0+cu124 (CUDA 12.4).
- CUDA/Driver: Nvidia driver 535.247.01 on a Tesla V100-SXM2-32GB GPU.
- Hardware: Dual-socket Intel Xeon Gold 6230 CPUs (2.10 GHz, 40 total logical cores, 2 NUMA nodes), 64 GB RAM, Ubuntu 20.04 LTS.
- GPU Details: Tesla V100-SXM2 with 32 GB VRAM.
- All major computations (membership updates, centroids) are implemented in PyTorch to leverage GPU processing capabilities for efficiency.
- Type – Reduction Choices:
  - Nie-Tan (NT): Midpoint of two weighted-average centroid updates (Nie & Tan, 2008). Computationally light.
  - Karnik-Mendel (KM): Iterative solver on sorted data (Karnik & Mendel, 2001). Computationally heavy.
- A grid search is implemented to arrive at ideal parameters: q, r_1, r_2, α.
- To ensure runtime is not excessive:
  - Convergence tolerance is kept at $\varepsilon = 10^{-5}$
  - Maximum number of iterations = 200
- Random seed for initializing $u_{mid}\ and\ z_{mid}$ seed has been kept fixed ensuring reproducibility in results.

4.2 Results and Analysis

4.2.1 Pavia University

For applying IT2-FCMM on the Pavia University dataset, after a grid search being deployed, the most optimal parameters are identified. Table 4.2 shows the parameter values when NT and KM type-reduction approaches are used. Table 4.3 shows the results

of the algorithm on the performance metrics for FCM, IT2-FCMM (NT) and IT2-FCMM (KM).

Table 4.2 Pavia University - Optimal Parameters

For NT Type Reduction	For KM Type Reduction
• Subclusters, $q = 15$	• Subclusters, $q = 17$
• Final clusters, $c = 9$	• Final clusters, $c = 9$
• Fuzzifier, $r_1 = 1.05$	• Fuzzifier, $r_1 = 1.275$
• Fuzzifier, $r_2 = 1.57$	• Fuzzifier, $r_2 = 1.52$
• Weighing parameter, $\alpha = 1.5$	• Weighing parameter, $\alpha = 8.0$

Table 4.3 Clustering Performance Metrics on Pavia University Dataset

Class	FCM	IT2-FCMM (NT)	IT2-FCMM (KM)
OA (%)	52.13%	70.97%	66.54%
AA (%)	55.60%	56.59%	55.72%
Kappa	0.4262	0.6003	0.5568

From the results, it is observed that:

- In terms of OA, IT2-FCMM (NT) showed dramatic improvement by 18.84 percentage points over FCM. Although the KM variant outperforms FCM, it lags the NT variant.
- All the three methods average around 55%-56% AA, showing that certain minority classes remain challenging even though OA has improved substantially.
- A significant improvement in Kappa by both variants – NT and KM – again confirm that IT2 uncertainty modelling significantly reduces the misassignments of the mixed pixels.
- Both IT2-FCMM variants maintain smoother and more coherent segments compared to FCM with NT being slightly smoother.

Overall, IT2-FCMM (NT) shows the biggest improvement over traditional FCM for the Pavia University HSI dataset in all three metrics with the KM variant trailing slightly behind the NT variant.

