MINISTRY OF EDUCATION AND TRAINING MINISTRY OF NATIONAL DEFENCE

MILITARY TECHNICAL ACADEMY

MAI DINH SINH

FUZZY CLUSTERING TECHNIQUES FOR REMOTE SENSING IMAGES ANALYSIS

MATHEMATICS DOCTORAL THESIS

HA NOI - 2021

MINISTRY OF EDUCATION AND TRAINING   MINISTRY OF NATIONAL DEFENCE

MILITARY TECHNICAL ACADEMY

MAI DINH SINH

FUZZY CLUSTERING TECHNIQUES FOR REMOTE SENSING IMAGES ANALYSIS

MATHEMATICS DOCTORAL THESIS

Major: Mathematical Foundations for Informatics

Code: 9 46 01 10

ADVISORS:

1. **Assoc/Prof.Dr Ngo Thanh Long**
2. **Assoc/Prof.Dr Trinh Le Hung**

HA NOI - 2021

DECLARATION

I hereby declare that this dissertation entitled "Fuzzy clustering techniques for remote sensing image analysis" is the bonafide research carried out by me under the guidance of **Prof. Ngo Thanh Long** and **Prof. Trinh Le Hung**. The dissertation represents my work which has been done after registration for the degree of PhD at Military Technical Academy, Hanoi, Vietnam, and that no part of it has been submitted in a dissertation to any other university or institution.

This dissertation was prepared in the compilation style format based on published papers listed in dissertation related publications. All related journal/ conference papers were conducted and written during the author's candidature.

Hanoi, March 2021

PhD Candidate

MAI DINH SINH

ACKNOWLEDGEMENTS

I would like to especially thank my supervisor, **Prof. Ngo Thanh Long**, who has been more than a supervisor to me. His passionate enthusiasm, unwavering dedication to research, and insightful advice have motivated me to overcome all challenges that arose during my PhD journey. I do appreciate all the support and opportunities that he has provided to me. I want to acknowledge my co-supervisor, **Prof. Trinh Le Hung** for his valuable advice on my research.

I would also like to thank all the members of the Department of Information Systems and Department of Survey and Mapping for their helpful discussion about research, collaboration in work. In particular, I wish to express my sincere thanks to the leaders of the Faculty of Information Technology and Institute of Techniques for Special Engineering, Military Technical Academy for providing me with all the necessary facilities for the research and continuous encouragement. I am very grateful to work in a pleasing and productive research group full of friendly, motivated, and helpful colleagues that have been a constant source of my motivation.

During the time of the dissertation, I have received valuable supports and grants. I would like to appreciate the Vietnam National Foundation for Science and Technology Development (**NAFOSTED**) sponsored the scholarship to attend a science conference in Japan in 2018. Sincerely thank the **Newton Fund**, under the NAFOSTED - UK academies collaboration programme for internship scholarship in the UK in 2019. I also want to thank the Vingroup Innovation Foundation (**VINIF**), Vingroup BigData Institute for sponsoring the scholarships for outstanding Ph.D student in 2019; University of Technology Sydney (**UTS**), Australia sponsored the scholarship to attend the research summer school at Ho Chi Minh City University of Technology in 2018. I would also like to deeply thank **Prof. Pham The Long**, who has inspired and helped me a lot in the process of applying for this internship scholarship. The tremendous support from **Prof. Hani Hagras** at the University of Essex in the UK during my internship here is also profusely thanked.

Last but not least, I would like to especially thank my family, especially my wife **Nguyen Thi Giang**, my daughters **Mai Bao Chau** and **Mai Bao Ngoc**. Who experienced all of the ups and downs of my research. Without their continued support and encouragement, I would not have had the courage to overcome all difficulties in doing research.

ABSTRACT

Remote sensing images have been widely used in many fields thanks to their outstanding advantages such as large coverage area, short update time and diverse spectrum. On the other hand, this data is subject to a number of drawbacks, including: a high number of dimensions, numerous nonlinearities, as well as a high level of noise and outlier data, which pose serious challenges in practical applications.

The dissertation develops a number of fuzzy clustering techniques applied to the remote sensing image analysis problem. The proposed methods are based on the type-1 fuzzy clustering and interval type-2 fuzzy clustering. Learning techniques and labeled data are used to overcome some disadvantages of existing methods. The problem of classification and detection of land-cover changes from remote sensing image data is applied to prove the effectiveness of the proposed methods.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Algorithms

ABBREVIATIONS

No.	Abb.	Meaning
1	CE-I	Classification Entropy index
2	CS-I	Cluster separation index
3	D-I	Dunn's separation index
4	DBSCAN	Density-based spatial clustering of applications with noise
5	DFCM	Density-based fuzzy clustering algorithm
6	EIASC	The enhanced iterative algorithm and stopping condition algorithm
7	EP	Evolutionary programming
8	FOU	Footprint of uncertainty
9	FCM	Fuzzy c-means clustering algorithm
10	FPR	False Positive Rate
11	FTM	Forest Type Mapping dataset
12	GA	Genetic algorithm
13	GSPFCM	General semi-supervised possibilistic fuzzy c-Means clustering algorithm
14	GIT2SPFCM	General interval type-2 semi-supervised possibilistic fuzzy c-Means clustering algorithm
15	IQI	Image Quality Index
16	IT2FS	Interval type-2 fuzzy set
17	IT2FLS	Interval type-2 fuzzy logic system
18	IT2FCM	Interval type-2 fuzzy c-means clustering algorithm
19	IT2PFCM	Interval type-2 semi-supervised possibilistic fuzzy c-Means clustering algorithm

20	IT2ANFIS	Interval type-2 adaptive neural fuzzy inference system
21	ISC	The improved spectral clustering algorithm
22	KFCM	The kernel fuzzy c-means clustering
23	MKIT2FCM	Multiple kernel interval type-2 fuzzy c-means clustering algorithm
24	MSE	Mean squared error
25	NDVI	Normalized difference vegetation index
26	PCM	Possibilistic c-means algorithm
27	PC-I	Bezdek's partition coefficient index
28	PFCM	Possibilistic fuzzy c-Means clustering algorithm
29	PS3VM	The self-trained semi-supervised support vector machine
30	PSO	Particle swarm optimization
31	RGB	Red-Green-Blue
32	RS	Remote sensing
33	S-I	The Separation index
34	S2KFCM	The semi-supervised kernel fuzzy c-means
35	SC	Spectral clustering algorithm
36	SFCM	Semi-supervised fuzzy c-means clustering algorithm
37	SFCM-PSO	The hybrid approach of semi-supervised fuzzy clustering and PSO
38	SKFCM-F	Semi-supervised kernel fuzzy c-means clustering
39	SMKFCM	semi-supervised multiple kernel fuzzy c-means clustering algorithm
40	SSE	Sum of Squared Error
41	SSFCM	Spatial-spectral fuzzy c-means clustering
42	T1FS	Type-1 fuzzy set

43	T2FS	Type-2 fuzzy set
44	T1FLS	Type-1 fuzzy logic system
45	T2FLS	Type-2 fuzzy logic system
46	TPR	True Positive Rate
47	VNRSC	Vietnam National Remote Sensing Center
48	ULC	Urban Land Cover dataset
49	XB-I	Xie and Beni's index

PREAMBLE

1. Problem statement

Remote sensing (RS) technology is one of the most important techniques used to collect information regarding the Earth's surface. RS image data with many advantages such as wide coverage, short update time can provide much essential data for applications [22], [54] including urban planning, mapping, classification and detection of land-cover changes, climate change, weather forecast, etc. On the other hand, RS images are also characterized by a multi-dimension nature and a high level of nonlinearities [26]; due to the effect of natural conditions during data acquisition. Therefore, they usually contain many uncertainties and vaguenesses.

In recent years, the strong development of satellite technology has led to an explosion of RS data sources [31] which has necessitated for processing of large amounts of data. In RS image analysis, the data clustering is at an early stage, but is essential for advanced image analysis issues. For clustering problems, the boundaries between objects may be unclear or overlapping, meaning that some data objects belong to different clusters. Objects on the land surface are continually changing (shape, size, color, etc) such as the change in the color of vegetation during development, change in population distribution due to socioeconomic development.

RS data collection also faces many challenges, such as the sheer volume of data and their global magnitude. The algorithms need to be sufficiently robust for for problem-solving on large datasets. There has not been a comprehensive and systematic study of classification and detection of land-cover changes from RS image data. Most studies are based on traditional classification methods such as measurement and digitization, minimum distance, maximum likelihood, object-oriented classification, etc. Other studies use NDVI image or RGB color image, which do not adequately describe the land-cover information.

Those who utilize fuzzy clustering methods also have difficulty determining the optimal parameters. Often these parameters are determined by experts based on their experience, which does not always result in the optimal selections [68]. Most fuzzy clustering methods are unsupervised [43] while supervised learning methods often require large amounts of labeled data for training.

Keeping those challenges in mind, the utilization of remote sensing image analysis is still an open question which calls for further investigation.

2. Motivations

With their many advantages, RS image data applications have been widely utilized in different applications. The rapid development of satellite technology has led to a large amount of RS image data that needs to be processed. Besides, It also faces many challenges, such as "big data", high volume and multi-dimension nature of data as well as a high degree of uncertainties and vagueness.

The urbanization process is causing constant changes to the features on the surface of the Earth. For the problem of land-cover mapping, traditional methods of creating land-cover maps are increasingly unfeasible due to budget and time constraints, which leads to the need for more modern and powerful new techniques.

For those reasons, it has become apparent that the study of RS image analysis problem is highly justified and has a great potential for academic research as well as practical applications. These are great motivations to help me choose the topic ***"Fuzzy clustering techniques for remote sensing image analysis"*** for my dissertation.

The dissertation contents will focus on developing robust clustering algorithms based on the fuzzy set including the type-1 fuzzy clustering, interval type-2 fuzzy clustering; combined with a number of learning methods and labeled data to overcome the drawbacks of previous methods. With the advantage of uncertain data processing [30], [46], fuzzy clustering is a good choice for RS image analysis problems. Moreover, the approach to semi-supervised learning method is a solution suitable for problems with very little labeled data [51], [77]. The issue of selecting the optimal parameters can be solved by using optimization techniques [72], [114].

The explanation of reasons, motivations and methods in the dissertation is as follows:

Spatial information: This method rests upon the fundamental concept that geographic regions have similar colors, so detecting those regions is good. The author has established a measurement of information about pixels' color similarity with pixels in a defined neighborhood. Such that the larger the spatial informational measure value, the higher the color similarity of the neighboring points. Furthermore, the new idea is that the larger the measure of information by neighboring pixels of the same size, the greater the chance of representing a terrain area. With that in mind, this similarity depends on two main factors: distance in color space (spectrum) and Euclidean distance of neighboring pixels. Based upon this observation, the dissertation establishes a formula for the desired measure of information. This increases the separation between pixels in one geographic area and another, which can help achieve more accurate classification. Moreover, the dissertation also proposes a method to measure the density of pixels of similar color in a neighborhood defined by a super sphere with a radius determined by the minimum standard deviation according to image channels. This density information, used as the initial focus, can stabilize the algorithm while allowing it to achieve higher accuracy.

Large data: Remote sensing images usually have many spectral channels; different image channels are usually suitable for different problem layers, which means that not all problems need to use all image channels. To reduce computational complexity, the author only selects an appropriate number of image channels based on each object's spectral reflectance characteristics.

Multi-spectrum data: This is a type of multidimensional data. The single kernel fuzzy clustering method aims to convert the image space into the single-kernel space characterized by a transform function, such as the Gaussian or the Polynomial function. The process of separating the distribution of pixels is fairly straightforward. The dissertation utilizes the multiple kernel fuzzy clustering method defined as a linear combination of Gaussian function and polynomial function. This is a complex multi-kernel transform but can improve clustering efficiency, requiring the multi-kernel linear combination optimization by the learning process.

Semi-supervised method: To optimize the clustering process, the dissertation takes advantage of the semi-supervised learning method with a limited number of samples to optimize the clustering process by determining the value of suitable parameters, including linear combination parameters of multiple multiplication function, cluster center values and parameters of the target function.

From the above analysis, it can be observed that the contribution of the dissertation compared to previous studies includes:

+ Proposing a new formula for calculating spatial information and density information;

+ Proposing a method to formulate multiple kernel functions with corrected weights during clustering;

+ Developing hybrid methods between fuzzy clustering type-1, interval type-2 with PSO technique;

+ Establishing a new objective function with tighter constraints by adopting the semi-supervised method with a limited number of samples.

Those are the basis for improving the accuracy of the proposed methods.

3. Objectives and scopes

The main objective of the dissertation is to research and develop fuzzy clustering techniques on remote sensing image data in order to improve accuracy and improve clustering quality of clustering algorithms.

The research scope of the dissertation includes the type-1, interval type-2 fuzzy clustering, and several learning methods include the semi-supervised method, kernel technique, and particle swarm optimization (PSO). The problem of classification and detection of land-cover changes from RS image is applied to prove the effectiveness of the proposed method.

4. Research method

The dissertation uses analytic tools to set up mathematical equations which are then utilized to determine optimal solutions and constructs, and prove the theorems in fuzzy clustering. The dissertation also uses programming methods to install algorithms.

Cluster quality evaluation indicators and labeled data are used to compare the dissertation's research results with others to confirm the effectiveness of the proposed solutions.

The dissertation has been conducted with strict adherence to scientific guidelines and under the supervision of academic advisors. The dissertation proposed solutions to presented problems and proved effectiveness through experiments with published research works in prestigious conferences and journals.

5. Scientific and practical meanings

Theoretically, the dissertation adopts a modern approach, while taking the advantages of the existing methods into consideration. The proposed methods also open the door to the possibility of researching solutions to apply fuzzy clustering to RS image in the case where very little labeled data is available.

Regarding practical implications, the results of the dissertation can be used in problems of land-cover classification and change detection. Besides improving the accuracy compared to some other methods, the proposed methods are more automated, thereby being more time-saving and cost-effective compared to the method using Erdas Imagine RS software.

6. Contributions of the dissertation

Most of the work described in this dissertation was conducted at the Military Technical Academy (MTA)[1] in Vietnam. The dissertation has following main contributions:

1. The dissertation proposes two unsupervised fuzzy c-means clustering algorithm (FCM), including density fuzzy c-means clustering (DFCM) [Pub7] and improved fuzzy c-means clustering (IFCM) [Pub1], [Pub3]. DFCM algorithm proposes using density information for selecting initial centroids for FCM algorithm. IFCM algorithm proposes to using the spectral clustering and spatial information as a preprocessing step to map the original data space to a new space based on the main components. The proposed methods can improve the accuracy of the algorithm compared to the original algorithm.

2. The dissertation develops three semi-supervised fuzzy c-means clustering algorithms, including semi-supervised multiple-kernel fuzzy c-means clustering (SMKFCM [Pub8]), semi-supervised fuzzy c-means clustering and the particle swarm optimization technique (SFCM-PSO) [Pub2] and interval type-2 semi-supervised possibilistic fuzzy c-means

[1] http://mta.edu.vn/

clustering and PSO technique (GIT2SPFCM-PSO [Pub9]). SMKFCM proposes the multiple-kernel technique to make data better separated; moreover, it uses labelled data to adjust the focus during clustering with the hope that the algorithm runs more stable. SFCM-PSO is a hybrid algorithm between semi-supervised method and PSO optimization technique. GIT2SPFCM-PSO is a hybrid clustering algorithm developed by the semi-supervised possibilistic fuzzy c-means clustering based on interval type-2 fuzzy set with the parameters optimized by PSO technique [Pub4], [Pub5], [Pub6]. By using PSO technique for finding the optimal parameters. The proposed methods achieve better accuracy than existing methods.

The proposed methods can be applied to many types of RS images (radar, optics) and spatial resolutions ($10m$, $30m$). Most of the experiments are used to the problem of the land cover classification of RS images. Although some limitations exist, the proposed methods can provide significantly better classification results than some other recent classification methods.

7. Organization of the dissertation

The dissertation is organized into three chapters and two sections, as follows:

Introduction: This section introduces the general issues of the dissertation. The content presented in this section includes the urgency of the research topic, motivations, objectives and scopes, contributions, scientific and practical meanings and organization of the dissertation.

Chapter 1 discusses the main issues and foundational theories used in the dissertation's studies. In this chapter, an overview of the research and some of the related works is introduced. Several reviews of previous studies with analyses of their advantages and disadvantages are also provided.

Chapter 2 introduces two unsupervised fuzzy clustering algorithms, including the density-based fuzzy c-means clustering (DFCM) and the improved fuzzy c-means clustering (IFCM).

Chapter 3 presents three semi-supervised fuzzy clustering algorithms, including the semi-supervised multiple kernel fuzzy c-means clustering (SMKFCM), semi-supervised fuzzy c-means clustering and the particle swarm optimization technique (SFCM-PSO), the interval type-2 semi-supervised possibilistic fuzzy c-means clustering and the particle swarm optimization technique (GIT2SPFCM-PSO).

Conclusions: Summary of dissertation contents, achieved issues and main contributions of the dissertation, some limitations and future research directions.

Chapter 1
BACKGROUND AND RELATED WORKS

This chapter presents the basic knowledge used in the dissertation including fuzzy clustering, interval type-2 fuzzy clustering, and learning techniques. Some methods evaluated the accuracy of the clustering algorithm is also given as a way to demonstrate the effectiveness of the method proposed in the dissertation. This chapter also addresses a number of the previous works with an analysis of their advantages and disadvantages.

1.1 Background concepts

1.1.1 Fuzzy clustering

Before introducing the fuzzy set, it is necessary to know the classical set. Let X be the space of the objects x, x be a data pattern (element) of X. A classic set A, $\mathrm{A} \in \mathrm{X}$, is a set of elements $\mathrm{A} \in \mathrm{X}$, therefore for each $\mathrm{A} \in \mathrm{X}$ may or may not belong to set A.

Definition 1.1. *Classical set A is a set of element pairs $(x, 0)$ with* $\mathrm{x} \notin \mathrm{A}$ *or $(x, 1)$ with* $\mathrm{x} \in \mathrm{A}$. *With the above definition, we can describe classical set A through the characteristic function:* $\mathrm{A} = \{(\mathrm{x}, \mu_A(\mathrm{x}))\, | \mathrm{x} \in \mathrm{X}\}$

Where $\mu_A(x)$ is a characteristic function that is defined as follows:

$$\mu_A(x) = \begin{cases} 0, x \notin A \\ 1, x \in A \end{cases} \text{ with } \forall x \in X$$

Definition 1.2. *If X is a set of objects x, a fuzzy set A, $A \subseteq X$ is defined as a set of element pairs of degree as follows:* $\mathrm{A} = \{(\mathrm{x}, \mu_A(\mathrm{x})) \,|\mathrm{x} \in \ \mathrm{X}\}$

Where $\mu_A(x)$ is a MF for the fuzzy set A [97]. The MF maps each element $x \in X$ to the interval $[0, 1]$.