- Visualization

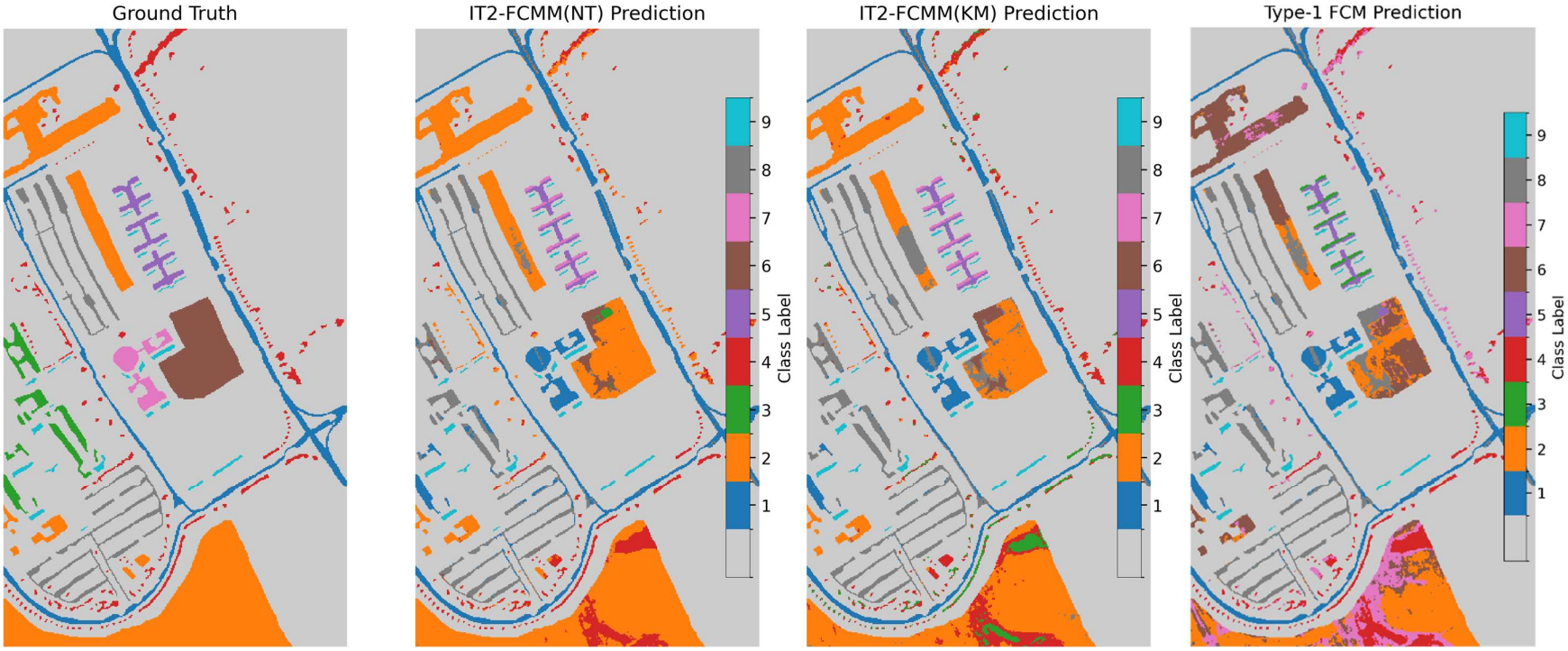

Figure 4.1 Pavia University - Ground Truth and Predictions

4.2.2 Salinas

Similar to the approach that is adopted for the Pavia University dataset, a grid search is done for identifying the optimal parameters for the Salinas dataset. Table 4.4 shows the parameter values for NT and KM type-reduction variants.

Table 4.4 Salinas - Optimal Parameters

For NT Type Reduction	For KM Type Reduction
• Subclusters, $q = 23$	• Subclusters, $q = 23$
• Final clusters, $c = 16$	• Final clusters, $c = 16$
• Fuzzifier, $r_1 = 1.475$	• Fuzzifier, $r_1 = 1.25$
• Fuzzifier, $r_2 = 1.625$	• Fuzzifier, $r_2 = 1.625$
• Weighing parameter, $\alpha = 7.75$	• Weighing parameter, $\alpha = 7.25$

Table 4.5 shows the results of the algorithm on the Salinas dataset with the performance of the traditional FCM method and type-reduction variants of IT2-FCMM.

Table 4.5 Clustering Performance Metrics on Salinas Dataset

Class	**FCM**	**IT2-FCMM (NT)**	**IT2-FCMM (KM)**
OA (%)	61.37%	75.58%	68.54%
AA (%)	59.36%	75.82%	69.80%
Kappa	0.5814	0.7244	0.6499

- Visualization:

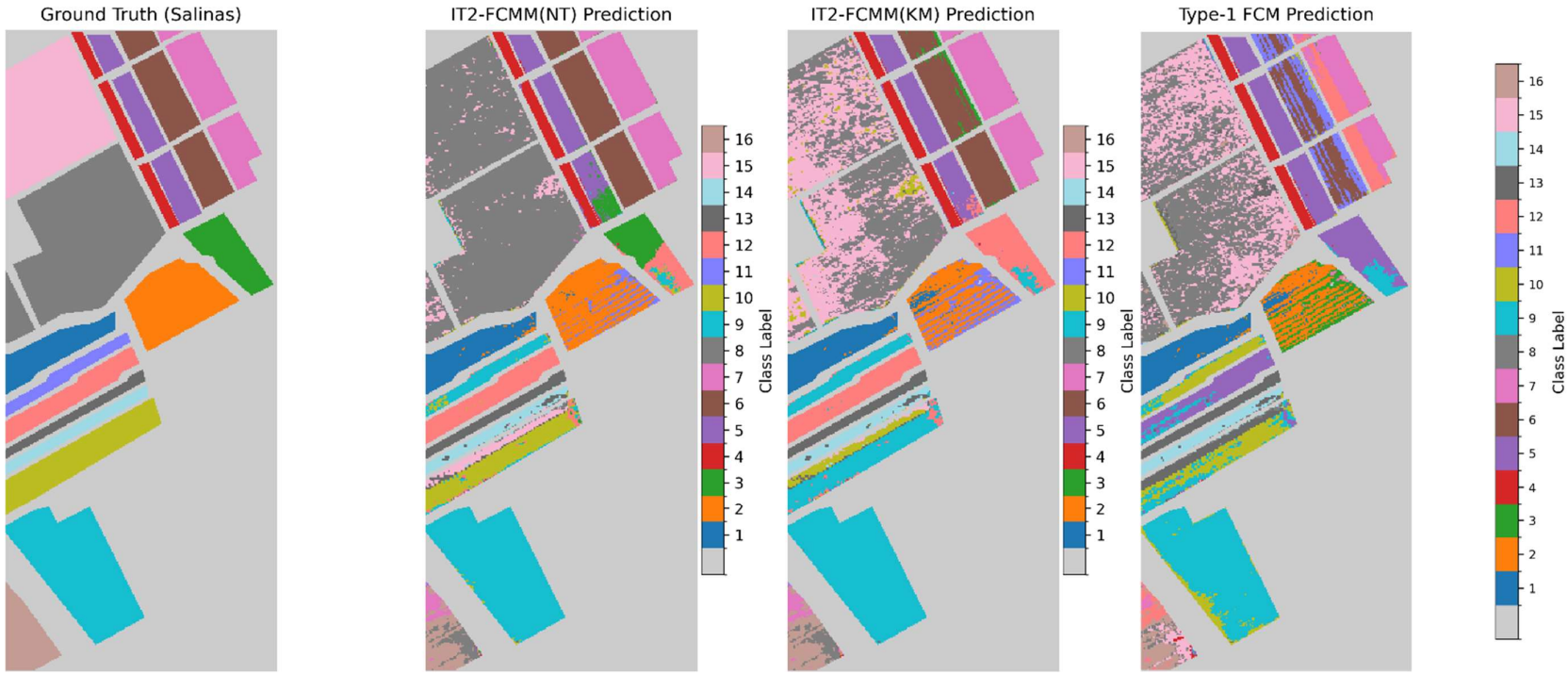

Figure 4.2 Salinas - Ground Truth and Predictions

From the results Table 4.5 and visualization shown in Fig 4.2, we see that:

- Here IT2-FCMM (NT) achieves an OA of 71.54%, a substantial 7.70 percentage point improvement over FCM's 63.84%.
- The KM variant outperforms FCM, though its OA is approximately 1.5 percentage points lower than NT's.
- These improvements highlight the benefit of using IT2 modelling along with subclusters in reducing mixed-pixel mislabeling – especially in Salinas where the adjacent crop types (for instance Grapes, Vinyard) have very similar spectral signatures.
- The larger AA gain (relative to Pavia University) indicates that IT2-FCMM not only improves overall pixel-level correctness but also balances the accuracy across the classes.
- Visually, the NT variant achieves the cleanest field boundaries – especially the narrow crops rows – while the KM variant occasionally introduces minor speckle noise. On the other hand, FCM exhibits classification noise prevalent across all 16 classes, reinforcing that IT2 along with two-stage clustering results in more precise segmentation than the classic FCM.

4.2.3 Indian Pines

With the Indian Pines dataset, a similar approach is adopted with a grid search for identifying the optimal parameters. Table 4.6 shows the chosen parameter values.