With this definition, in contrast to classical sets, fuzzy sets have a MF that allows values between 0 and 1. Thus fuzzy sets are a simple extension of the classical set in which the characteristic function instead of only 0 or 1, the MF allows their values to be in the range $[0, 1]$. The MF of fuzzy set A, once returned to only 0 or 1, the fuzzy set A becomes a classical set.

a. Fuzzy c-means clustering

One of the widely used fuzzy set applications is FCM algorithm [7]. This algorithm allows each data element to belong to many different clusters according to different membership grades.

This algorithm considers MF values based on the distance from each data pattern to cluster centroids [6]. FCM algorithm model is to optimize the objective function:

$$\min\{J_m(U, V, X) = \sum_{i=1}^{c}\sum_{k=1}^{n} \mu_{ik}^m d_{ik}^2\} \tag{1.1}$$

Where $U = [\mu_{ik}]_{cxn}$ is a fuzzy MF, $V = (v_1, v_2, ..., v_c)$ is a vector of (unknown) cluster centers, $X = \{\mathrm{x}_k, \mathrm{x}_k \in \mathrm{R}^M, \mathrm{k} = 1, ..., \mathrm{n}\}$, $d_{ik} = \|v_i - x_k\|$. With the following constraints:

$$m > 1; 0 \le \mu_{ik} \le 1; \sum_{i=1}^{c} \mu_{ik} = 1; 1 \le i \le c; 1 \le k \le n \tag{1.2}$$

The objective function $J_m(U, V, X)$ reaches the smallest value when and only if:

$$v_i = \sum_{k=1}^{n} \mu_{ik}^m x_k / \sum_{k=1}^{n} \mu_{ik}^m \tag{1.3}$$

$$\mu_{ik} = 1 / \sum_{j=1}^{c} (d_{ik}/d_{jk})^{2/(m-1)} \tag{1.4}$$

Equations 1.3, 1.4 can be obtained based on the Lagrange multiplier theorem with the constraints by Equation 1.2. FCM algorithm will perform iterations according to Equations 1.3, 1.4 until the objective function $J_m(U, V, X)$ reaches the minimum value.

b. Possibilistic fuzzy c-means clustering

Possibilistic c-means algorithm (PCM) is proposed by Krishnapuram and Keller [41], which was introduced to avoid the sensitivity of FCM algorithm. Instead of using the fuzzy MFs such as FCM, PCM uses possibilistic MFs to represent typicality by τ_{ik}, the typicality matrix as $T = [\tau_{ik}]_{cxn}$.

The PCM model is the constrained optimization problem:

$$\min \{J_\eta(T, V; X, \gamma) = \sum_{i=1}^{c} \sum_{k=1}^{n} \tau_{ik}^\eta d_{ik}^2 + \sum_{i=1}^{c} \gamma_i \sum_{k=1}^{n} (1-\tau_{ik})^\eta\} \tag{1.5}$$

Where $T = [\tau_{ik}]_{cxn}$ is a possibilistic MF, $V = (v_1, v_2, ..., v_c)$ is a vector of cluster centers, $\gamma_i > 0$ is a user-defined constant. With the following constraints:

$$\eta > 1; 0 \leq \tau_{ik} \leq 1; \sum_{k=1}^{n} \tau_{ik} = 1; 1 \leq i \leq c; 1 \leq k \leq n \tag{1.6}$$

Krishnapuram and Keller also suggests using the results of FCM algorithm as a good way to initialize PCM algorithm, and the parameter γ_i should be calculated according to the following equation:

$$\gamma_i = K \sum_{k=1}^{n} \mu_{ik}^\eta d_{ik}^2 / \sum_{k=1}^{n} \mu_{ik}^\eta \tag{1.7}$$

Where μ_{ik} is the fuzzy membership from the results of FCM algorithm, K is a user-defined constant (usually selected by 1).

FCM and PCM are the most popular approaches of fuzzy clustering and possibilistic clustering, respectively. However, they suffer from drawbacks such as high sensitivity to noise and difficulty in working with overlapping data. PFCM algorithm [67] is a hybrid algorithm between FCM and PCM inheriting the advantages of both FCM and PCM. PFCM algorithm has two types of MFs, including the fuzzy MF in FCM algorithm and the possibilistic MF in PCM algorithm.

PFCM model is the constrained optimization problem:

$$J_{m,\eta}(U,T,V,X,\gamma) = \sum_{i=1}^{c}\sum_{k=1}^{n}(a\mu_{ik}^{m} + b\tau_{ik}^{\eta})d_{ik}^{2} + \sum_{i=1}^{c}\gamma_i\sum_{k=1}^{n}(1-\tau_{ik})^{\eta} \quad (1.8)$$

Where $X = \{\mathrm{x}_k, \mathrm{x}_k \in \mathrm{R}^M, \mathrm{k} = 1,..., \mathrm{n}\}$ and $U = [\mu_{ik}]_{cxn}$ is a fuzzy partition matrix, which contains the fuzzy membership degree, $T = [\tau_{ik}]_{cxn}$ is a typicality partition matrix, which contains the possibilistic membership degree, $V = (v_1, v_2, ..., v_c)$ is a vector of cluster centers, m is the weighting exponent for fuzzy partition matrix and η is the weighting exponent for the typicality partition matrix. $\gamma_i > 0$ are constants given by the user.

Subject to the constraints:

$$m,\eta > 1; a,b > 0; 0 \le \mu_{ik},\tau_{ik} \le 1; \sum_{i=1}^{c}\mu_{ik} = 1; \sum_{k=1}^{n}\tau_{ik} = 1; 1 \le i \le c; 1 \le k \le n \quad (1.9)$$

The objective function $J_{m,\eta}(U,T,V,X)$ reaches the smallest value with the constraints 1.9 when and only if:

$$v_i = \left(\sum_{k=1}^{n}(a\mu_{ik}^{m} + b\tau_{ik}^{\eta})x_i / \sum_{k=1}^{n}(a\mu_{ik}^{m} + b\tau_{ik}^{\eta})\right) \quad (1.10)$$

$$\mu_{ik} = 1/\sum_{j=1}^{c}\left(d_{ik}^2/d_{jk}^2\right)^{2/(m-1)} \tag{1.11}$$

$$\tau_{ik} = 1/\left(1 + \left(bd_{ik}^2/\gamma_i\right)^{1/(\eta-1)}\right) \tag{1.12}$$

In which, with the constraints 1.9, Equations 1.10 and 1.11 achieved in the same way as FCM algorithm, Equation 1.12 achieved in the same way as PCM algorithm.

1.1.2 Interval type-2 fuzzy c-means clustering

A T2FS in X is denoted $\tilde{A}$, and its membership grade of $x \in X$ is $\mu_{\tilde{A}}(x,u), u \in J_x \subseteq [0,1]$ [37], [57], which is a T1FS in $[0,1]$. The elements of domain of $\mu_{\tilde{A}}(x,u)$ are called primary memberships of x in $\tilde{A}$ and memberships of primary memberships in $\mu_{\tilde{A}}(x,u)$ are called secondary memberships of x in $\tilde{A}$.

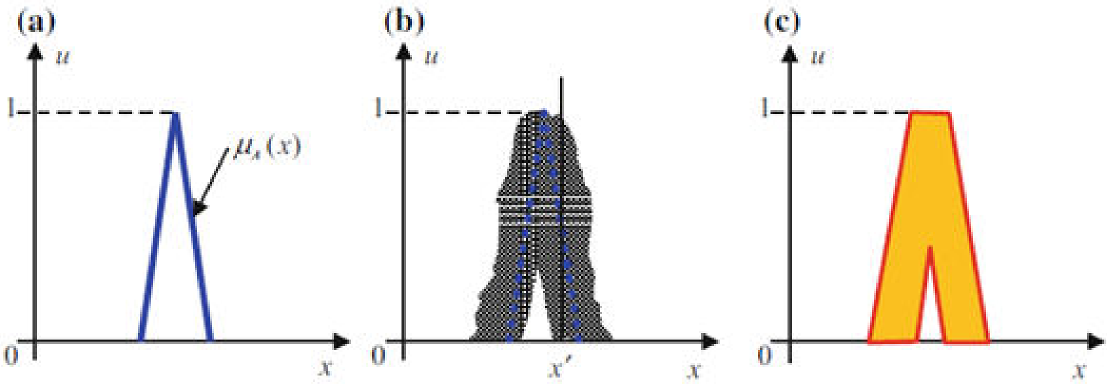


Figure 1.1: The T1FS, blurred T1FS and T2FS with uncertainty [56]

Definition 1.3. *A T2FS, denoted $\tilde{A}$, is characterized by a type-2 MF $\mu_{\tilde{A}}(x,u)$ where $x \in X$ and $u \in J_x \subseteq [0,1]$, i. e.* ,

$$\tilde{A} = \{((x,u), \mu_{\tilde{A}}(x,u)) | \forall x \in X, \forall u \in J_x \subseteq [0,1]\} \tag{1.13}$$

or

$$\tilde{A} = \int_{x\in X}\int_{u\in J_x} \mu_{\tilde{A}}(x,u))/(x,u), J_x \subseteq [0,1] \tag{1.14}$$

in which $0 \leq \mu_{\tilde{A}}(x,u) \leq 1$.

At each value of x, say $x = x'$, the 2-D plane whose axes are u and $\mu_{\tilde{A}}(x',u)$ is called a vertical slice of $\mu_{\tilde{A}}(x,u)$. A secondary MF is a vertical slice of $\mu_{\tilde{A}}(x,u)$. It is $\mu_{\tilde{A}}(x = x', u)$ for $x \in X$ and $\forall u \in J_{x'} \subseteq [0,1]$, i. e.

$$\mu_{\tilde{A}}(x = x', u) \equiv \mu_{\tilde{A}}(x') = \int_{u\in J_{x'}} f_{x'}(u)/u, J_{x'} \subseteq [0,1] \tag{1.15}$$

in which $0 \leq f_{x'}(u) \leq 1$.

T2FSs are called an IT2FSs if the secondary MF $f_{x'}(u) = 1$ $\forall u \in J_x$ i. e. an IT2FS is defined as follows:

Definition 1.4. *An IT2FS $\tilde{A}$ is characterized by an interval type-2 MF $\mu_{\tilde{A}}(x,u) = 1$ where $x \in X$ and $u \in J_x \subseteq [0,1]$, i. e. ,*

$$\tilde{A} = \{((x,u),1) | \forall x \in X, \forall u \in J_x \subseteq [0,1]\} \tag{1.16}$$

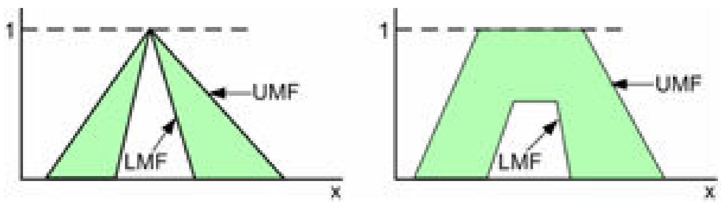


Figure 1.2: The MF of an IT2FS [45]

Uncertainty of $\tilde{A}$, denoted FOU, is union of primary functions i. e. $FOU(\tilde{A}) = \bigcup_{x\in X} J_x$. Upper/lower bounds of MF (UMF/LMF), denoted $\bar{\mu}_{\tilde{A}}(x)$ and ${}_{\tilde{A}}(x)$, of $\tilde{A}$ are two type-1 MF and bounds of FOU [58].

IT2FCM is an extension of FCM algorithm by using two fuzziness parameters m_1, m_2 to make FOU, corresponding to upper and lower values of fuzzy clustering [30]. The use of fuzzifiers gives different objective functions to be minimized as follows:

$$J_{m_1}(U,V,X) = \sum_{k=1}^{N}\sum_{i=1}^{C} u_{ik}^{m_1} d_{ik}^2 \quad and \quad J_{m_2}(U,V,X) = \sum_{k=1}^{N}\sum_{i=1}^{C} u_{ik}^{m_2} d_{ik}^2 \tag{1.17}$$

In which $d_{ik} = \|x_k - v_i\|$ is the Euclidean distance between the pattern x_k and the centroid v_i, C is number of clusters and N is the number of patterns. Upper/lower degrees of membership, $\bar{u}_{ik}$ and $\underline{u}_{ik}$ are determined as follows:

$$\bar{u}_{ik} = \begin{cases} 1/\sum_{j=1}^{C} (d_{ik}/d_{jk})^{2/(m_1-1)} & if 1/\sum_{j=1}^{C}(d_{ik}/d_{jk}) < \frac{1}{C} \\ 1/\sum_{j=1}^{C} (d_{ik}/d_{jk})^{2/(m_2-1)} & otherwise \end{cases} \tag{1.18}$$

$$\underline{u}_{ik} = \begin{cases} 1/\sum_{j=1}^{C} (d_{ik}/d_{jk})^{2/(m_1-1)} & if 1/\sum_{j=1}^{C}(d_{ik}/d_{jk}) \geq \frac{1}{C} \\ 1/\sum_{j=1}^{C} (d_{ik}/d_{jk})^{2/(m_2-1)} & otherwise \end{cases} \tag{1.19}$$

In which $i = \overline{1,C}$, $k = \overline{1,N}$.

Because each pattern has a membership interval as the upper $\bar{u}$ and the lower $\underline{u}$, each cluster centroid is represented by the interval between v^L and v^R. The algorithm to find cluster centroids is the enhanced iterative algorithm and stopping condition algorithm (EIASC) [56]. This algorithm has been shown to significantly reduce the time of cluster centroid determination compared to KM and EKM algorithms [56].

EIASC algorithm to find the v_i^R centroid is described in detail as follows:

The process to find the v_i^L centroid is similar to the Algorithm 1.1, only with changes made in steps 3, 4 and 5. EIASC algorithm to find

Algorithm 1.1 EIASC algorithm to find the v_i^R centroid

Input: Dataset $X = \{\mathrm{x}_k,\mathrm{x}_k \in \mathrm{R}^M, \mathrm{k} = 1,..., \mathrm{n}\}$, the number of clusters $c(1 < c < n)$, fuzzifier parameters m_1, m_2, m.
Output: The centroid matrices v_i^R.
Step 1: Without loss generality assum that sort n patterns on each of M features in an ascending order: $x_1 \leq x_2 \leq ... \leq x_n$ ($\bar{u}_{ik}, {}_{ik}, u_{ik}$ will also change the order corresponding to $x_1 \leq x_2 \leq ... \leq x_n$).
Step 2: Compute by using equations 1.18, 1.19 and $u_{ik} = (\bar{u}_{ik} + \underline{u}_{ik})/2$.
Step 3: Initialize $a_i = \sum_{k=1}^{n} \bar{\mathrm{u}}_{ik}\mathrm{x}_k; b_i = \sum_{k=1}^{n} \bar{\mathrm{u}}_{ik}; t = n.$
Step 4: Compute
$a_i = a_i + \mathrm{x}_t(\bar{\mathrm{u}}_{it} - {}_{it}); b_i = b_i + (\bar{\mathrm{u}}_{it} - \underline{\mathrm{u}}_{it})$
$v_i^R = a_i/b_i; t = t - 1$
Step 5: If $v_i^R > x_t$ stop, else go to Step 4.

the v_i^L centroid is described in detail as follows:

Algorithm 1.2 EIASC algorithm to find the v_i^L centroid

Input: Dataset $X = \{\mathrm{x}_k,\mathrm{x}_k \in \mathrm{R}^M, \mathrm{k} = 1,..., \mathrm{n}\}$, the number of clusters $c(1 < c < n)$, fuzzifier parameters m_1, m_2, m.
Output: The centroid matrices v_i^R.
Step 1: Without loss generality assum that sort n patterns on each of M features in an ascending order: $x_1 \leq x_2 \leq ... \leq x_n$ ($\bar{u}_{ik}, {}_{ik}, u_{ik}$ will also change the order corresponding to $x_1 \leq x_2 \leq ... \leq x_n$).
Step 2: Compute u_{ik} by using equations 1.18, 1.19 and $u_{ik} = (\bar{u}_{ik} + \underline{u}_{ik})/2$.
Step 3: Initialize $a_i = \sum_{k=1}^{n} \underline{\mu}_{ik}\mathrm{x}_k; b_i = \sum_{k=1}^{n} \underline{\mathrm{u}}_{ik}; t = 0.$
Step 4: Compute
$t = t + 1; a_i = a_i + \mathrm{x}_t(\bar{\mathrm{u}}_{it} - \underline{\mathrm{u}}_{it}); b_i = b_i + (\bar{\mathrm{u}}_{it} - \underline{\mathrm{u}}_{it})$
$v_i^L = a_i/b_i$
Step 5: If $v_i^L \leq x_{t+1}$ stop, else go to Step 4.

After obtaining v_i^R and v_i^L, a type-reduction operator is applied to obtain centroid of the i^{th} cluster. We defuzzify the interval set by using average of v_i^R and v_i^L as follows:

$$v_i = (v_i^R + v_i^L)/2 \tag{1.20}$$

For membership grades:

$$u_i(x_k) = (u_i^R(x_k) + u_i^L(x_k))/2, i = 1, ..., C \tag{1.21}$$

in which

$$u_i^L = \sum_{l=1}^{M} u_{il}/M, u_{il} = \begin{cases} \bar{u}_i(x_k) & \text{if} \quad x_{il} \quad \text{uses } \bar{u}_i(x_k)\text{for} v_i^L \\ \underline{u}_i(x_k) & \qquad otherwise \end{cases} \tag{1.22}$$

$$u_i^R = \sum_{l=1}^{M} u_{il}/M, u_{il} = \begin{cases} \bar{u}_i(x_k) \text{ if } & x_{il} \text{ uses } \bar{u}_i(x_k)\text{for}v_i^R \\ \underline{u}_i(x_k) & otherwise \end{cases} \quad (1.23)$$

Cluster centroids is computed in the same way of FCM as follows:

$$v_i = \sum_{k=1}^{N} u_{ik}^m x_k / \sum_{k=1}^{N} u_{ik}^m \quad (1.24)$$

Next, defuzzification for IT2FCM is conducted as if $u_i(x_k) > u_j(x_k)$ for $j = 1, ..., C$ and $i \neq j$ then x_k is assigned to cluster i.