Table 4.6 Indian Pines - Optimal Parameters

For NT Type Reduction	For KM Type Reduction
• Subclusters, $q = 24$	• Subclusters, $q = 23$
• Final clusters, $c = 16$	• Final clusters, $c = 16$
• Fuzzifier, $r_1 = 1.025$	• Fuzzifier, $r_1 = 1.25$
• Fuzzifier, $r_2 = 1.475$	• Fuzzifier, $r_2 = 1.45$
• Weighing parameter, $\alpha = 4.75$	• Weighing parameter, $\alpha = 4.25$

Table 4.7 shows the results of the algorithm and its variants on the Indian Pines dataset with their performance compared with the traditional FCM method.

Table 4.7 Clustering Performance Metrics on Indian Pines Dataset

Class	FCM	IT2-FCMM (NT)	IT2-FCMM (KM)
OA (%)	33.94%	49.15%	35.29%
AA (%)	36.58%	42.31%	36.77%
Kappa	0.2802	0.4149	0.2928

- Visualization:

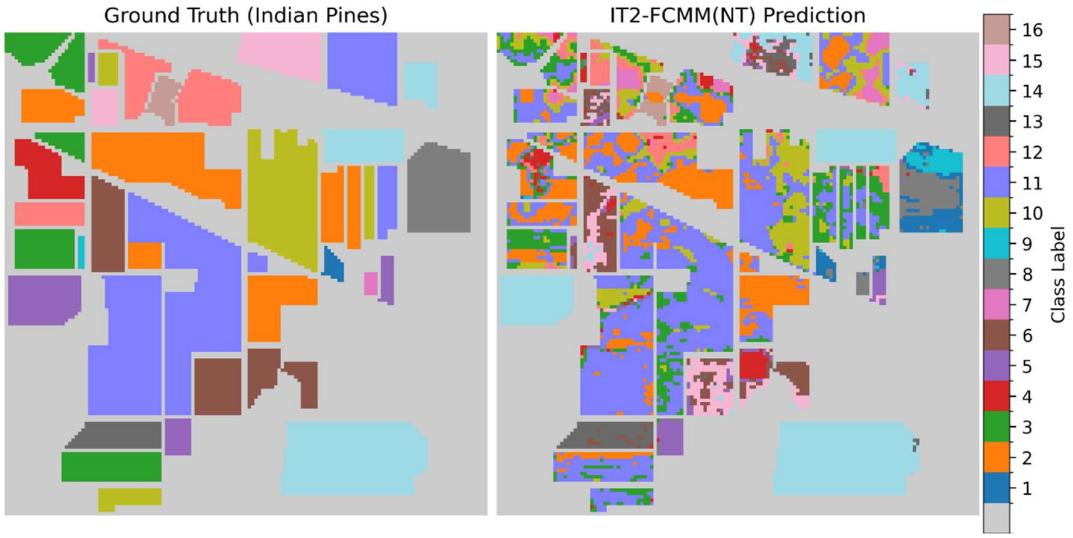

Figure 4.3 Indian Pines – Ground Truth & IT2-FCMM (NT)