Algorithm 1.3 Interval type-2 fuzzy c-means algorithm (IT2FCM)

Input: Dataset $X = \{\mathrm{x}_k, \mathrm{x}_k \in \mathrm{R}^M, \mathrm{k} = 1, ..., \mathrm{n}\}$, the number of clusters $C (1 < C < n)$, fuzzifier parameters m_1, m_2, m, and $t = 0$;
Output: The membership matrix U and the centroid matrix V.
Step 1: Initialize the centroid matrix $V^{(t)} = [v_i^{(t)}], V^{(t)} \in R^{MxC}$ by choosing randomly from the input dataset X.
Step 2: Compute $U^{(t)}$ by using Equations 1.18, 1.19, 1.21, 1.22 and 1.23).
Step 3: Repeat
3.1 $t = t + 1$
3.2 Update the centroid matrix $V^{(t)} = [v_1^{(t)}, v_2^{(t)}, ..., v_C^{(t)}]$ by using the Algorithm 1.1 or 1.2.
3.3 Compute $U^{(t)}$ by using Equations 1.18, 1.19, 1.21, 1.22 and 1.23).
3.4 Assign data x_k to the i^{th} cluster if $u_{ik} \geq u_{jk}, j = 1, ..., C; j \neq C$.
3.5 Check if $max(\left\|U^{(t+1)} - U^{(t)}\right\|) \leq \varepsilon$. If yes then stop else go to Step 3.
Defuzzification: Assign x_k to the i^{th} cluster if $u_{ik} \geq u_{jk}, j = 1, ..., C; j \neq C$.

1.1.3 Some learning methods

a. Semi-supervised method

One of the research directions that many scientists are interested in is the semi-supervised clustering method [91], which takes advantage of both supervised and unsupervised methods. They are often used in cases where the labelling data is limited to monitoring and adjusting the clustering process.

There are many semi-supervised clustering approach methods, in which the method of using additional information is commonly used. Yasunori et al. [102] proposed a semi-supervised fuzzy clustering algorithm with additional information that is used as an additional MF in the objective function of FCM algorithm.

Accordingly, the objective function of FCM algorithm is changed as follows:

$$J = \sum \sum \left| u_{\text{ij}} - \bar{u}_{\text{ij}} \right|^m \left| x_i - v_j \right|^2 \min \tag{1.25}$$

Where $\bar{u}_{\text{ij}}$ is the additional MF, which is determined by expert experience or labeled data. Subject to the constraints:

$$\bar{u}_{\text{ij}}, u_{\text{ij}} \in [0,1], \forall i = \overline{1,N}; j = \overline{1,C}; \sum_{j=1}^{C} \bar{u}_{\text{ij}} \leq 1; \sum_{j=1}^{C} u_{\text{ij}} = 1; \tag{1.26}$$

The goal of a semi-supervised clustering method is to add additional information to the clustering process to improve the accuracy of clustering results.

b. Kernel technique

There are two ways of making linear classifiers non-linear in input space:

- The first is to choose a mapping φ which explicitly gives us a kernel k [78].
- The second is choosing a Mercer kernel k which implicitly corresponds to a fixed mapping φ [25].

Though mathematically equivalent, kernels are often much easier to define and have the intuitive meaning of serving as a similarity measure between data patterns $x, y \in \chi$.

The kernel method realizes the clustering in the feature space. First, a nonlinear map is applied to map the data space to the feature space. Then, the problem can be easily solved in the high dimensional feature space. The key idea in the kernel is that we have conducted the high dimensional feature space quickly. The product in the high dimensional feature space can be calculated through the kernel function in the input space R^P [86].

However, not any symmetric function k can be used as a kernel. The necessary and sufficient conditions of k: $\chi * \chi \to R$ to be a kernel is given by Mercers theorem.

Theorem 1.1. *Functions of kernels. Let k_1: $\chi * \chi \to R$ and k_2: $\chi * \chi \to R$ be any two Mercer kernels. Then, the functions k: $\chi * \chi \to R$ is given by:*

- $k(x, y) = k_1(x, y) + k_2(x, y)$
- $k(x, y) = c * k_1(x, y) + k_2(x, y), \forall c \in R^+$
- $k(x, y) = k_1(x, y) + c, \forall c \in R^+$
- $k(x, y) = k_1(x, y) * k_2(x, y)$
- $k(x, y) = f(x) * f(y), \forall f : \chi \to R$

  are also Mercer kernels.

Theorem 1.2. *Let k_1: $\chi * \chi \to R$ be any Mercer kernels. Then, the functions k: $\chi * \chi \to R$ given by:*

- $k(x, y) = (k_1(x, y) + \theta_1)^{\theta_2}, \forall \theta_1 \in R^+, \forall \theta_2 \in N$
- $k(x, y) = \exp(k_1(x, y)/\delta^2), \forall \delta \in R^+$
- $k(x, y) = \exp(-\frac{k_1(x,x) - 2k_1(x,y) + k_1(y,y)}{2\delta^2}), \forall \delta \in R^+$
- $k(x, y) = \frac{k_1(x,y)}{\sqrt{k_1(x,x)k_1(y,y)}}$

  are also Mercer kernels.

The commonly used kernel functions are:

- Degree polynomial $K(x,y) = \left(\langle x,y\rangle_{\chi}\right)^{P}, P \in N^{+}$
- Complete polynomial $K(x,y) = \left(\langle x,y\rangle_{\chi} + c\right)^{P}, c \in R^{+}, P \in N^{+}$
- Gaussian kernel $K(x,y) = \exp\left(\frac{\|x-y\|_{\chi}}{2\partial^{2}}\right), \partial \in R^{+}$

There are two main forms of kernel-based fuzzy cluster. The first form calculates the prototype in feature space and is called KFCM-F (the F stands for feature space). In the second type, referred to as KFCM-K, the prototype is retained in the kernel space, and the prototype must be approximated in the feature space by computing an inverse mapping from kernel space to feature space [86]. The objective function of KFCM-F and KFCM-K has the same constraints as FCM as follows:

The KFCM-F objective function:

$$Q = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m}(\Phi(x_k) - \Phi(v_i))^2 \tag{1.27}$$

The KFCM-K objective function:

$$Q = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m}(\Phi(x_k) - v_i)^2 \tag{1.28}$$

c. Spectral clustering

Spectral clustering techniques make use of the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions [89]. The similarity matrix is provided as an input and consists of a quantitative assessment of the relative similarity of each pair of points in the dataset [49].

Let $X = \{x_1, x_2, ..., x_n\}$ be the set of n points to be clustered, and S be the nxn similarity matrix with its elements, s_{ij}, showing pairwise similarities between n points. Let $G = (V, S, X)$ be a weighted, undirected graph with V representing n nodes ($x_i \in X$ to be clustered), and S defining the edges. When constructing similarity graphs the goal is to model the local neighborhood relationships between the data points. There are other several popular constructions to transform the data points with pairwise similarities s_{ij} into a graph. S is usually constructed as a Gaussian function based on (often Euclidean) distances, $d(x_i, x_j)$, between samples x_i, x_j:

$$s_{\mathrm{ij}} = \exp\left(-\frac{d^2(\mathrm{x}_i,\mathrm{x}_j)}{\sigma^2}\right) \tag{1.29}$$

with a global parameter σ determining the decay of the similarity. This definition requires either a user-set σ value or a selection among many σ values to find the optimal value. D is a diagonal matrix and its elements are the degrees of the nodes of G. The degree of each node, d_i, is computed with:

$$d_i = \sum_j s(i, j) \tag{1.30}$$

The Laplacian matrix L, is constructed using the similarity matrix S and degree matrix D, depending on the approach for graph-cut optimization [79]. Ng et al. [62] defined a normalized Laplacian matrix, L_{norm}, as:

$$L_{norm} = D^{-1/2} S D^{-1/2} \tag{1.31}$$

Then L_{norm} is used for extraction of k clustering by finding its k eigenvectors with the k highest eigenvalues [89]. The spectral clustering algorithm can be summarized as follows:

d. Particle swarm optimization

PSO is an adaptive evolution algorithm based on finding the optimal

Algorithm 1.4 Spectral clustering algorithm (SC)

1. Calculate a similarity matrix S (1.29), diagonal degree matrix D (1.30), and L_{norm} (1.31).
2. Find the k eigenvectors $\{e_1, e_2, ..., e_k\}$ of L_{norm}, associated with the k highest eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_k\}$.
3. Construct the nxk matrix $E = [e_1, e_2, ..., e_k]$ and obtain nxk matrix U by normalizing the rows of E to have norm 1, i.e. $u_{\text{ij}} = e_{\text{ij}} / \sqrt{\sum\limits_k e_{ik}^2}$.
4. Cluster the n rows of U with the clustering algorithm into k clusters.

solution for the population; the principle of algorithms comes from the hunting behaviour of the birds [114]. Each problem will converge at one or several optimal solutions in the search space, considering each individual a particle and each set of particles a population.

Each state of the population in the search space is considered as a candidate solution. An optimal solution is found by moving particles in the search space according to the position and velocity as the following equation:

$$\begin{aligned} vt_i^{k+1} &= w * vt_i^k + c_1 * r_1 * (P_{ibest} - v_i^k) + c_2 * r_2 * (G_{ibest} - v_i^k) \\ v_i^{k+1} &= v_i^k + vt_i^{k+1} \end{aligned} \tag{1.32}$$

In which, v_i^k is the position of individual i^{th} in k^{th} generation, vt_i^k is the velocity of individual i^{th} in k^{th} generation, ω is the coefficient of inertia, c_1, c_2 is the acceleration coefficient, with a value of 1.5 to 2.5; r_1, r_2 is the random number, with values in the range $[0, 1]$.

Algorithm 1.5 Particle swarm optimization algorithm (PSO)

1. for $i := 1$ **to** n
  1.1 initialize v_i and vt_i.
  1.2 $P_{i\text{best}} = v_i$
2. while stop conditions not satisfied **do**
  2.1 $v_i^{(t+1)} = v_i^{(t)} + vt_i^{(t+1)}$
  2.2 update $P_{i\text{best}}$ and G_{ibest}.
  2.3 $vt_i^{(t+1)} = \omega * vt_i^{(t)} + c_1 * r_1 * (P_{i\text{best}} - v_i^{(t)}) + c_2 * r_2 * (G_{i\text{best}} - v_i^{(t)})$

In each loop, the optimal position search is performed by updating the velocity and position of the individual. In addition to each loop, the location of each individual is determined by an objective function.

1.1.4 Evaluation methods

Most of the clustering methods are unsupervised methods, so clustering results do not guarantee a consistently accurate result. There are two commonly used methods, including the internal evaluation and external evaluation. In this dissertation, both approaches are used to evaluate the quality of cluster results.

Internal evaluation: This method considers the degree of separation between clusters, clustering results are good when the data patterns in the same cluster are as close to each other as possible and the data patterns belonging to two different clusters as far as possible. This can be achieved by comparing the distance between the set of data patterns of this cluster with the set of data patterns of the remaining clusters.

Also, comparing by the variance value is a good approach, the smaller the variance shows the high degree of convergence of the objects in the same cluster. The internal evaluation method is widely used due to the easy calculation. Commonly used indices are the Bezdek's partition coefficient (PC-I), Dunn's separation index (D-I), Xie and Beni's index (XB-I), τ index, Classification Entropy index (CE-I), Cluster separation index (CS-I) [8,14,55,110], image quality index (IQI) [112], Mean squared error (MSE) [113], and Sum of Squared Error (SSE).

Large values with indexes PC-I, D-I, and IQI as well as small values with indexes SSE, CE-I, XB-I, $\tau - I$, CS-I and MSE are good for clustering results. However, to match the proposed method, we modify the calculation of some indicators accordingly as follows:

- Partition Coefficient index:

$$PC-I=\frac{1}{n}\sum_{i=1}^{C}\sum_{k=1}^{n}(\mu_{ik}^2+\tau_{ik}^2) \tag{1.33}$$

- Classification Entropy index: The classification entropy (CE) is similar to the PC but it measures the fuzziness of the cluster partition only by Bezdek.

$$CE-I=-\frac{1}{n}\sum_{i=1}^{C}\sum_{k=1}^{n}(\mu_{ik}\log\mu_{ik}+\tau_{ik}\log\tau_{ik}) \tag{1.34}$$

- Dunn's index is defined as follows:

$$\begin{aligned} &D-I=\min_{i=1,\dots,C}\left\{\min_{j=1,\dots,C;j\neq i}\left\{\delta(A_i,A_j)/\max_{t=1,\dots,C}\{\Delta(A_t)\}\right\}\right\}\\ &\delta(A_i,A_j)=\min\{d(x_i,x_j)|x_i\in A_i,x_j\in A_j\}\\ &\Delta(A_t)=max\{d(x_i,x_j)|x_i,x_j\in A_t\}\end{aligned} \tag{1.35}$$

With d is a distance function, and A_i is the set whose elements are the data points assigned to the i^{th} cluster. The main drawback with direct implementation of Dunn's index is its computation since calculation becomes much more computationally expensive as c and n increases. If a data set contains well-separated clusters, the distances among the clusters are usually large, and the diameters of the clusters are expected to be small. Therefore, large values of Dunn's index corresponds to a good clustering solution.

- Xie and Beni's index: The index (XB) aims to quantify the ratio of the total variation within clusters and the separation of clusters by Xie and Beni.

$$XB-I=\frac{1}{n}\sum_{i=1}^{C}\sum_{k=1}^{n}\mu_{ik}^m d_{ik}^2/\min_{i,j=1,\dots,C;i\neq j}\|v_i-v_j\|^2 \tag{1.36}$$

- The index τ is defined as follows:

$$\tau = \frac{1}{n} \sum_{i=1}^{C} \sum_{k=1}^{n} \tau_{ik}^{\eta} d_{ik}^{2} / \min_{i=1,\dots,C; \forall x_k \notin v_i} \|v_i - x_k\|^2 \tag{1.37}$$

- The CS measure is proposed to evaluate clusters with different densities and/or sizes. It is computed as:

$$CS - I = \frac{\frac{1}{c} \sum_{i=1}^{c} \left\{ \frac{1}{|A_i|} \sum_{x_j \in A_i} max_{x_k \in A_i} \{d(x_j, x_k)\} \right\}}{\frac{1}{c} \sum_{i=1}^{c} \{\min_{j \in c, j \neq i} \{d(v_i, v_j)\}\}} \tag{1.38}$$

Where $|A_i|$ is the number of elements in cluster A_i and $d(x_j, x_k)$ is a distance function. The smallest CS measure indicates a valid optimal clustering.

- MSE (Mean Squared Error index): $X = \{x_i\} = \{x_1, x_2, ..., x_n\}$ and $V = \{v_i\} = \{v_1, v_2, ..., v_c\}$ are the initial pixels and the centroid of the clusters, respectively. The small MSE index represents clustering results as well.

$$MSE(x, v) = \frac{1}{n} \sum_{i=1}^{c} \sum_{k=1}^{n} (x_{ik} - v_i)^2 \tag{1.39}$$

- Sum of Squared Error: SSE index is calculated by the distance from the object to its cluster centroid, the SSE index is defined as follows.

$$SSE = \sum_{i=1}^{c} \frac{1}{|A_i|} \sum_{x_j \in A_i} \|x_j - v_i\|^2 \tag{1.40}$$

- Image Quality Index:

$$IQI = \frac{4\sigma_{xy}\overline{xy}}{(\sigma_x^2 + \sigma_y^2)(\overline{x}^2 + \overline{y}^2)} \tag{1.41}$$

With $\overline{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$, $\overline{y} = \frac{1}{N} \sum_{i=1}^{N} y_i$, $\sigma_x^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \overline{x})$, $\sigma_y^2 = \frac{1}{N-1} \sum_{i=1}^{N} (y_i - \overline{y})$, $\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})$. The best value 1 is achieved if and only if $y_i = x_i$, the lowest value of -1 occurs when $y_i = 2\overline{x} - x_i$ with $i = \overline{1, N}$.

External evaluation: This method often uses a labeled data set to compare with cluster results; the advantage of external assessment method is its high reliability. Besides, statistical data that is stored in government agencies can also be used as a good way to estimate the effectiveness of the proposed method. However, the disadvantage is its dependency on the quantity and reliability of data labeled.

There are several terms that are commonly used along with the description of accuracy. It is necessary to distinguish between the misclassification of positive samples (FP) and negative samples (FN) and the correct classification of positive samples (TP) and negative samples (TN). Thus, $TP+FN$ is the number of all positive assessments; $FP+TN$ is the number of all negative evaluations. To assess the accuracy of the classification results, the performance of the classification was evaluated with the True Positive Rate (TPR) and False Positive Rate (FPR), which are defined by the following equations:

$$TPR = \frac{TP}{TP + FN} \tag{1.42}$$

Where TPR is the ratio between the number of true positive assessments and the number of all positive assessments, TP is the number of correctly classified data, and FN is the number of incorrectly misclassified data.

$$FPR = \frac{FP}{FP + TN} \tag{1.43}$$

in which FPR is the ratio between the number of false positive assessment) and the number of all negative evaluations, FP is the number of incorrectly classified data, and TN is the number of correctly misclassified data. According to equation 1.42, the value of TPR is as large as possible, while the value of FPR is as small as possible (equation 1.43).

Accuracy (ACC) is the most intuitive performance measure. Accuracy is simply a ratio of the correctly predicted classifications (both True Positives and True Negatives) to the total test dataset.

$$Accuracy(ACC) = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.44}$$

The clustering results or classification quality are shown in the indicators *TPR*, *FPR*, and *Accuracy.* The efficient algorithms have a larger values *TPR*, *Accuracy* and smaller *FTR* value.

We also compared the correct classification rate on the number of labeled pixels according to the land-cover. It is calculated by the following equations:

$$\begin{aligned} Percentage_{class_i} &= N_i^{true}/N_i \\ Percentage_{total} &= N^{true}/N \end{aligned} \tag{1.45}$$

Where $Percentage_{class_i}, Percentage_{total}$ are the correct classification rate according to the land cover and the correct classification rate for the entire area; N, N_i are the number of labeled pixels on the entire area and the i cluster; N^{true}, N_i^{true} are the number of labeled pixels correctly classified on the entire area and on the i cluster, $i = 1, 2, ..., c$.

The algorithms selected for comparison include the pre-improvement algorithms and the algorithms that have a similar approach to the proposed methods (unsupervised and semi-supervised). The content of the dissertation only focuses on developing unsupervised and semi-supervised fuzzy clustering techniques on the assumption that there are no or very little labeled data.

The purpose of the comparison method is to prove the effectiveness of the proposed method, or the approach of the dissertation has better accuracy than before improvement. Finally, to evaluate the applicability in practice, the results are compared with the best results when classified on Erdas software, which is widely used in Vietnam and around the world.

1.2 Related works

1.2.1 Overview of fuzzy clustering

Generally, clustering methods can be categorised by one of two ways: hard clustering and soft clustering (fuzzy). In hard clustering, each data pattern only belongs to a single cluster, while in fuzzy clustering, each data pattern can simultaneously belong to many clusters with different proportions. One of the widely used hard clustering algorithms is the k-Means. This algorithm is constrained by the requirement that clusters do not overlap and separate well, which many types of data cannot meet.

Fuzzy sets theory is first introduced in 1965 by L. A. Zadeh [97], and they have become a major research area in many sciences and have been widely applied in many fields. The widely used fuzzy clustering algorithm is fuzzy c-mean (FCM). The original concept was proposed by Bezdek [6]. This method was later improved and applied in the geographic data analysis problems which had shown fuzzy partitions in geographic data regions [7]. FCM algorithm has advantages in describing uncertain data. However, they are sensitive to noise and outliers. Moreover, their MFs are crisp, and they can not fully describe the types of data whose membership is not a crisp one or data with fuzzy MFs.