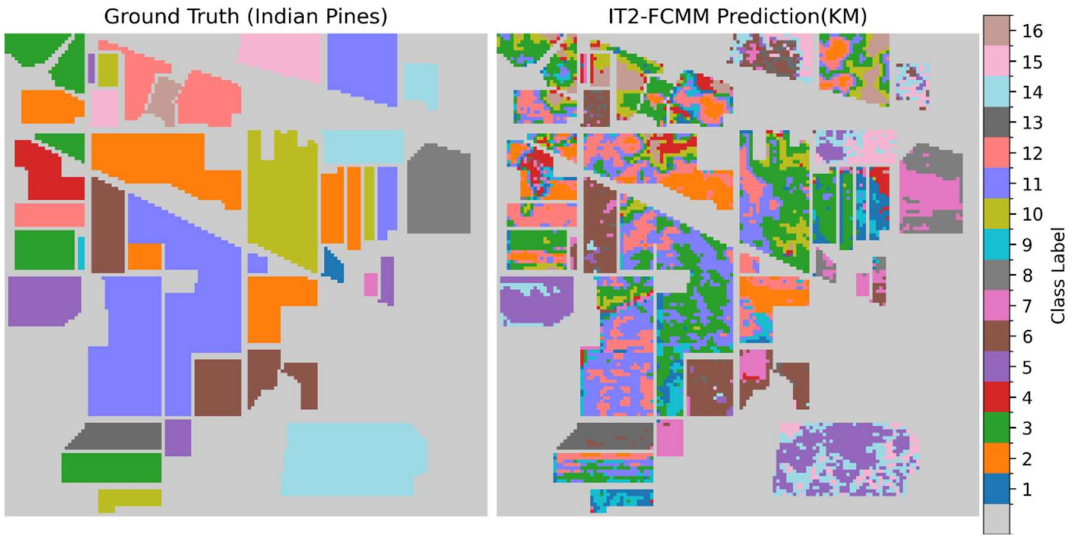

Figure 4.4 Indian Pines – Ground Truth & IT2-FCMM (KM)

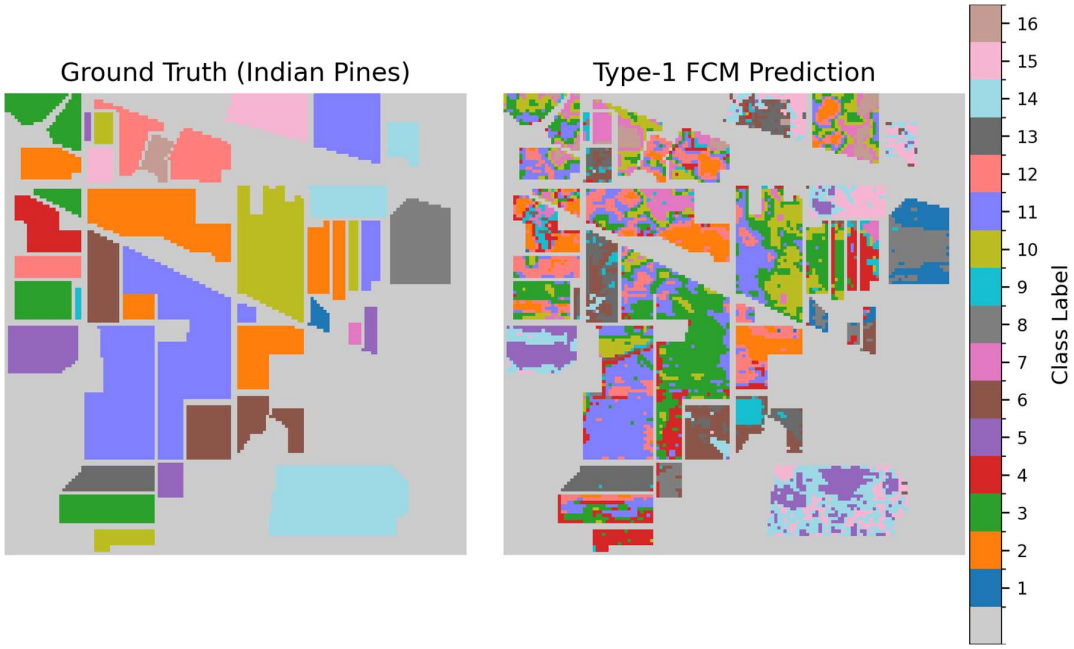

Figure 4.5 Indian Pines – Ground Truth & Type-1 FCM

From the results shown in Table 4.7 and visualizations shown in Fig 4.3, 4.4 and 4.5, we see that:

- With a 15.21 percentage point improvement over traditional FCM, IT2-FCMM – the NT variant successfully demonstrates the interval membership's ability to mitigate misclassification of boundary-pixels.

- The AA value of 42.31% by the NT variant indicates there is more uniform accuracy across all the 16 classes compared to FCM's AA = 36.58%.
- Visually, FCM exhibits misclassification throughout every class. Although the KM variant shows reduced noise compared to FCM, it does show over-fragmentation. The NT variant comparatively produces the cleanest segmentation – uniform regions for most classes, with only a few boundary pixels misassigned.
- On a class-by-class observation, the "Grass pasture" (Class 3) blocks are approximately 80% correct under the NT variant but only approximately 25% under traditional FCM.
- Overall, on the highly mixed Indian Pines scene, the NT variant of IT2-FCMM presents itself the best option in terms of accuracy.

5 Conclusions and Future Work

In this thesis, an Interval Type-2 Fuzzy C-Multiple-Means (IT2-FCMM) clustering algorithm is proposed for hyperspectral images, incorporating two distinct type-reduction strategies (NT and KM) and comparing those against a standard Type-1 Fuzzy C-Means (FCM) baseline. The experiments on three benchmark datasets (Pavia University, Salinas, and Indian Pines) demonstrate that IT2-FCMM yields consistently higher Overall Accuracy (OA), Average Accuracy (AA), and Cohen's Kappa as compared to the traditional FCM.

Returning to the questions asked in the problem statement:

1. How does the performance of the proposed method improve as compared to traditional FCM method?

Answer: The proposed IT2-FCMM algorithm with its NT and KM variants substantially outperforms standard Type-1 FCM in OA, AA and Kappa.

- In Pavia University, FCM achieved approximately 52% OA, whereas IT2-FCMM (NT variant) reached a maximum 71% OA – an improvement of approximately 19%. AA hovered around 55-56% and Kappa increased by a maximum of 40.9%.
- In Salinas, FCM OA was approximately 61% while IT2-FCMM OA increased to 75%.
- In Indian Pines, FCM OA averaged around 34% while IT2-FCMM improved the OA to approximately 49%.

Overall, in every case, IT2 based membership provided more robustness against mixed pixels, spectral variability and noise – factors that degrade T1 FCM's performance and accuracy.

2. Do type-reductions methods influence the proposed method's performance?

Answer: NT currently outperforms KM in all the 3 datasets in terms of OA and is also computationally efficient:

- Nie–Tan (NT) (Nie & Tan, 2008) uses a simple midpoint of lower-bound and

upper-bound weighted centroids. In the observed results, NT variant outperformed the KM variant in terms of OA on all datasets, while running roughly 1.5-2x faster (since it requires only two direct weighted-mean updates per centroid rather than an iterative solver).

- Karnik–Mendel (KM) iteratively sorts pixels (and subclusters in this case) values along each spectral dimension and solves for the type-reduction midpoint. KM sometimes produced marginally tighter "interval centroids," but in the experimentation results, the extra precision yielded at most a 0.3-0.5% gain (if any) in OA/AA vs. NT—hardly justifying its 2–3× longer runtime.
- For larger HSIs where speed matters, IT2-FCMM (NT) can be preferred, while KM remains an option when maximal interval precision is critical and runtime is less constrained.

5.1 Key Contributions and Observations

1. Second-Order uncertainty modelling improves clustering

   By representing each membership as an interval $(\underline{u}, \overline{u})$ at the pixel-to-subcluster stage and $(\underline{z}, \overline{z})$ at the subcluster-to-final cluster stage, the proposed IT2-FCMM is able to more faithfully capture the spectral variability, mixed pixels at class boundaries and sensor noise, something that traditional FCM's single-value memberships are unable to fully model and struggle to adapt to the "fuzziness of fuzziness".

2. Subcluster architecture captures intraclass variability

   First introduced by Yang et al., (2023), the concept of including q subclusters per class allows each land-cover category to be modelled with multiple centroids. This two-stage clustering (pixel → subcluster → final cluster) and the ability to control the effect of subcluster → final cluster using the weighing parameter α, substantially reduces the effect of outlier pixels and is extremely helpful in resolving multimodal spectral distributions that could not have been handled by a single centroid FCM.

3. GPU Acceleration makes large-scale IT2-FCMM data processing possible
   Implementing NT and KM type reductions using GPU processing via PyTorch (CUDA 11.7) reduced the runtime from hours (on CPU) to tens of seconds on the full HSI dataset—making it possible to apply the algorithm to HSIs of size $\geqslant$ 50,000 (Salinas) labelled pixels and tens of spectral bands.
4. NT vs. KM Trade-Off: Speed vs. Precision
   - NT**:** The midpoint-based solver is computationally lightweight. It yields OA/AA/Kappa much better than KM in most cases and runs 1.5–2× faster.
   - KM: The iterative solver is computationally heavier (2-3x of NT in time runtime) while only giving marginal accuracy gains. Best suited when the tightest possible interval centroids are required.
5. Robust Hyperparameter Tuning via Grid Search
   The comprehensive grid search to find the ideal hyperparameters (q, r_1, r_2, α) with a fixed seed made it possible to find near-optimal values resulting in the improvement of OA. It was ensured that class labels would not be dropped as well when choosing the ideal parameters.