Recently, the commonly used method to improve accuracy is semi-supervised. The advantage of this method is that it can be utilized in cases where very little labeled data is available. Figure 1.3 shows the number of publications per year, including citations and patents when searching for the term "semi-supervised fuzzy" and "semi-supervised fuzzy in RS"[1]. Noticeably, the number of studies on semi-supervised fuzzy clustering and semi-supervised fuzzy clustering in RS is increasing rapidly from 2015 to 2020.

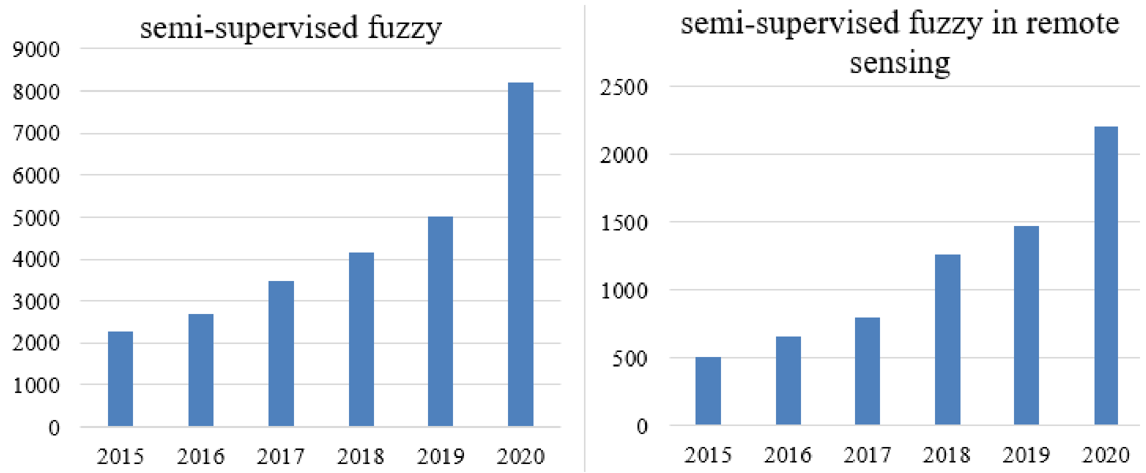


Figure 1.3: The number of papers, citations and patents on the term "semi-supervised fuzzy"

A review of previous studies indicated that, most fuzzy cluster-based approaches were an extension of FCM algorithm [21]. According to FCM algorithm, MF values are calculated by the distance between data patterns and cluster centres. There are various ways to determine the distance between the data pattern and cluster centres, which is most commonly used as the Euclidean distance. This distance is preferable with spherical clusters but less desirable with in cases where complex shapes and overlapping data are involved [7].

[1]Data from https://scholar.google.com/ on Feb 6, 2021

There are many different approaches to improving FCM algorithm, such as using kernel technique [24, 104]; using the complementary information [4, 103, 108]; using the semi-supervised method [13, 51, 95]; hybridization with other algorithms [43, 47, 102], etc. Recently, type-2 fuzzy sets (T2FSs) and interval type-2 fuzzy sets (IT2FSs) extended from original type-1 fuzzy sets (T1FSs) have shown the advantages in handling uncertainties [36, 45]. It has been developed and applied in many different problems [56,60], including RS image analysis [69,94].

The kernel fuzzy c-means clustering (KFCM) algorithm was proposed to overcome the above drawbacks by mapping input data into an appropriate space using a nonlinear function [23, 27]. This approach has received considerable attention because kernels make it possible to map data into Hilbert feature space with a higher dimensional to increase the representable capability of a linear clustering [105]. There are two typical kernel-based fuzzy clustering approaches: one with the prototypes located in the feature space (KFCM-F) and the other where the prototypes are distributed in the kernel space (KFCM-K).

A number of studies were conducted using kernel technique in clustering [78,87]. Girolami [25] proposed a kernel-based clustering method for a wider variety of clusters; Tzortzis and Likas [85] also introduced an algorithm based on kernel methods to deal with the cluster initialization problem. Later, Zhang and Chen [98] proposed the kernel-based fuzzy c-means (KFC) algorithm, which allows for incomplete data as well. Graves and Pedrycz [23, 24] launched a comprehensive comparative analysis of kernel-based fuzzy clustering and fuzzy clustering.

Because of the advantages of clustering methods based on the kernel techniques, these algorithms have been applied in many different fields, particularly in image processing. Several studies and applications, including the kernelized FCM clustering (KFCM) [104] used a kernel-induced distance metric and a spatial penalty on MFs. A novel modified kernel FCM (NMKFCM) [93] algorithm based on conventional KFCM incorporates the neighbour term in its objective function.

Recently, the density-based spatial clustering of applications with noise (DBSCAN) has been commonly used [5, 39]. This algorithm requires only one input parameter and supports the users in determining an appropriate value for it. It discovers clusters of arbitrary shape and divides high-density areas into cluster without depending on data size. In terms of implementation, this method is limited by the fact that optimal radius of the density function around each pixel is hard to determine. To overcome these limitations, Peherstorfer et al. [73] presented a grid-based density estimation method to improve the speed of clustering. Chen et al. [11] improved DBSCAN algorithm by expanding the clusters which use the bound of the objects to reduce the computation time. These improvements significantly reduce clustering time.

A variant of the fuzzy clustering based on the possibilistic approach was first proposed in [40]. This method determines a possibilistic partition with the possibilistic membership to define the typicality degree of the data pattern. One major drawback of this method is the difficulty in separating similar clusters. For improvement, Zhang et al. [100] proposed a possibilistic approach based on c-means clustering (PCM) to deal with similar clusters. However, users of PCM still have difficulty in selecting parameters, and this approach is not effective with clusters that contain complex structures and shapes.

Subsequently, in 2005, Nikhil et al. [67] proposed a possibilistic fuzzy c-means (PFCM) algorithm. PFCM is a hybridization algorithm of PCM and FCM algorithm to make use of the advantages of both the aforementioned algorithms. Although, PFCM can overcome the coincident cluster problem of PCM and the outliers of FCM, it still suffers from the drawbacks of T1FS, such as difficulty in selecting parameters and sensitivity to noise. A generalized entropy-based PFCM algorithm (GEPFCM) is proposed by Askari et al. [2] for clustering noisy data. The main objective of GEPFCM is to determine accurate cluster centres of noisy data by generalizing entropy c-means (ECM) combined with PFCM.

One of the other popular approaches to improving the accuracy of fuzzy clustering algorithms is to use the semi-supervised method. The semi-supervised algorithm introduced by Yasunori et al. [91] can be viewed as a typical algorithm in using additional information to improve the accuracy. Yin et al. [92] developed a novel semi-supervised metric-based fuzzy clustering algorithm called SMUC by introducing metric learning and entropy regularization simultaneously into the conventional fuzzy clustering algorithm based on prior membership degrees. Mai and Long [51] introduced a semi-supervised FCM clustering (SFCM) algorithm for the change detection problem on multi-spectral satellite images. The additional information from labeled pixels is added in the objective function to adjust cluster centroids and reduce the ability to fall into local optimization. Zhang et al. [101] introduces another semi-supervised clustering approach for kernel FCM algorithm (SSKFCM). Accordingly, the global optimization is obtained through repeatedly updating the fuzzy memberships and the optimized kernel parameter. However, it is difficult to choose the number of kernels for kernel-based methods, which affects both accuracy and computational complexity.

Furthermore, an approach using an ensemble of semi-supervised classifiers were proposed for change detection in remotely sensed images [77] by using multiple classifier systems in semi-supervised (learning) framework instead of a single weak classifier. In the semi-supervised change detection method, Yuan et al. [95] suggested a new distance metric learning framework by abundant spectral information in noisy hyperspectral image condition. Liu et al. [50] proposed a novel semi-supervised SVM (PS3VM) model using the self-training approach to solve the problem of RS land-cover classification.

Currently, many optimization methods do not need to use the derivative of objective functions. These methods are often called evolutionary methods [56] or methods of biological inspiration [72] such as EP, GA, PSO, simulated annealing, differential evolution, ant colony optimization, gravitational search, etc. These methods tend to be stronger than derivative-based methods because the process of finding a globally optimal solution is repeated many times until convergence is achieved.

The standard fuzzy clustering methods are also dependent upon the initialization of centroids and the selection of parameters. It is easily stuck in a local optimization [43, 54, 114]. To deal with this issue, Zhang et al. [102] proposed a hybrid method of fuzzy clustering and PSO to find the optimal parameters. Lilin et al. [47] proposed an improved fuzzy clustering method based on a self-adaptive cell genetic algorithm.

These bio-inspired methods usually require a large number of loops to find the optimal solution, and they need a significant amount of time to evaluate the objective function for each candidate. If the calculation time is not subject to constraints, these are potent methods. The advantage of PSO algorithm is its faster convergence than GA algorithm, which is suitable for large data sets such as satellite image data. Besides complex calculations, the disadvantage of using the derivative is the fact that the calculation will change as the upper MF or lower MF cause changes to the mathematical equation on the specified domain. Moreover, they are easily stuck at a local extreme [7, 37].

Despite their widespread use, existing fuzzy clustering methods still face some of the following issues [90]: (1) the Euclidean distance tends to work poorly if the importance of the features is different and (2) it is difficult to determine the optimal parameter for the objective function. (3) the fuzzy clustering based on T1FSs is not able to fully describe uncertainties because their MFs are crisp. Therefore, they cannot avoid the disadvantages of T1FSs.

1.2.2 Overview of type-2 fuzzy clustering

Over the years, there has been a significant increase in the study of high-level fuzzy logic types; especially the use of general type-2 and interval type-2 fuzzy sets. Figure 1.4 shows the number of publications per year, including citations and patents when searching for the term "type-2 fuzzy" and "type-2 fuzzy in RS"[2] from 2015 to 2020.

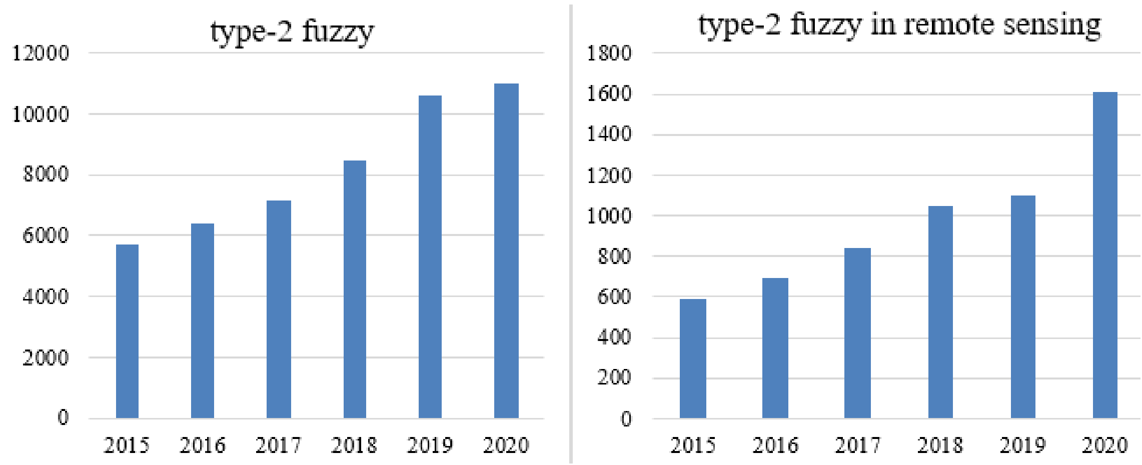


Figure 1.4: The number of papers, citations and patents on the term "type-2 fuzzy"

T2FS is an extension of the type-1 fuzzy set (T1FS) to deal with the uncertainty data [34, 37] which is applied in many fields, including satellite image classification [60, 64]. While two-way MFs characterize T1FSs, wherein each pattern of T1FS has a membership grade that is a crisp number in $[0, 1]$. T2FSs are characterized by self-fuzzy MFs, meaning that the membership grade for each pattern of a T2FS is a fuzzy set in $[0, 1]$ [36, 45]. MFs of T2FSs are three dimensional and include a footprint of uncertainty (FOU), it is the new third dimension of T2FSs and the footprint of uncertainty that make it possible to directly model and handle uncertainties [35, 37]. The T2FSs is useful in circumstances where it is difficult to determine the exact MF for a fuzzy set, which is helpful in incorporating uncertainties [57].

On the other hand, MF value of a T1FS is a crisp number. For many types of data patterns, it is difficult to determine the crisp values for MFs. Once the type-1 MF has been chosen, all uncertainties disappear

[2]Data from https://scholar.google.com/ on Feb 6, 2021

because type-1 MF is totally precise [34]. MF of T2FS is self-fuzzy, which may be modeled uncertainties better than the T1FS. In case MF is crisp, then T2FSs become T1FSs, the same probability becomes determined [56].

T2FSs can describe uncertainty better than T1FSs because their MFs are self-fuzzy. Once there is no uncertainty, then T2FS decreases to a T1FS, similar to the probability of returning to the determination. The difficulty in working with T2FSs is highly computational expensive [45]. In fact, the special case that is often used is the interval type-2 fuzzy set [58] and the clustering algorithm based on IT2FS is the interval type-2 fuzzy c-means clustering algorithm (IT2FCM) [30].

In the study [60], Melin et al. reviewed some applications of T2FS in classification and pattern recognition and pointed out that the general T2FS is limited by high computational complexity [57] and difficulty in installation. So, in practical applications, IT2FS, the special case of the general T2FS is more widely used [45, 58]. One of the ways to apply IT2FS in clustering is to use the interval type-2 FCM clustering algorithm (IT2FCM) [22,30]. In IT2FCM, FOU (footprint of uncertainty) of T2FS is built by using two fuzzier values to handle uncertainties. Some studies applying IT2FCM algorithm for RS image clustering problems can be found in [69, 94].

Recently, there have been some studies improving IT2FCM algorithm. Accordingly, in [52] and [26], a new distance was introduced to replace the traditional Euclidean distance in IT2FCM algorithm using both spectral information and spatial information for multispectral RS image clustering. In [68], SIIT2FCM algorithm expanded from IIT2FCM [52] for the problem of change detection on multispectral satellite images that have used the spatial information (SIIT2FCM) and the semi-supervised method to improve the accuracy of classification results. Some studies develop IT2FCM algorithm mentioned in [64, 65] using the multiple kernel technique for data classification.

Besides, Ji et al. [33] introduced an interval-valued PFCM algorithm using both fuzzy MFs and possibilistic MFs to model the uncertainties. This method can significantly improve accuracy when compared with the original PFCM algorithm. Wang et al. [111] proposed a supervised classification method for the high-resolution RS image based on IT2FS by analyzing the data characteristics and building an interval type-2 MF to model the uncertainty of pixels. However, this algorithm requires a lot of labeled data to train. An improvement from IT2FCM algorithm for land-cover classification from hyperspectral image data is proposed by Huo et al. [31], in which, the interval type-2 fuzzy MF is ranked by the confidence level based on the uncertainty of the spectral information.

1.2.3 Some limitations of the above methods and solutions

In Vietnam, with the introduction of the "National Program of Space Science and Technology"[3], the government has encouraged the research and application of science and technology to solving problems related to satellite images. Some recent studies have applied object-oriented classification techniques, support vector machines, etc. The land-cover classification studies from RS data mainly use pixel-based clustering methods

[3] http://spaceprogram.vast.vn/

such as minimum distance, maximum likelihood. However, the number of research projects of exploiting RS image is not commensurate with its potential and the practical applications are still quite limited.

There are very few studies applying fuzzy sets (T1FS, T2FS) for RS image analysis [51]. Most of the recent studies involve unsupervised methods which can be mentioned such as interval type-2 fuzzy clustering [64,69], collaborative fuzzy clustering [70], fuzzy co-clustering [75,76] or fuzzy clustering based on granular computing [84]. Furthermore, these studies only used RS images in natural color combinations [70] or normalized difference vegetation index (NDVI) images [64, 69] to classify, resulting in data uncertainty that is not fully described. Some hybrid methods only determine the optimal number of clusters and still have difficulty determining the optimal fuzzy parameters. The common point of these studies is that it is challenging to work on large image areas, unable to leverage the knowledge gained from previous classification results; and their accuracy is highly dependent on data sampling.

Although the fuzzy theory has been widely used in many problems [66, 71, 74], they still have limitations such as high computational complexity, especially when using T2FSs; difficulty in choosing the optimal parameters; sensitivity to noise and outlier elements, etc. Moreover, Euclidean distances in most algorithms are not effective with overlapping clusters and complex shapes [68, 81].

Firstly, most fuzzy clustering algorithms are unsupervised methods [22, 61], and the desired ac-curacy may not be guaranteed in the clustering results [49,102]. In contrast, supervised methods require a sufficiently large amount of labeled data for training. These methods are usually ineffective when labeled data is limited. In this case, the semi-supervised method is a more suitable solution [91]. This method has the advantages of the supervised method and overcomes the disadvantages of the unsupervised method [77, 95]. Many research results on semi-supervised methods recently in the field of RS image analysis and other areas show the enormous potential of this method [51, 81].

Second, selecting the optimal parameters is an challenging task for most fuzzy clustering algorithms. A commonly used approach to this issue is the hybridization method between fuzzy clustering and optimization techniques such as particle swarm optimization (PSO) [102, 114], genetic algorithm (GA) [63], evolutionary computation [9], neural networks [109].

Third, most clustering algorithms ignore spatial relationships between data patterns, which are often used in advanced image analysis problems. Therefore, the study of using additional information in the clustering process is necessary to improve accuracy [49]. Additional information can include information about spatial relationships, information about the density of objects, information about labeled data, so on [86, 107].

Fourth, RS image has high overlapping characteristics, complexly shaped clusters, and uncertain data. The use of Euclidean measurements has many limitations when working with data that contain overlapping clusters. Spatial transformations such as kernel techniques, spectral clustering, principal component analysis are often used to model data and separate clusters better to make clustering easier [12, 29, 107].

Keeping some limitations mentioned above in mind, the research and development of fuzzy clustering algorithms is still a research direction with great potential. The dissertation focuses on developing fuzzy clustering methods based on fuzzy logic, including T1FS, T2FS. The study also used optimization techniques and additional information to deal with the limitations of some previous studies. With the ability to cope with uncertainty and nonlinear data, fuzzy clustering promises to address the challenges of RS image processing problems. The experiments were performed on a multi-spectral satellite image (Landsat-5 TM, Landsat-7 ETM+, Landsat-8, Sentinel-2A) for land-cover classification and change detection.

1.3 Framework of remote sensing image analysis problem

Figure 1.5 shows the general framework of the proposed algorithms in the dissertation. In this framework, RS image data is extracted into components: spectral information, spatial information and labeled data. Depending on the image data collected, the input for the proposed algorithm may have one, two or all three components.