5.2 Future Work

Although IT2-FCMM along with its NT and KM variants has shown a clear advantage over traditional FCM, the following are avenues of research which will further strength its application in real-world scenarios:

1. Testing the algorithm and fine tuning its application on real-world and larger HSI datasets. How the algorithm performs under different atmospheric conditions (including space environment) will only further help in evaluating its robustness.
2. As the algorithm includes fixed seeding when centroids are randomly initialized, statistical significance testing – including Wilcoxon and ANOVA tests – can be done to test whether its performance is consistent with introduction of randomization.
3. The algorithm could also be combined with other methods such as local information of neighboring pixels as proposed by Wu & Guo, (2021) to further increase accuracy.

4. Additional hyperparameters can be added for lower and upper boundaries to further increase fine tune performance. Additionally, more efficient strategies can be explored such as Bayesian optimization to find the optimal parameters rather than a full-fledged grid search.

In conclusion, this thesis demonstrates that embedding Interval Type-2 uncertainty and subcluster modeling into fuzzy clustering (IT2-FCMM) produces more accurate and robust results than classical FCM. While Nie–Tan type reduction offers nearly the same accuracy as Karnik–Mendel type reduction at a fraction of the runtime, both approaches consistently outshine FCM across multiple HSI benchmark datasets. The outlined future work – especially large-scale real-world testing, improving pixel assignment using local information, and hyperparameter optimization – will further cement IT2-FCMM as a robust unsupervised clustering framework for hyperspectral imaging.

References

Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences, 10*(2–3), 191–203. https://doi.org/10.1016/0098-3004(84)90020-7

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Cetin, H., Pafford, J. T., & Mueller, T. G. (2005). Precision agriculture using hyperspectral remote sensing and GIS. In *Proceedings of the 2nd International Conference on Recent Advances in Space Technologies (RAST 2005)* (pp. 70–77). IEEE. https://doi.org/10.1109/RAST.2005.1512537

Chen, B., Liu, L., Zou, Z., & Shi, Z. (2023). Target detection in hyperspectral remote sensing image: Current Status and challenges. *Remote Sensing, 15*(13), 3223. https://doi.org/10.3390/rs15133223

Chen, S.-Y., Hsu, K.-H., & Kuo, T.-H. (2024). Hyperspectral target detection–based 2-D–3-D parallel convolutional neural networks for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 17*, 9451–9469. https://doi.org/10.1109/JSTARS.2024.3394704

Dunn, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics, 3*(3), 32–57. https://doi.org/10.1080/01969727308546046

FavPNG. (n.d.). Hyperspectral imaging data cube illustration [Stock image]. Used under license. https://favpng.com/png_view/multispectral-image-hyperspectral-imaging-data-cube-photon-etc-market-analysis-multispectral-image-png/dbMbRgXd

Fırat, H., Asker, M. E., & Hanbay, D. (2022). Classification of hyperspectral remote sensing images using different dimension reduction methods with 3D/2D

convolutional neural network. *Remote Sensing Applications: Society and Environment, 25*, 100694. https://doi.org/10.1016/j.rsase.2022.100694

Hagras, H. A. (2004). A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Transactions on Fuzzy Systems, 12*(4), 524–539. https://doi.org/10.1109/TFUZZ.2004.832538

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A K-means clustering algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics), 28*(1), 100–108. https://doi.org/10.2307/2346830

Huang, K., Li, S., Kang, X., & Fang, L. (2015). Spectral–spatial hyperspectral image classification based on K-nearest neighbors. *Sensing and Imaging, 17*(1), 1. https://doi.org/10.1007/s11220-015-0126-z

Huang, S., Lu, S., Mukundan, A., Karmakar, R., & Wang, H. (2025). Application of hyperspectral imaging in environmental monitoring: Air pollution classification and detection. *Optics Express, 33*, 17955–17964. https://doi.org/10.1364/OE.558189

Hwang, C., & Rhee, F. C.-H. (2007). Uncertain fuzzy clustering: Interval type-2 fuzzy approach to C-means. *IEEE Transactions on Fuzzy Systems, 15*(1), 107–120. https://doi.org/10.1109/TFUZZ.2006.889763

Karnik, N. N., & Mendel, J. M. (2001). Centroid of a type-2 fuzzy set. *Information Sciences, 132*(1–4), 195–220. https://doi.org/10.1016/S0020-0255(01)00069-X