The proposed algorithm will perform on the input data set, resulting in data clusters. The clusters are assigned to the corresponding land-covers based on labeled data samples. The accuracy of classification results is assessed based on cluster quality measurement indicators and labeled data (if any).

Experiments are written in Visual C++ 2015 environment using libtiff library[4]. Experimental data are taken from the UCI Machine Learn-

[4] http://www.libtiff.org

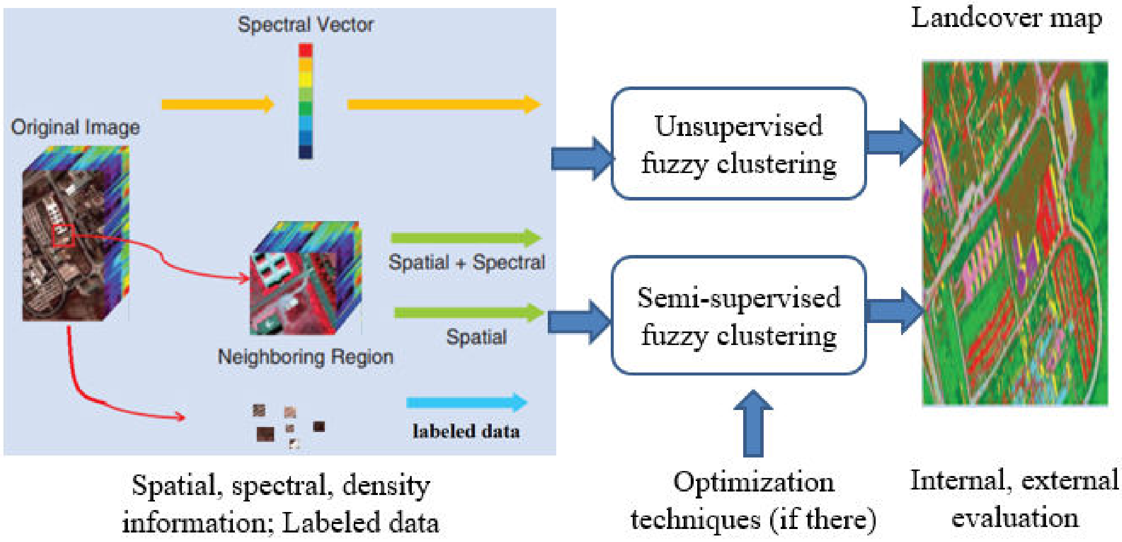


Figure 1.5: Framework of remote sensing image analysis problem

ing Repository library[5], the Vietnam National Remote Sensing Center (VNRSC) and a number of publicly available data sources provided on the internet[6]. Before using the proposed methods, data preprocessing problems such as geometry correction, atmospheric correction are handled by Erdas Imagine software (Version 2016).

The details of the framework consist of the following four main steps:

Algorithm 1.6 General steps of remote sensing image analysis problem

Input: Remote sensing image data, parameters of proposed algorithm
Ouput: Map of land-covers.
Step 1: Pre-processing step for remote sensing image data.
Step 2: Remote sensing image data is clustered by the proposed algorithm
Step 3: Clustering results will be classified into landcover classes.
Step 4: Compute the percentage of the individual regions:

$$S_i = n_i/N \tag{1.46}$$

where S_i is the area of the i^{th} region, n_i is the number of pixels of the i^{th} region, N is the total samples of n-bands imagery.

[5]https://archive.ics.uci.edu/
[6]http://earthexplorer.usgs.gov/, https://earth.esa.int/

1.4 Chapter summary

Chapter 1 has introduced an overview of research issues, related background theories, and reviewing previous work related to the dissertation. Several commonly used methods to evaluate the accuracy of RS image classification results are also introduced. In the next chapter, the dissertation will present some improvements of FCM algorithm.

Chapter 2
FUZZY C-MEANS CLUSTERING ALGORITHMS USING DENSITY AND SPATIAL INFORMATION

This chapter will present improved unsupervised-methods from FCM algorithm. Algorithms are introduced including DFCM and IFCM.

The main idea behind DFCM is to use density information to determine initial centroids instead of random initialization.

IFCM is the algorithm that uses information about spatial relationships of pixels to better identify geographical regions. The spectral clustering algorithm is also used in IFCM with the expectation of better data separation. Each proposed algorithm has a different approach, but the goal is to overcome some of the limitations that clustering algorithms are subject to.

2.1 Introduction

Despite it widespread use, FCM still has many limitations, such as difficulty in parameter selection and poor performance with overlapping clusters. This chapter will discuss and present some improvements of FCM algorithm, which uses the information on density and spectral clustering.

Firstly, the concept of density can be understood as the quantity representing the amount of matter in a unit of measure (length, area, volume) [5]. Usually, the centroid of a cluster is the average numerical value of the pixels. If the pixel has a high frequency of appearance, that pixel is closer to the centroid of the cluster.

The algorithm uses information about the density, which has been recently used as DBSCAN [39]. It discovers clusters of arbitrary shape and divides high-density areas into cluster regardless of data size. In terms of implementation, this algorithm is limited by the difficulty in finding an optimal radius of the density function around each pixel [73], [11]. In this chapter, the initial centroids are chosen based on the density of the pixels.

Secondly, the spectral clustering is a clustering method based on algebraic graph theory [89]. The advantage of this method is its capability to cluster with functional data separation. Spectral clustering is applied in various fields such as medical, remote sensing, bio-informatics. However, the raw spectral clustering is often based on Euclidean distance, while ignoring information about the spatial relationship between pixels. It suffers from several drawbacks, the inability to determine a reasonable cluster number and high sensitivity to initial condition and not robust to outliers.

The Gaussian kernel function is widely used for spectral clustering which measures the similarity between data points. However, choosing a suitable scaling parameter in the Gaussian kernel similarity measure is not an easy task. In this chapter, a robust approach named improved fuzzy c-means clustering (IFCM) is introduced. The proposed method combines spectral clustering and fuzzy clustering with spatial information to deal with clustering problems on RS image. This method can find the spatial distribution characteristics of complex data, and can further stabilize clusters. Experimental results show that it can improve the accuracy and minimize the risk of falling into local optimum.

The next sections of the chapter will provide further details of the proposed methods.

2.2 Density fuzzy c-mean clustering

In this section we propose a method for clustering satellite imagery based on density. It consists of two main steps: finding cluster centroid using density, and data clustering using fuzzy c-Means algorithm (DFCM). The results obtained in this study can be used to potentially improve classification accuracy of satellite images.

2.2.1 Proposed method

One of the difficulties of clustering algorithms is the initialization of the initial cluster centroid. This affects the steps taken and results in clustering, if the centroid of the initiator cluster is too close together or too far apart, it will quickly lead to local convergence, which negatively affects the accuracy or stability of the clustering algorithm. There should be an approach to initializing the centroid of clusters that makes clustering algorithms stable and efficient. In this study, initialization of cluster center was proposed based on the density of pixels and FCM algorithm applied to the land cover classification on RS image.

The image data is stored as numeric values, and partition problem is usually based on the degree of similarity among these values to decide whether an object belongs to any region in the image. Hence, the key to determine a pixel belonging to a particular area is based on the similarity in spectral values. This measurement is calculated through a distance function in the color space d_{ik} between the pattern x_k and the centroid v_i. Meanwhile, the centroid will be in the samples where the density surrounding the sample data is large.

For the first step, the mean pattern $\bar{x}_j$ is computed by the following equation:

$$\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{\text{ij}}, j = \overline{1, d} \tag{2.1}$$

And standard deviation s_i:

$$s_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (x_{\text{ij}} - \bar{x}_i)^2}, j = \overline{1, d} \tag{2.2}$$

In which, $i = \overline{1, d}$, $X = \{x_1, x_2, ..., x_n\}, x_k \in R^d, k = \overline{1, n}$. Considering the surrounding of each data point a m-dimensional box with a radius defined by the standard deviation of $r = \min\limits_{1<j<d}(s_j)$. Density D_i of pattern x_i is computed as:

$$D_i = \sum_{j=1}^{n} T(r - |x_j - x_i|) = \sum_{j=1}^{n} T(\Delta r); T(\Delta r) = \begin{cases} 1 & \Delta r \geq 0 \\ 0 & \Delta r < 0 \end{cases} \tag{2.3}$$

Call V a set of pixels in order of density from high to low. Our task is to find the pixel satisfying the condition: $D_i{}^* = \max\limits_{1 \leq j \leq d}(\mathrm{D}_i)$.

Put x_i into the result set V according to the following equations: $V = V \cup x_i$ and $X = X \backslash x_i$. If $X = \emptyset$ given a set of candidate points V, otherwise the process is repeat to find D_i .

If V is large, then we can proceed with this algorithm to reduce the number of candidate clusters. The calculations can be speeded up by dividing the input data set into subsets, and then the algorithm can be applied for each subset to find candidates set v_i. Call V is the set of all candidates, $\cup v_i = V$, apply this algorithm on the V set. The initial centroids can be initialized by choosing V according to the density of candidates.

Algorithm 2.1 Density-based fuzzy clustering algorithm (DFCM)

Input: Data set X with n data sample $X = \{x_1, x_2, ..., x_n\}, x_k \in R^d, k = \overline{1, n}$, the number of clusters is C, stop condition ϵ.
Output: Set of result clusters
Step 1. Calculate sample expectations and standard deviations by Equation 2.1 and 2.2, the radius of the sphere $r = \min\limits_{1<i<d}(s_i)$ in the m-dimensional space.
Step 2. Density calculation D_i by Equation 2.3.
Step 3. Find x_i by $D_i{}^* = \max\limits_{1\le i\le n}(\mathrm{D}_i)$, and assign x_i to result set by $V = V \cup x_i$ and $X = X\backslash x_i$.
Step 4. Calculate $Y = \{\mathrm{x}_j, \mathrm{r} - |x_i - x_j| \geq 0\}$ and set $X = X\backslash Y$. If $X = \emptyset$ the go to Step 5, else go to Step 1.
Step 5. Given set of centroids $V = \{v_1, v_2, ..., v_C\}$.
Step 6. Use the fuzzy clustering algorithm to cluster with the initial centroids just found.

The computational complexity of the proposed method includes the complexity of finding the initial centroid and the complexity of the clustering algorithm. The proposed algorithm needs c loops to finding the original centroids with the computational complexity being $O(ndc)$. The computational complexity of FCM algorithm is $O(ndcT_{max})$. DFCM algorithm will have a computational complexity of $O(ndcT_{max})$. With T_{max} is the maximum number of loops of FCM algorithm, d is the number of data dimensions.

2.2.2 Experiments

In the experiments, the authors have selected the problem of classification on RS image to test the proposed algorithm. The accuracy of the proposed method is compared with the clustering results by the algorithms k-Means, DBSCAN, and FCM. In that, step 1 is the initial pre-processing step, select the processing area on the satellite image and image geometry correction. Implement DFCM algorithm to classifying the SPOT-5 and Landsat-7 ETM+ multispectral images into six clusters (see Algorithm 1.6).

Table 2.1: The various validity indices computed from Landsat-7 ETM+ image

Index	***MSE***	***IQI***	***SSE***	***D-I***	***CS-I***
k-Means	0.2413	0.2843	147.9054	0.1873	0.9819
DBSCAN	0.1721	0.4183	111.7842	0.4762	0.5877
FCM	0.0982	**0.5643**	86.4691	0.4648	0.4769
DFCM	**0.0981**	0.5631	**69.4386**	**0.6742**	**0.2852**

Table 2.2: The various validity indices computed from SPOT-5 image

Index	***MSE***	***IQI***	***SSE***	***D-I***	***CS-I***
k-Means	0.3283	0.1987	132.9342	0.2654	1.2766
DBSCAN	0.1982	0.5762	109.7648	0.6811	0.3978
FCM	0.1098	0.6731	79.7632	0.7428	0.4991
DFCM	**0.0963**	**0.6984**	**65.9823**	**0.7829**	**0.3618**

In this study, to evaluate the quality of clusters, we considered the different validity indices, including MSE, IQI, SSE, D-I, and CS-I. These indices are calculated without the fuzzy membership information. It can be observed that the accuracy of clustering results using k-Means algorithm was very low. Many objects, such as bare soil and water, bare soil and sparse vegetation were misclassified. The accuracy of land cover clustering was improved when using DBSCAN and FCM algorithms; however, it was not sufficiently high. The results of calculation of MSE, IQI, SSE, D-I, and CS-I indices by four algorithms k-Means, DBSCAN, FCM and DFCM were shown in Table 2.3 and 2.2. It is apparent that the DFCM algorithms provided better clustering result than other algorithms, such as k-Means, FCM and DBSCAN.

The computational complexity of DBSCAN is $O(n^2 d T_{max})$. While the computational complexity of the k-Means algorithms, FCM and DFCM is $O(ndcT_{max})$, where n is the number of pixels.

DFCM algorithm enables the generation of initial centroids based on density. However, with clusters that are too close to or overlapping each other, this algorithm proves ineffective. In the next section, the dissertation will introduce a new algorithm based on spectral clustering and information about the spatial relationship between pixels.

2.3 Spatial-spectral fuzzy c-mean clustering

2.3.1 Proposed method

In image analysis, the key to determine a pixel belonging to a specific area is based on the similarity of these colors, which is calculated through a distance function in the color space $d_{ij} = \|x_i - x_j\|$ e.g. Euclidean distance between the pattern x_i and x_j. The shape and structure of the cluster also have a certain influence on clustering results. This means that together with information about color, the local information of pixels also needs to be considered when clustering.

Spectral clustering is a clustering method that uses the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction before clustering the data in fewer dimensions. We use a mask of size nxn to position on the image; the center pixel of the mask is the considered pixel. The number of neighbouring pixels P is determined according to the selected type of mask size, i.e. 8 pixels for mask $3x3$, 24 pixels for mask $5x5$, 48 pixels for mask $7x7$, so on.

To determine the degree of influence of the neighboring pixels for the center pixels, a local information measure M_i is defined on the basis of the distance $\|x_i - x_j\|$ and the attraction distance r_{ij}:

$$M_i = \sum_{j=1}^{P} \left(\|x_i - x_j\| \, r_{ij}\right)^{-1} / \sum_{j=1}^{P} r_{ij}^{-1} \tag{2.4}$$

In which $\|x_i - x_j\|$ is the distance of the all neighboring element x_j on the mask to the cluster x_i. The distance attraction r_{ij} is the squared Euclidean distance between elements (x_i, y_i) and (x_j, y_j) regarding their positions on the mask. According to the above expression, local information of each pixel comes with a higher value if its color is similar to the color of neighboring pixels and vice versa. We use the inverse distance r_{ij}^{-1} because the closer the neighbors x_j of the center x_i are the more influence they exert on the result and vice versa.

The idea behind the use of this spatial relationship information can be outlined as follows: Considering the local nxn mask and for sliding the mask on the image. Calculating the spatial information of the center pixel x_i based on the location of the center pixel x_i with the pixels x_j in the mask and the distance in color space $\|x_i - x_j\|$.

This aims to reduce the effect of noise on the image. From the above description, this method of similarity measure fully considers the local information and can eliminate the influence of the image noise.

Set $r = max(r_{ij})_{\forall i,j}$ is the radius of the largest circle in which pixels that affect the central pixel. Next, without loss of generality, we standardized similar measurements on the following equation:

$$\overline{M}_i = \frac{M_i - \min\,(M_i)_{\forall i}}{max(M_i)_{\forall i} - \min\,(M_i)_{\forall i}} \tag{2.5}$$

From above description, a new similarity measure is defined as follows:

$$s_{\mathrm{ij}} = \exp\left(-\frac{d^2(\mathrm{x}_i,\mathrm{x}_j)}{r^2}\right) \tag{2.6}$$

where s_{ij} shows pairwise similarities between pixels x_i and x_j; $d(x_i, x_j) = \|x_i - x_j\|$ is the Euclidean distance between x_i and x_j; r is the radius of the largest circle in which pixels affect the central pixel. Similarity matrix s_{ij} is usually constructed according to the equation 2.6.

With degree matrix D, it is established by adding local spatial information of each pixel, the degree of each pixel, d_i, is computed with:

$$d_i = \overline{M}_i * \sum_j s(i,j) \tag{2.7}$$

From the above description, the new Laplacian matrix L_{new}, is constructed using the new similarity matrix S and new degree matrix D:

$$L_{new} = D^{-1/2} S D^{-1/2} \tag{2.8}$$

Figure 2.1 is a schematic diagram of the implementation steps of IFCM algorithm.

The main steps of the proposed method are given as follows:

Algorithm 2.2 Improved fuzzy c-means algorithm (IFCM)

Input: Matrix size used to calculate local spatial information, number of clusters c, ϵ, T_{max}, $t = 0$.
Output: Clustering results $C_1, C_2, ..., C_c$ with $C_i = \{\mathrm{x_j}|\mathrm{u}_{ij} \in \mathrm{c}_i\}$. Evaluate accuracy, assign color to layers, and display results.
Step 1. Calculate local information measure M_i by Equation 2.5.
Step 2. Calculate a new similarity matrix S by Equation 2.6.
Step 3. Calculate a diagonal degree matrix D by Equation 2.7.
Step 4. Calculate a new matrix L_{new} by Equation 2.8.
Step 5. Find the c eigenvectors $\{e_1, e_2, ..., e_c\}$ of L_{new}, associated with the c highest eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_c\}$ and define the c dimensional space $Y = (y_i)_{i=1,...,n} \in R^c$.
Step 6. Running FCM algorithm on new space
6.1 t++
6.2 Calculates the function value u_{ij} by Equation 1.4.
6.3 Update centroids $c_i, i = 1, ..., c$ by Equation 1.3.
6.4 Calculate the J function value.
6.5 Checks the stop condition: If $max\{\left\|J^{(t+1)} - J^{(t)}\right\|\} \leq \varepsilon||(t > T_{max})$ go to **Output**, otherwise return to **Step 6**.