Li, N., Wang, X., Zhang, Y., Chen, L., & Liu, J. (2018). Multiparameter optimization for mineral mapping using hyperspectral imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11*(4), 1348–1357. https://doi.org/10.1109/JSTARS.2018.2814617

Liu, L., Wang, B., & Zhang, L. (2009). Decomposition of mixed pixels based on Bayesian self-organizing map and Gaussian mixture model. *Pattern Recognition Letters, 30*(9), 820–826. https://doi.org/10.1016/j.patrec.2008.05.026

Liu, S., Li, H., Jiang, C., & Feng, J. (2024). Spectral–spatial graph convolutional network with dynamic-synchronized multiscale features for few-shot hyperspectral image classification. *Remote Sensing, 16*(5), 895. https://doi.org/10.3390/rs16050895

Madhu, A., Kumar, A., & Jia, P. (2021). Exploring fuzzy local spatial information algorithms for remote sensing image classification. *Remote Sensing, 13*(20), 4163. https://doi.org/10.3390/rs13204163

Mai, D. S., Ngo, L. T., Trinh, L. H., & Hagras, H. (2021). A hybrid interval type-2 semi-supervised possibilistic fuzzy C-means clustering and particle swarm optimization for satellite image analysis. *Information Sciences, 548*, 398–422. https://doi.org/10.1016/j.ins.2020.10.003

Melgani, F., & Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing, 42*(8), 1778–1790. https://doi.org/10.1109/TGRS.2004.831865

Mendel, J. M., & John, R. I. B. (2002). Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems, 10*(2), 117–127. https://doi.org/10.1109/91.995115

Nie, M., & Tan, W. W. (2008). Towards an efficient type-reduction method for interval type-2 fuzzy logic systems. In *Proceedings of the 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)* (pp. 1425–1432). IEEE. https://doi.org/10.1109/FUZZY.2008.4630559

Polk, S. L., Cui, K., Chan, A. H. Y., Coomes, D. A., Plemmons, R. J., & Murphy, J. M. (2023). Unsupervised diffusion and volume maximization–based clustering of hyperspectral images. *Remote Sensing, 15*(4), 1053. https://doi.org/10.3390/rs15041053

Ren, Z., Sun, L., & Zhai, Q. (2020). Improved k-means and spectral matching for hyperspectral mineral mapping. *International Journal of Applied Earth Observation and Geoinformation, 91*, 102154. https://doi.org/10.1016/j.jag.2020.102154

Wang, C., Wang, X., Wu, D., Kuang, M., & Li, Z. (2022). Meticulous land cover classification of high-resolution images based on interval type-2 fuzzy neural network with Gaussian regression model. *Remote Sensing, 14*(15), 3704. https://doi.org/10.3390/rs14153704

Wang, L., Wang, H., Yin, S., & Wang, L. (2025). Masked vision transformer for fast hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing, 63*, 1–16. https://doi.org/10.1109/TGRS.2025.3572242

Wang, L., & Wang, Q. (2022). Fast spatial–spectral random forests for thick cloud removal of hyperspectral images. *International Journal of Applied Earth Observation and Geoinformation, 112*, 102916. https://doi.org/10.1016/j.jag.2022.102916

Wang, Y., Hong, D., Sha, J., Gao, L., Liu, L., Zhang, Y. Y., & Rong, X. (2022). Spectral–spatial–temporal transformers for hyperspectral image change detection. *IEEE Transactions on Geoscience and Remote Sensing, 60*, Article 5536814. https://doi.org/10.1109/TGRS.2022.3203075

Wu, C., & Guo, X. (2021). A novel single fuzzifier interval type-2 fuzzy local C-means clustering with local information for land-cover segmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14*, 5903–5916. https://doi.org/10.1109/JSTARS.2021.3085606

Yang, X., Zhu, M., Sun, B., Wang, Z., & Nie, F. (2023). Fuzzy C-Multiple-Means clustering for hyperspectral image. *IEEE Geoscience and Remote Sensing Letters, 20*, 1–5. https://doi.org/10.1109/LGRS.2023.3246633

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*(3), 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X

Zhang, C., Shen, H., & Liang, S. (2020). A hybrid random forest and CNN method for hyperspectral image classification. *Remote Sensing, 12*(10), 1682. https://doi.org/10.3390/rs12101682