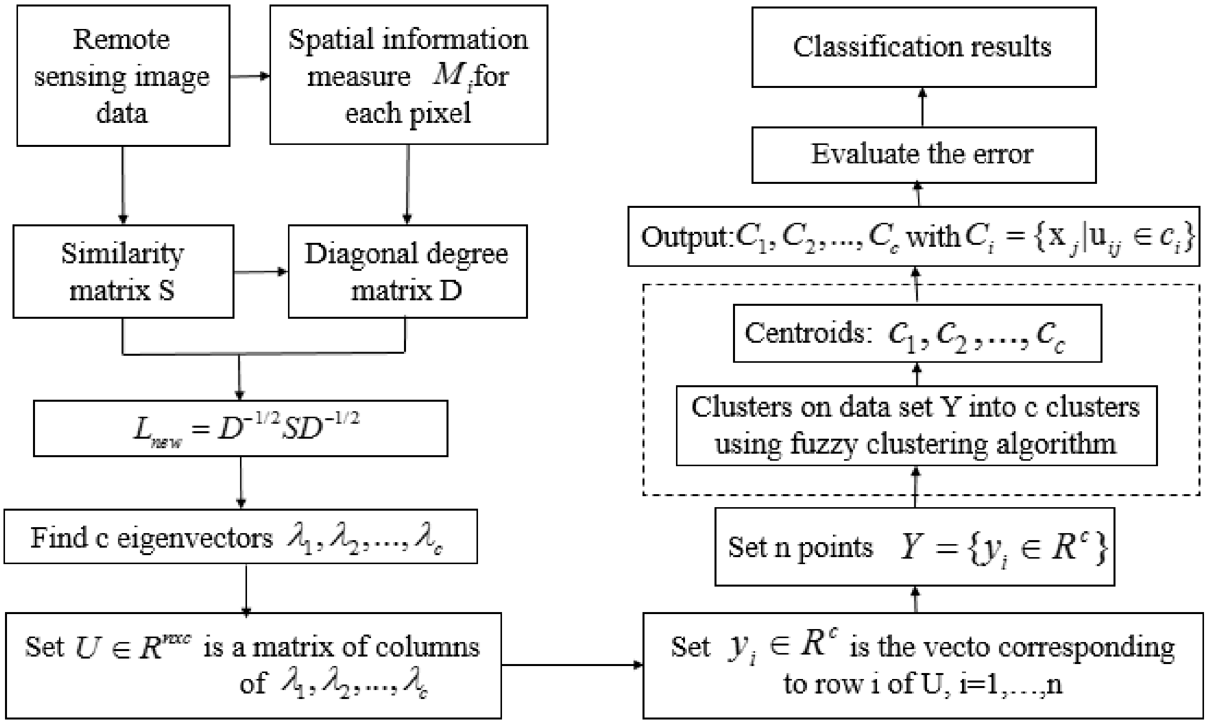


Figure 2.1: Diagram of the implementation steps of IFCM algorithm

With this approach, the new algorithm will overcome the challenges in the selection of parameters σ and pepper salt noise reduction in the image. This can increase the accuracy of clustering results by the spectral clustering algorithm.

It is a flexible class of clustering algorithms that can produce high-quality clusterings on small data sets, but has limited applicability to solve large scale problems due to its computational complexity of $O(n^3d)$. The computational complexity of determining spatial information is $O(nd)$ and FCM algorithm is $O(ndcT_{max})$. Thus, the computational complexity of IFCM algorithm is $O(n^3d)$.

2.3.2 Experiment

The test data is Landsat-7 ETM+ remote sensing image taken in Hanoi central area in 2009[1], coordinates from ($105^0 38' 38.8289" E$, $21^0 07' 5.3254" N$) to ($105^0 58' 53.5268" E$, $20^0 58' 14.9711" N$) with an area of $564.13 km^2$ (see Figure 2.2).

Figure 2.2: Results of land-cover classification in Hanoi area, FCM (a), ISC (b), IFKM (c) and the IFCM (d)

The results of land cover classification were shown in Figure 2.2, in which Figure 2.2(a, b, c and d) are classification results of FCM, ISC, IFKM and IFCM proposed algorithm, respectively. Whereas, IFKM is the improvement algorithm of fuzzy k-means [Pub 1].

[1] https://earthexplorer.usgs.gov/

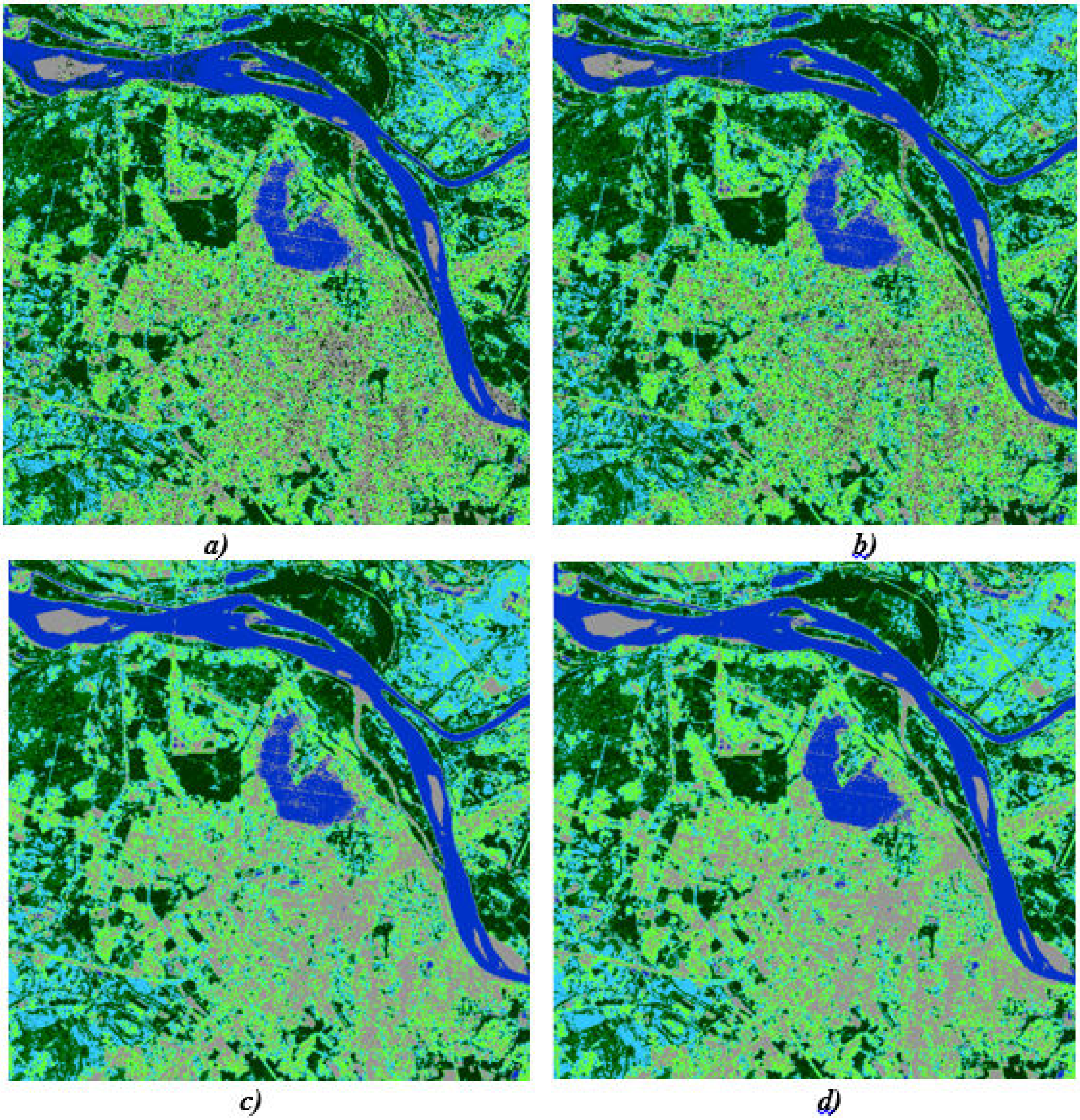


Figure 2.3: Remote sensing image in Hanoi center

Table 2.3: Performance of the FCM, ISC, IFKM and the IFCM algorithms

Index	*FCM*	*ISC*	*IFKM*	*IFCM*
MSE	0.1469	0.1497	0.1483	**0.1108**
IQI	0.7290	0.7362	0.7879	**0.9022**
DI	0.1022	0.1043	0.1085	**0.1279**
CSI	1.0081	0.8772	0.7139	**0.4887**
SSE	32.5587	31.1285	22.3652	**18.8745**

In this study, to evaluate the quality of clusters, we considered the different validity indices, such as MSE, IQI, DI and CSI. The calculation results of IQI, MSE, DI and CSI indices by four algorithms FCM, ISC, IFKM and IFCM were shown in Table 2.3. The proposed IFCM algorithm gives the best results in all five indicators.

The advantages of the proposed algorithms is clearly demonstrated with its capability of reducing noise on the image. Test results show that proposed algorithm has high segmentation accuracy and significantly reduces the computational complexity of classical spectral clustering algorithm and through experimental results, according to the visual and validity indexes MSE, IQI, DI and CSI, basically IFCM for sharper image quality, better noise reduction.

2.4 Application

2.4.1 SAR image segmentation

The proposed method is tested on SAR images. In particular, SAR image of the oil spill area on the Gulf of Mexico in 2010.

Synthetic Aperture Radar (SAR) used to obtain high-resolution images from broad areas of terrain [81]. SAR is capable of operating under inclement weather conditions, day or night. SAR images have wide applications in RS and mapping of the surfaces of both the Earth and other planets [63]. There are many other applications for this technology. from environmental monitoring, earth-resource mapping, surveillance and targeting information to military operations, oil spill classification [114], etc. However, SAR image clustering represents a major challenge in RS applications due to the influence of the speckle noise (see Fig. 2.4). Therefore, conventional methods will not be inefficient with speckle noise. To test the proposed algorithm, the SAR image is used to classify the oil spill on the sea.


(a)

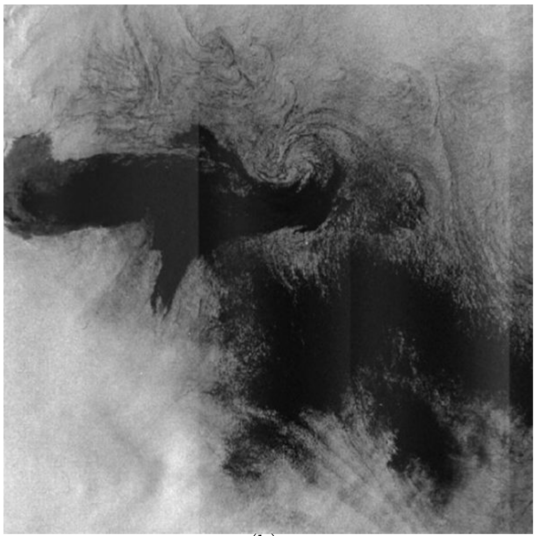

(b)

Figure 2.4: Spill oil area on Envisat ASAR image in Gulf of Mexico (a) $26April2010$, (b) $29April2010$

Test data include Asar Envisat images taken of a spill oil area in Gulf of Mexico on $26April2010$ (2.4a) and 29 April 2010 (2.4b), with coordinates ($0^{0}14'02.75"N$, $0^{0}03'56.39"E$ to $0^{0}04'27.33"N$, $0^{0}22'13.94"E$), covering an area of 23.32 hectare. Oil stains can be easily recognized on Fig. 2.4a with clearer boundaries, whereas in Fig. 2.4b, the contrast between the surrounding waters and the boundaries of oil stains is not clear, many parts showing mixed areas of water and oil stains because the oil stains have long existed on the sea.

Classification results are shown in Fig. 2.5 in the Gulf of Mexico on $26April2010$, FCM, ISC [48], DFCM and IFCM algorithm with Fig. 2.5a, Fig. 2.5b, Fig. 2.5c and Fig. 2.5d, respectively. A high level of noise exists on Fig. 2.5a, Fig. 2.5b and Fig. 2.5c, especially, on Fig.

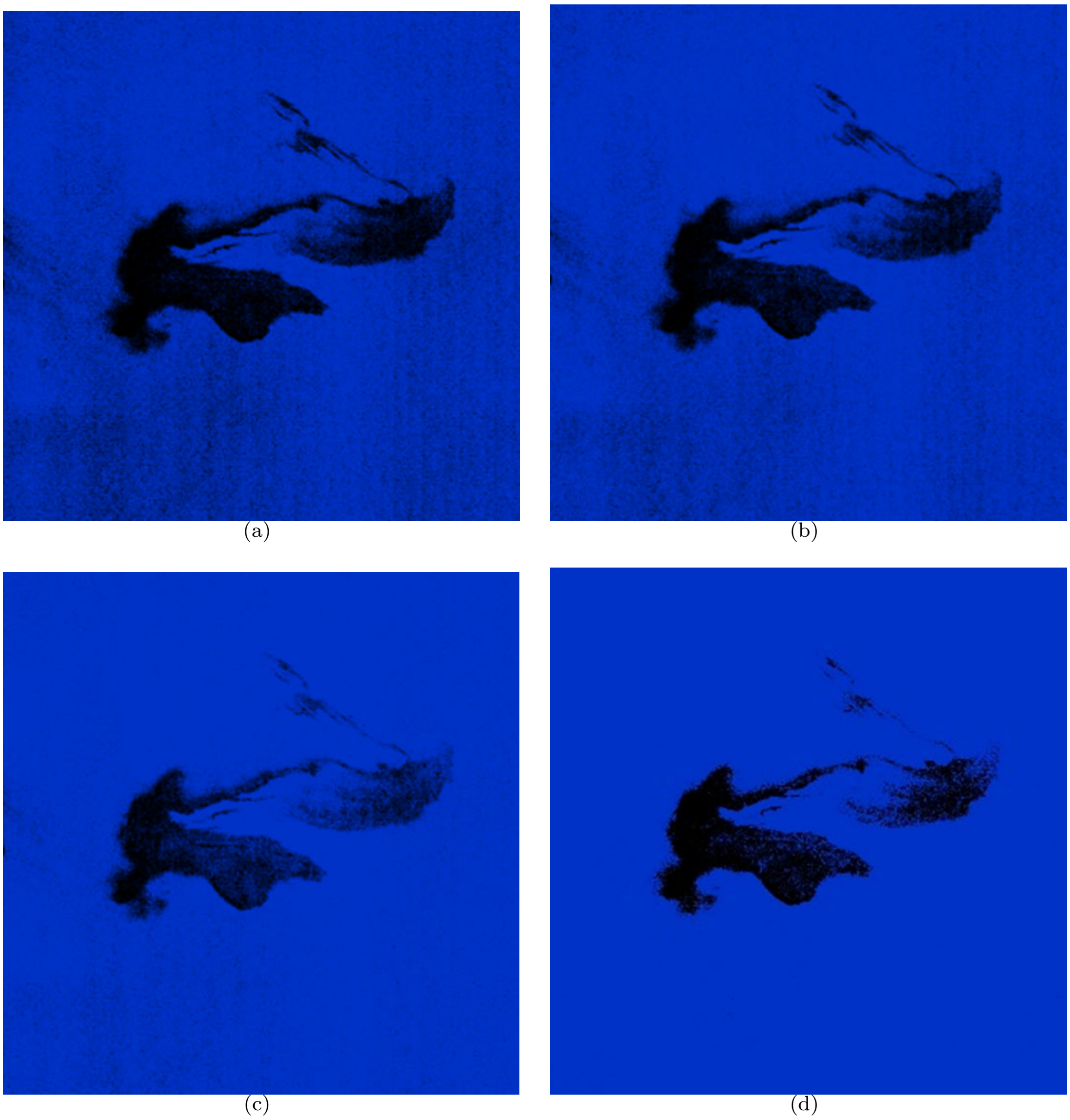


Figure 2.5: Oil spill classification results from the Envisat ASAR image in Gulf of Mexico on 26*April*2010

2.5a. Classification results on Fig. 2.5d shows that the noise is almost nonexistent and the water layer spill area is also clear than other results.

Table 2.4: Indicators for evaluating oil stain classification results on 26*April*2010

Index	***FCM***	***ISC***	***DFCM***	***IFCM***
MSE	0.1871	0.1212	0.1189	**0.0986**
IQI	0.4595	0.7851	0.8876	**0.8968**
DI	0.0186	0.0561	0.0604	**0.0659**
CSI	1.1872	0.8725	0.7628	**0.6521**
SSE	32.7884	17.4663	16.4726	**15.3742**

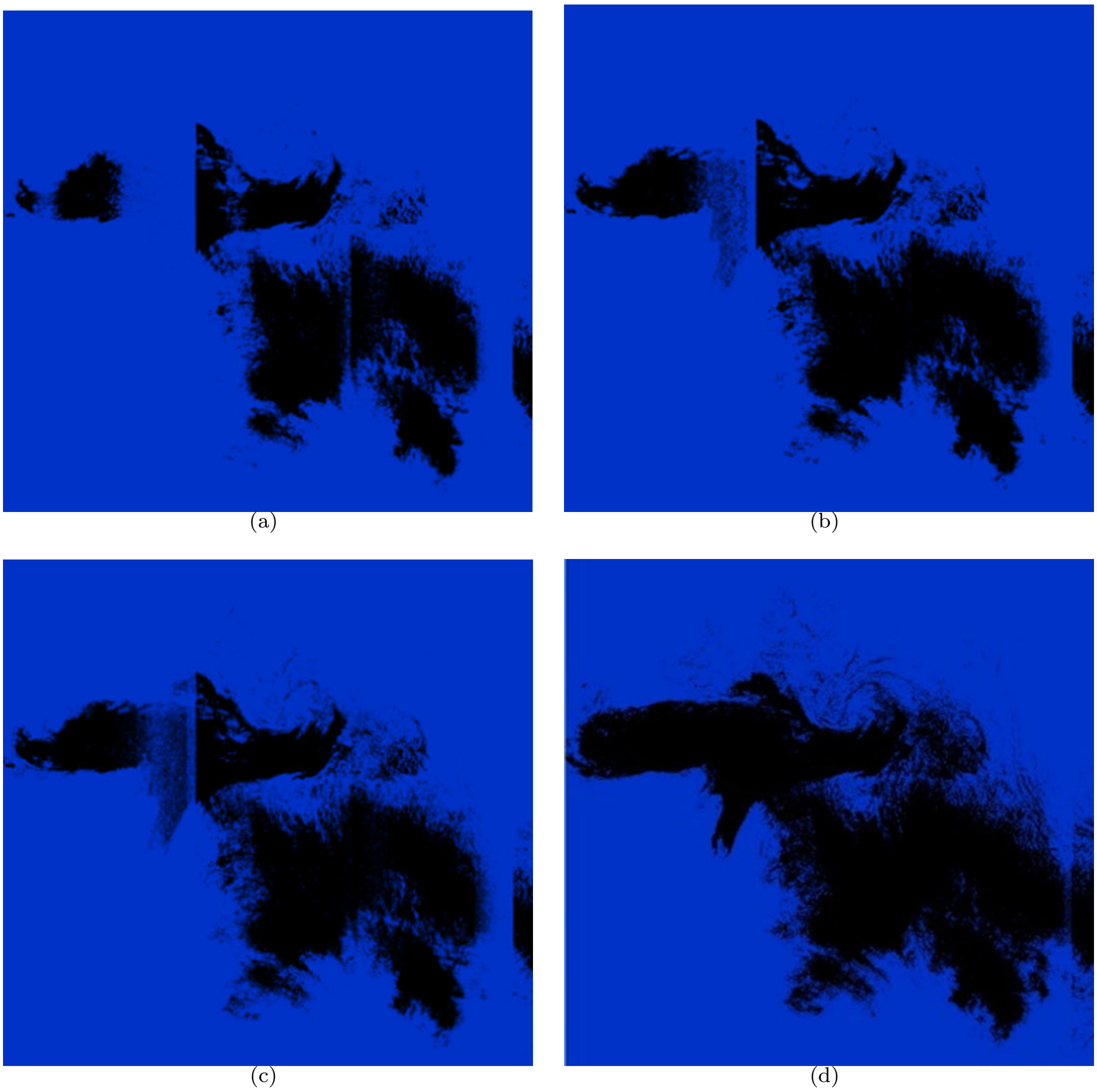


Figure 2.6: Oil spill classification results from the Envisat ASAR image in Gulf of Mexico on $29April2010$

Fig. 2.6 shown classification results in the Gulf of Mexico on $29April2010$ by algorithms FCM, ISC, DFCM and IFCM in Fig. 2.6a, Fig. 2.6b, Fig. 2.6c and Fig. 2.6d, respectively. It is apparent that a relatively high level of noise reduction has been achieved on all the results in Fig. 2.6 because SAR image data has relatively little pepper salt noise (see Fig. 2.6b). However, a large amount of information about the oil stains on the Fig. 2.6a, 2.6b and 2.6c are mistaken for noise, therefore undetectable. Test result with our algorithm on Fig. 2.6d shows that not only is noise reduction possible, but oil stains classification result is also more complete and clearer.

Table 2.5: Indicators for evaluating oil stain classification results on $29 April 2010$

Index	***FCM***	***ISC***	***DFCM***	***IFCM***
MSE	0.1761	0.1082	0.0864	**0.0082**
IQI	0.4862	0.6823	0.8635	**0.9447**
DI	0.0372	0.0598	0.0749	**0.0872**
CSI	1.5786	0.8873	0.7786	**0.5619**
SSE	15.6455	**8.4629**	8.4871	8.4631

Table 2.4 and table 2.5 indicate the value of the index assessing the quality of classification results. Overall, the proposed algorithm produced better results than the algorithms FCM, ISC and DFCM. Based on the value of this index, FCM algorithm for clustering result produced the poorest results, followed by algorithms ISC and DFCM.

2.4.2 Landcover classification

The second experiments are more visible and could be performed on multi-spectral RS images. The pixel information in these images is acquired from different temporal sensors. The proposed method is tested using Landsat 7-ETM+ images taken at Lam Dong province in the central-highland of Vietnam, see Fig. 2.7, ($107^{0}28'24.61"E$, $12^{0}13'04.66"N$ to $108^{0}54'52.39"N$, $11^{0}37'36.87"E$), with an area is of $9958.6km^2$ and a capacity of $116.89Mb$.

The image data are clustered to 6 classes as follows: Class 1: Surface water; Class 2: Bare land; Class 3: Grass, shrubs; Class 4: Planted forests, low woods; Class 5: Perennial tree crops; Class 6: Dense vegetation.

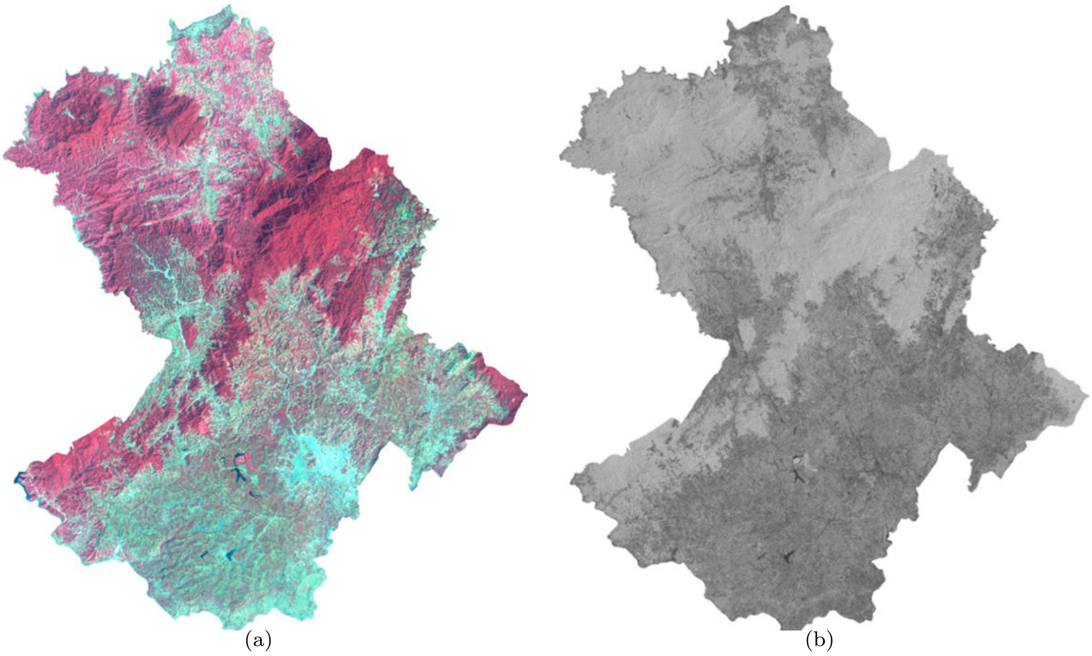


Figure 2.7: Landsat 7-ETM+ image of Lamdong area: a) Color Image; b) NDVI Image

To compare the proposed method with the previously studied methods, empirical tests were performed on four methods FCM, ISC, DFCM and IFCM (which is the proposed method in this dissertation). The output images are shown in Figure 2.8 below.

Table 2.6: Indicators for evaluating land-cover classification results of Lamdong area

Index	FCM	ISC	DFCM	IFCM
MSE	0.1763	0.1075	0.0982	**0.0918**
IQI	0.5623	0.6732	0.7849	**0.8721**
DI	0.0123	0.0365	0.0428	**0.0452**
CSI	1.2512	0.7784	**0.7750**	0.7751
SSE	98.6389	78.8599	52.8752	**46.3986**

To evaluate the accuracy of the proposed method, several validity indexes such as MSE, IQI, DI, CSI, and SSE were considered. Table 2.6 shows that IQI index and DI index are the largest with 0.8721 and 0.0452 on IFCM algorithm, while these figures for DFCM algorithm are 0.7849

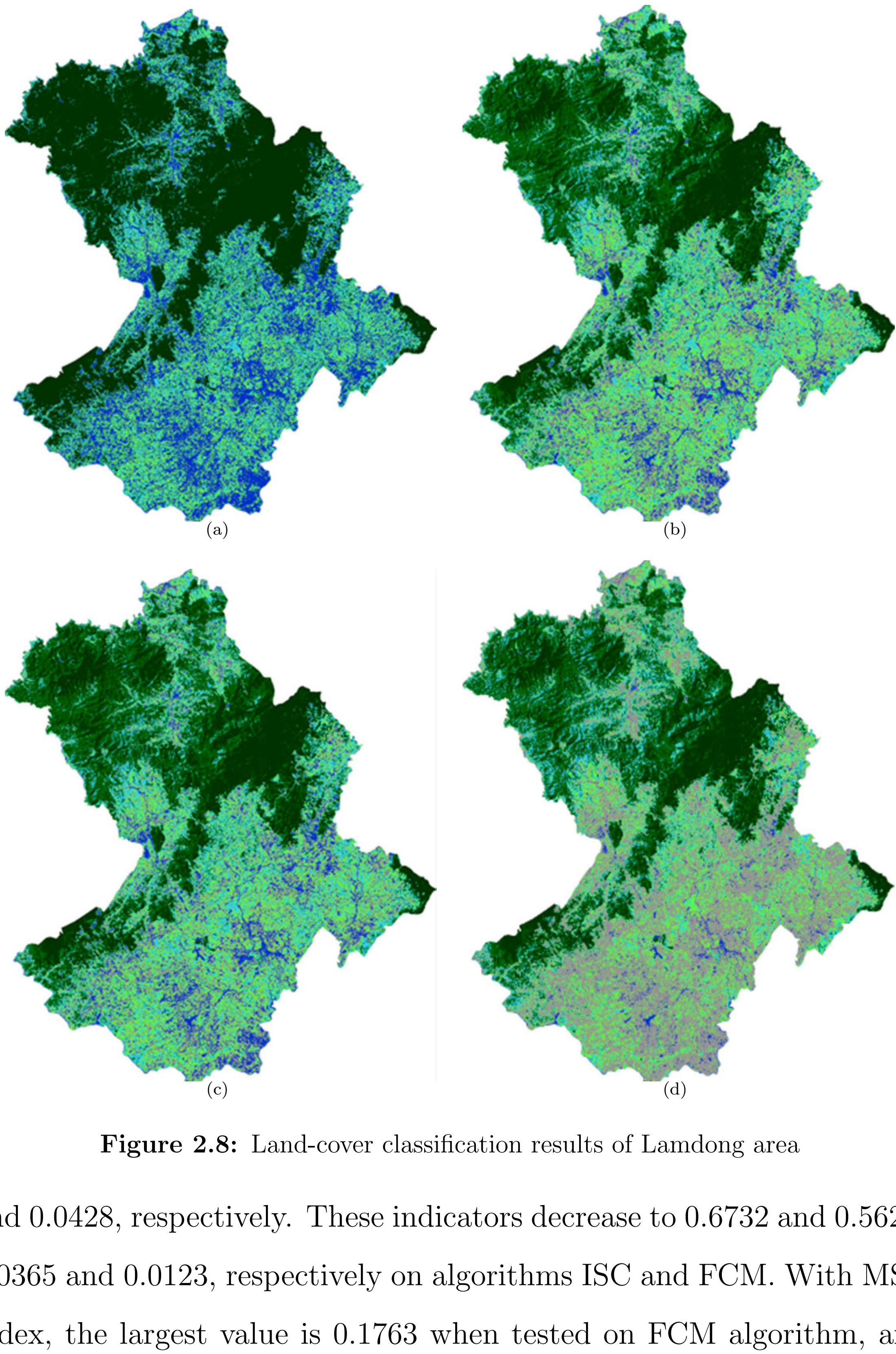


Figure 2.8: Land-cover classification results of Lamdong area

and 0.0428, respectively. These indicators decrease to 0.6732 and 0.5623; 0.0365 and 0.0123, respectively on algorithms ISC and FCM. With MSE index, the largest value is 0.1763 when tested on FCM algorithm, and descending on algorithms ISC, DFCM and IFCM. The lowest value is

0.0918 on IFCM algorithm. The smallest value of CSI index is 0.7750 with DFCM algorithm, while the values achieved by IFCM algorithm and ISC algorithm are 0.7751 and 0.7784, respectively. It is apparent that these variations are insignificant. Meanwhile, the CSI index with FCM algorithm is 1.2512. Thus, based on the value of the index clustering quality evaluation, in most cases, the IFCM algorithm was shown to have produced better clustering results than the algorithm DFCM, ISC and FCM did.

Despite possessing clear advantages over the original algorithms, two algorithms DFCM and IFCM are subject to serious drawbacks in that they are unsupervised and sensitive to clusters that are too close together.

2.5 Chapter summary

This chapter presents two algorithms DFCM, IFCM. The main idea of DFCM algorithm is to use density information as a preprocessing step to select initial centroids while IFCM algorithm is based on local information and spectral clustering to improve data separation, thereby making clustering easier and more accurate.

The proposed methods in this chapter were published in the International Journal of Fuzzy System Applications (**2019, Scopus, Q2**) [Pub7], Vietnam Journal of Science and Technology (**2018**) [Pub1], and the international conference ACIIDS (**2018**) [Pub3].

Although it has overcome some disadvantages of FCM algorithm, the implementation of the proposed algorithms is restricted by the difficulty in choosing parameters for fuzzy algorithms. Furthermore, as FCM clustering is based on fuzzy set type-1, they cannot fully describe the characteristics of the data, especially data with a high level of uncertainty.

In the next chapter, the dissertation presents the semi-supervised multiple kernel fuzzy c-means clustering algorithm and hybrid algorithms between fuzzy clustering and PSO technique to overcome this problem.

Chapter 3
IMPROVED FUZZY C-MEANS CLUSTERING ALGORITHMS WITH SEMI-SUPERVISION

In this chapter, the author presents three semi-supervised fuzzy clustering methods including SMKFCM, SFCM-PSO and GIT2SPFCM-PSO.

The SMKFCM algorithm combines labeled and unlabeled data to improve performance. The labeled patterns are used to calculate the centrality of clusters considered as the initial centroids, which are added to the objective functions.

The idea behind SFCM-PSO algorithm is to use the combination of SFCM and PSO techniques in the clustering process to determine the optimal parameters for each specific problem where SFCM is an extension of FCM, constructed by adding additional information to the clustering process.

Furthermore, from this approach, GIT2SPFCM-PSO algorithm has been extended from PFCM using type-2 fuzzy sets. The results show that they can handle uncertainty better than previous methods. Most of the experiments are applied to the problem of the land cover classification of RS images.

3.1 Introduction

In numerous clustering problems, highly complex shaped data represent a challenge in separating patterns. In the previous studies, kernel-based methods have exhibited the effectiveness in the partition of such data. This chapter also proposed a robust semi-supervised clustering method based FCM algorithm using multiple kernel technique [106], called SMKFCM, in which the initial centroids are directly used in the clustering process.

The hybridization of algorithms often aims to take advantage of the two previous methods to establish a new, more efficient ones. There are many optimization techniques and their variations. In RS image analysis, the semi-supervised technique can improve the accuracy of unsupervised fuzzy clustering due to the addition of some labeled data [51]. However, these algorithms are often subject to difficulty in choosing parameters and initial centroids [68]. The selection of fuzzy parameters and cluster centroids can be made by optimization techniques [72] such as PSO, GA and their variations. This dissertation chooses PSO as one of the techniques to hybridize. The advantage of PSO is that it is easier to install than the optimal algorithms of the genetic family. Although PSO does not guarantee convergence, but to reduce the risk, for each experiment, the we performed ten runs and selected the best results among them.

The results obtained from the SFCM-PSO algorithm serve as a suggestion for the authors to continue researching the hybrid method between the interval type-2 fuzzy clustering algorithm and PSO technique. Therefore, algorithm GIT2SPFCM-PSO is an extension and continuation of the research direction from the result of SFCM-PSO algorithm.

Moreover, land cover classification studies are still limited because most fuzzy clustering algorithms and their variants are unsupervised methods [102]. The clustering process only uses information about the spectral value of the object. Objects also have additional information such as information about spatial relationships, information on shape, information about labeled data, etc. Therefore spectral information does not adequately describe the object characteristics, and clustering results may not achieve the desired accuracy.

On the other hand, the goal of the clustering process is to optimize the objective function. Therefore, besides the use of additional information to propose a new and more suitable objective function, the hybrid of optimization techniques to select the optimal parameters will provide additional stability and efficiency to the algorithm.

SMKFCM algorithm can be used to both clustering and classification problems. The experimental results show that SMKFCM algorithm can improve the accuracy compared to the semi-supervised kernel fuzzy c-means (S2KFCM), the semi-supervised fuzzy c-means (SFCM) and the self-trained semi-supervised SVM algorithm (PS3VM).

In this chapter, hybrid approaches of fuzzy clustering and particle swarm optimization method based on the semi-supervised technique for RS imagery analysis (SFCM-PSO, GIT2SPFCM-PSO) is proposed to overcome the above disadvantages. The main idea of this method consists of two stages. Stage 1 involves the utilization of labeled data to refine the objective function by adding constraints, which can stabilize the algorithm and reduce the risk of falling into local optimization. Stage 2 uses PSO technique to determine the optimal parameters. This technique is integrated into the clustering process. Once this process ends the final parameters will be the optimal parameters. This study can also be used in cases where very little data has been labeled.

Experiments on different types of RS images show that the proposed methods can significantly improve the accuracy of classification results compared to the original methods.

3.2 Semi-supervised multiple kernel fuzzy c-means clustering

3.2.1 Semi-supervised kernel FCM clustering

Land cover maps are a vital input variable to many types of environmental research and management. While they can be produced automatically by machine learning techniques, to achieve high levels of accuracy, these techniques require substantial training data which are not always available.

In most cases, fuzzy clustering algorithms will determine the prototype of clusters depending on the structure of patterns. When a small number of patterns in the entire datasets could be labeled, the semi-supervised clustering algorithms are implemented on the combination of the labeled and unlabeled data to improve performance.

In the proposed method, the labeled data was used to calculate the rudimentary centroid of clusters, denoted by V^*. The idea of the approach is to use the rudimentary centroids V^* to adjust centroids to move closer to V^* by extending to semi-supervised kernel FCM in feature space (SKFCM-F).

The measure of the difference between the rudimentary clusters and the final clusters are determined as follows:

$$\left\|\phi(v_i^*) - \phi(v_i)\right\|^2 = k(v_i^*, v_i^*) + k(v_i, v_i) - 2k(v_i, v_i^*) \tag{3.1}$$

The distance d_{ij} between the pattern x_j and the prototype v_i in the kernel space is computed as follows:

$$\left\|\phi(x_j) - \phi(v_i)\right\|^2 = k(x_j, x_j) + k(v_i, v_i) - 2k(x_j, v_i) \tag{3.2}$$

The prototype v_i are constructed in the kernel space so it obtains the objective function as follow:

$$J_m(U, v) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{\mathrm{ij}}^m \left(\left\|\phi(x_j) - \phi(v_i)\right\|^2 + \left\|\phi(v_i^*) - \phi(v_i)\right\|^2\right) \tag{3.3}$$

In which u_{ij} satisfies the constraint $\sum\limits_{i=1}^{c} u_{\mathrm{ij}} = 1$, n is the number of patterns, c is the number of clusters. When minimizing the objective function, Lagrange multiplier is used to find the solution by the following function:

$$L(u_{\mathrm{ij}}, \lambda_j) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{\mathrm{ij}}^m \left(\left\|\phi(x_j) - \phi(v_i)\right\|^2 + \left\|\phi(v_i^*) - \phi(v_i)\right\|^2\right) + \sum_{j=1}^{n} \lambda_j (1 - \sum_{i=1}^{c} u_{\mathrm{ij}}) \tag{3.4}$$

Calculate the first derivative of function $L(u_{\mathrm{ij}}, \lambda_j)$ follow u_{ij} and v_{i}:

$$\begin{aligned}
&\Delta_{u_{\mathrm{ij}}} L(u_{\mathrm{ij}}, \lambda_j) = 0\\
&m.u_{\mathrm{ij}}^{m-1}(\left\|\phi(x_j) - \phi(v_i)\right\|^2 + \left\|\phi(v_i^*) - \phi(v_i)\right\|^2) - \lambda_j = 0\\
&2m.u_{\mathrm{ij}}^{m-1}(2 - k(x_j, v_i) - k(v_i^*, v_i)) - \lambda_j = 0\\
&u_{\mathrm{ij}} = \left(\frac{\lambda_j}{2m(2 - k(x_j, v_i) - k(v_i^*, v_i))}\right)^{1/(m-1)}
\end{aligned} \tag{3.5}$$

With the constraint $\sum\limits_{i=1}^{c} u_{\mathrm{ij}} = 1$ we have:

$$u_{\mathrm{ij}} = \left(\frac{\frac{1}{2(2 - k(x_j, v_i) - k(v_i^*, v_i))}}{\sum\limits_{j=1}^{c}\left[\frac{1}{2(2 - k(x_j, v_i) - k(v_i^*, v_i))}\right]^{1/(m-1)}}\right)^{1/(m-1)} \tag{3.6}$$

$$v_{\mathrm{i}} = \frac{\sum_{j=1}^{n} u_{\mathrm{ij}}^{m}[k(x_j, v_i)x_j + k(v_i^*, v_i)v_i^*]}{\sum_{j=1}^{n} u_{\mathrm{ij}}^{m}[k(x_j, v_i) + k(v_i^*, v_i)]} \tag{3.7}$$

With Gaussian kernel $k(x, y) = \exp(-\|x - y\|^2/r^2)$, it obtains the equations of membership function 3.6 and prototypes 3.7. Following are the detailed steps of SKFCM-F algorithm.

Algorithm 3.1 Semi-supervised kernel fuzzy c-means clustering (SKFCM-F)

Input: Given a set of n patterns $X = \{\mathrm{x}_i\}_{i=1}^{n}$ and the desired number of clusters c, number of loops T_{max}.
Output: Membership matrix $U = \{\mathrm{u_{ij}}\}_{i,j=1}^{n,c}$.
Step 1: Estimate the rudimentary centroids from labeled data.
1.1 Extract labeled patterns from dataset.
1.2 Calculating centroid $V^* = \{\mathrm{v}_i^*\}, \mathrm{v}_i^* \in R^n$ from the labeled patterns.
Step 2: Initialization
2.1 Choose fuzzifier $m, (1 < m)$, error ϵ.
2.2 Initialization membership matrix $U_{\mathrm{ij}}^{(0)}$.
Step 3:
3.1 t++;
3.2 Update centroids $V_j = [\mathrm{v}_{j1}, \mathrm{v}_{j2}, ..., \mathrm{v}_{jc}]$ by using Equation 3.7.
3.3. Compute the membership matrix $U_{\mathrm{ij}}^{(t)}$ by Equation 3.6.
3.4 **IF** $max(|U^{(t)} - U^{(t-1)}|) < \varepsilon$ or $t > T_{max}$ **THEN** go to step 4 **ELSE** go to Step 3.
Step 4: Report results clustering.
4.1 Return (t) and assign a pattern to a cluster.
4.2 Report results of clustering.

3.2.2 Semi-supervised multiple kernel FCM clustering

It should be noted that kernel-based methods depend on the usage of a suitable kernel function. If the kernel method only selects a single kernel from a predefined group, then it is sometimes not insufficient to represent all datasets. Besides, individual features of the selected input data can result in different clusters corresponding to individual kernels. Therefore, combining multiple kernels from a set of basis kernels has been proposed to better refine clusters rather than using a single kernel method. The most important key in the kernel method is how to use the formulation of suitable kernel function [106], [20], [29], [96]. Thus, kernel fuzzy clustering algorithms are necessary to be extended with the aggregation of kernel functions from different sources. The additional information of centroids was also added to the objective function to adjust the centroids through the iterative computing process.

The kernel-based methods deal with the difficulty in the combination or selection of the best kernels among the extensive possibilities. This combination is often strongly influenced by prior knowledge about the data and by the patterns which are discovered. Besides, many real-world clustering problems often contain many useful features when combined together. Therefore, it is necessary to use multiple kernels with their weights to aggregate for features from different sources into a final kernel function. A semi-supervised multiple kernel fuzzy c-means clustering (SMKFCM) algorithm is extended from SKFCM-F by combining different kernels to obtain better results.

SMKFCM maps the data from the feature space into kernel space H by using transform functions: $\psi = \{\psi_1, \psi_2, ..., \psi_M\}$ where $\psi_k(x_i)^T\psi_k(x_j) = K_k(x_i, x_j)$ and $\psi_k(x_i)^T\psi_{k'}(x_j) = 0 \,|k \neq k'$

The prototypes v_i is constructed in the kernel space, the general framework of SMKFCM aims to minimize the objective function like SKFCM-F function:

$$J_m(U, v) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{\mathrm{ij}}^m \left(\|\psi(x_j) - \psi(v_i)\|^2 + \|\psi(v_i^*) - \psi(v_i)\|^2\right) \qquad (3.8)$$

In which, $\sum\limits_{i=1}^{c} u_{\mathrm{ij}} = 1$, n is the number of patterns, c is the number of clusters, $\psi(x) = \omega_1\psi_1(x) + \omega_2\psi_2(x), ..., \omega_M\psi_M(x)$.

Subject to $\omega_1 + \omega_2 + \omega_M = 1$ and $\omega_k \geq 0, \forall k$, where v_i is the centroid of the i^{th} cluster in the kernel space, $(\omega_1, \omega_2, ..., \omega_M)$ is a vector of weights for features, respectively. The distance d_{ij} concerns the j^{th} data (pattern) and thei^{th} prototype:

$$\|\psi(x_j) - \psi(v_i)\|^2 = (\psi(x_j) - \psi(v_i))^T(\psi(x_j) - \psi(v_i)) \tag{3.9}$$

Some learning algorithms could automatically adjust the weights ω_k on a typical kernel learning method like multiple-kernel regression and classification which have been studied. Here, the dissertation proposes a similar algorithm for SMKFCM using linearly combined kernels in the typical kernels such as Gaussian kernel and polynomial kernel by introducing the Lagrange term of the constraint of weights into the objective function, defined as follows:

$$\begin{aligned} L(v_i, u_{\mathrm{ij}}, \omega_k) &= \sum_{i=1}^{c}\sum_{j=1}^{n} u_{\mathrm{ij}}^m \left(\|\psi(x_j) - v_i\|^2 + \|v_i^* - v_i\|^2\right) \\ &+ \sum_{j=1}^{n} \lambda_j (1 - \sum_{i=1}^{c} u_{\mathrm{ij}}) + \sum_{j=1}^{n} \beta_j (1 - \sum_{k=1}^{M} \omega_k) \end{aligned} \tag{3.10}$$

with α_j, β_j are constants. Optimizing the objective function 3.10 is expressed as:

$$\frac{\partial L(v_i, u_{\mathrm{ij}}, \omega_k)}{\partial v_i} = 0, \frac{\partial L(v_i, u_{\mathrm{ij}}, \omega_k)}{\partial u_{\mathrm{ij}}} = 0, \frac{\partial L(v_i, u_{\mathrm{ij}}, \omega_k)}{\partial \omega_k} = 0 \tag{3.11}$$

Solving the system of Equations 3.11 gives:

$$v_i = \sum_{j=1}^{n} u_{\mathrm{ij}}^m (\psi(x_j) + v_i^*) \Bigg/ 2\sum_{j=1}^{n} u_{\mathrm{ij}}^m \tag{3.12}$$

$$u_{ij} = \frac{\left(\frac{1}{m((\psi(x_j)-v_i)^2 + (v_i^* - v_i)^2)}\right)^{1/(m-1)}}{\sum\limits_{i=1}^{c}\left(\frac{1}{m((\psi(x_j)-v_i)^2 + (v_i^* - v_i)^2)}\right)^{1/(m-1)}} \tag{3.13}$$

$$\omega_k = \frac{\beta_j + 2\sum\limits_{i=1}^{c} u_{\mathrm{ij}}^m v_i \psi_k(x_j)}{2\sum\limits_{i=1}^{c} u_{\mathrm{ij}}^m \psi_k^T(x_j)\psi_k(x_j)} \tag{3.14}$$

With $\omega_1 + \omega_2 + \omega_M = 1$ and after some mathematical transformations we have:

$$\beta_j = 2\sum_{k=1}^{M}\sum_{i=1}^{c} u_{\text{ij}}^m \psi_k^T(x_j)\psi_k(x_j)\left(1 - \sum_{k=1}^{M}\frac{\sum\limits_{i=1}^{c} u_{\text{ij}}^m v_i \psi_k(x_j)}{\sum\limits_{i=1}^{c} u_{\text{ij}}^m \psi_k^T(x_j)\psi_k(x_j)}\right) \tag{3.15}$$

Now it can calculate the distance d_{ik} concerning the j^{th} data and the i^{th} prototype as:

$$d_{\text{ij}}^2 = \|\psi(x_j) - \psi(v_i)\|^2 = \psi^T(x_j)\psi(x_j) - 2\psi(x_j)\psi(v_i) + \psi^T(v_i)\psi(v_i) \tag{3.16}$$

By replacing the v_i in Equation 3.12 and $\psi^T(x)\psi(y) = K(x,y) = \sum\limits_{k=1}^{M}\omega_k k_k(x,y)$ in the above equations, we have:

$$\begin{aligned} d_{\text{ij}}^2 = {} & \sum_{k=1}^{M}\omega_k^2 K_k(x_j,x_j) - \frac{\sum\limits_{k=1}^{M}\sum\limits_{j=1}^{n} u_{\text{ij}}^m \omega_k^2 (K_k(x_j,x_j) + K_k(x_j,v_i^*))}{\sum\limits_{j=1}^{n} u_{\text{ij}}^m} \\ & + \frac{\sum\limits_{k=1}^{M}\sum\limits_{j=1}^{n} u_{\text{ij}}^{2m}\omega_k^2 (K_k(x_j,x_j) + 2K_k(x_j,v_i^*) + K_k(v_i^*,v_i^*))}{\left(\sum\limits_{j=1}^{n} u_{\text{ij}}^m\right)^2} \end{aligned} \tag{3.17}$$

$$\beta_j = 2\sum_{k=1}^{M}\sum_{i=1}^{c} u_{\text{ij}}^m K_k(x_j,x_j)\left(1 - \sum_{k=1}^{M}\frac{\sum\limits_{i=1}^{c}\sum\limits_{j=1}^{n} u_{\text{ij}}^m (K_k(x_j,x_j) + K_k(x_j,v_i^*))}{2\sum\limits_{j=1}^{n}\sum\limits_{i=1}^{c} u_{\text{ij}}^m K_k(x_j,x_j)}\right) \tag{3.18}$$

$$d_{\text{ij}}^2 = \frac{\beta_j + \sum\limits_{i=1}^{c}\sum\limits_{j=1}^{n} u_{\text{ij}}^m (K_k(x_j,x_j) + K_k(x_j,v_i^*))}{2\sum\limits_{j=1}^{n} u_{\text{ij}}^m K_k(x_j,x_j)} \tag{3.19}$$

To construct multi-kernel, we consider Gaussian kernel as K_1 and Polynomial kernel as K_2:

$$K_1(x,y) = \exp(-\|x-y\|^2/r^2), K_2(x,y) = (x^T y + d)^p \tag{3.20}$$

Where $r, d \in R^+, p \in N^+$.

Algorithm 3.2 Semi-supervised multiple kernel fuzzy c-means (SMKFCM)

Input: Given a set of n patterns $X = \{\mathrm{x}_i\}_{i=1}^{n}$, a set of kernel functions $\{\mathrm{K}_k\}_{k=1}^{M}$, and the desired number of clusters c, number of loops T_{max}.
Output: Membership matrix $U = \{\mathrm{u_{ij}}\}_{i,j=1}^{n,c}$ and weights $\{\omega_k\}_{k=1}^{M}$ for the kernels. To construct multiple kernels, we use the Gaussian kernel as K_1 and Polynomial kernel as K_2.
Step 1: Estimating centroids from the labeled data
1.1 Extracting the labeled patterns from the dataset.
1.2. Calculating the rudimentary centroids $V^* = [\mathrm{v}_i^*], \mathrm{v}_i^* \in R^n$ from labeled patterns.
Step 2: Initialization
2.1 Choose fuzzifier $m, (1 < m)$, error ϵ.
2.2 Initialize membership matrix $U^{(0)}$.
Step 3:
3.1 t++;
3.2 Calculate constants β_j by using Equation 3.18.
3.3 Update weights ω_k by using Equation 3.14.
3.4 Calculate the distance in kernel space d_{ij} by using Equation 3.19.
3.5 Update memberships $U^{(t)}$ by using Equation 3.13.
3.6 Verify if the termination condition is satisfied:
IF $(|U^{(t)} - U^{t-1}|) < \epsilon$ or $t > T_{max}$ **THEN** go to step 4 **ELSE** go to step 3.
Step 4: Report results clustering.
4.1. Return (t) and ω_k with $k = 1, 2, ...,$.
4.2. Assign a pattern to a cluster and report the results of clustering.

Following are the detailed steps of SMKFCM algorithm.

The computational complexity of SMKFCM is $O(n^2dcM)$ per iteration with M is the multiplier used.

3.2.3 Experiments

The proposed method has been experimented on a number of different data sets including RS image data and data from the UCI Machine Learning Repository library[1]. Details are presented in experiments 1 and 2. The parameters and terminal conditions: The number of iterations $L = 30$ and the error $\epsilon < 0.00001$, the fuzzy parameter m is set to 2. Set $\delta^2 = 4$ in kernel K_1 and $d = 10$ and $p = 2$ in kernel K_2.

Experiment 1

The first experiment, was implemented on the well-known datasets from UCI, consisting of Urban Land Cover (ULC) with 168 instances and

[1] https://archive.ics.uci.edu/

148 attributes, Landsat Satellite Data Set (LSDS) with 6435 instances and 36 attributes, Forest Type Mapping (FTM) with 326 instances and 27 attributes. To evaluate the classification results, we deployed the previous algorithms such as SFCM [51], S2KFCM [99], and PS3VM [82] to compare with the proposed algorithms (SMKFCM and SKFCM-F).

The aim of the classification, ULC data is to distinguish between three classes of water, ponds, lakes; plants and buildings, roads. FTM data is to distinguish three types of forest, and LSDS dataset consists of the multi-spectral values of pixels in $3x3$ neighbourhoods in a satellite image, and the classification associated with the central pixel in each neighbourhood. The aim is to predict this classification, given the multi-spectral values.

The classifiers were performed 30 times with the averages calculated as the final results. The labeled data accounted for 5%, 10%, 15% and 30% of the dataset ULC, LSDS, and FTM, respectively.

Table 3.1: Classification results by the algorithms SFCM, S2KFCM, PS3VM, SKFCM-F and SMKFCM

Data	***Rate(%)***	***SFCM***	***S2KFCM***	***PS3VM***	***SKFCM-F***	***SMKFCM***
ULC	TPR	88.21 ± 3.12	92.86 ± 2.98	95.56 ± 1.56	93.16 ± 2.42	$\mathbf{96.32 \pm 1.32}$
c=3	FPR	4.31 ± 1.14	3.84 ± 1.85	1.61 ± 0.92	1.38 ± 0.63	$\mathbf{1.15 \pm 0.46}$
	ACC	95.66 ± 1.52	96.56 ± 1.31	98.13 ± 1.04	97.81 ± 0.99	$\mathbf{98.34 \pm 0.12}$
LSDS	TPR	90.99 ± 3.92	92.45 ± 3.01	96.98 ± 0.98	94.24 ± 2.69	$\mathbf{97.79 \pm 1.02}$
c=2	FPR	3.98 ± 1.46	4.11 ± 1.19	1.07 ± 0.41	1.24 ± 0.65	$\mathbf{1.04 \pm 0.28}$
	ACC	96.68 ± 1.63	97.13 ± 1.62	98.57 ± 0.78	97.37 ± 1.04	$\mathbf{99.02 \pm 0.31}$
FTM	TPR	89.05 ± 2.92	93.27 ± 2.91	96.32 ± 1.02	93.18 ± 1.02	$\mathbf{96.48 \pm 0.97}$
c=3	FPR	3.19 ± 1.68	3.02 ± 1.14	1.23 ± 0.87	$\mathbf{1.09 \pm 0.68}$	$\mathbf{1.09 \pm 0.68}$
	ACC	96.67 ± 1.87	96.92 ± 1.32	99.03 ± 0.55	96.89 ± 1.28	$\mathbf{99.21 \pm 0.49}$

Table 3.1 shows that the correctly classifying ratios obtain the best values from SMKFCM algorithm on datasets ULC, FTM and LSDS. The SFCM algorithm produces the classes with the lowest rate.

On the datasets, LSDS, ULC and FTM, SMKFCM algorithm obtain the largest TPR of 97.79%, 96.32% and 96.48%, which are higher than the other algorithms. Meanwhile the FPR produced by the SMKFCM algorithm obtains the smallest values of 1.15% and 1.09%, respectively. For LSDS dataset, SMKFCM algorithm obtains the best TPR of 97.79% in comparison with 96.98% from PS3VM and the FPR of 1.02%, 1.07%, and 1.04% from SMKFCM.

For the Acc index, the proposed method results in more than 98% accuracy for all data sets. Specifically for ULC data set, Acc reached 98.34% compared with 95.66% to 98.13%; with LSDS data set, Acc reached 99.0%2 compared with 96.68% to 98.57%; for FTM data set, Acc reached 99.21% compared with the 96.67% to 99.03% of the other algorithms, respectively.

In summary, the experiment exhibits that SMKFCM obtains better ratios than algorithms including PS3VM, SKFCM-F, S2KFCM, and SFCM in the all cases on the considered datasets (ULC, FTM, LSDS).

Experiment 2

The second experiment involves the clustering problem on RS image. The dissertation has proposed using labeled data to adjust the centroid of the clusters. Besides, the use of kernel techniques allows for better data separation, which can help improve the accuracy of cluster results.

The second series of experiments involve multi-spectral RS images. The pixel information is composed of different bands as a feature vector. The different kernels for pixel intensities are defined by applying the combined kernel in a multiple-kernel learning algorithm. Landsat-7

ETM+ satellite image data in three different regions in 2014 was selected for the experiment (see Algorithm 1.6). The spatial resolution of the imagery is $30m$.

In this proposal, the data is divided into two sets consisting of the labeled and unlabeled components. The labeled data is used to estimate the centroid of clusters. These centroids are called the rudimentary centroids, denote $V^* = [\mathrm{v}_1^*, \mathrm{v}_2^*, ..., \mathrm{v}_c^*]$, which are used into processing the RS image clustering.

Hanoi capital area $(11^0 24' 02.32" N, 107^0 36' 26.74" E)$ to $(10^0 50' 24.61" N, 108^0 09' 50.57" E)$ with an area of $3161.304 km^2$. The two bands No. 3 and No. 4 are displayed in Fig. 3.1.

Bao Loc city area, Lam Dong province $(11^0 18' 29.13" N, 108^0 18' 10.57" E)$ to $(11^0 58' 29.63" N, 107^0 01' 44.93" E)$ with an area of $1707.31 km^2$ and the total number of pixels is $1,897,008$.

Thai Nguyen city area $(105^0 37' 16.0190" E, 21^0 37' 39.8284" N)$ to $(105^0 59' 49.7296"$ $21^0 28' 58.9896" N)$ with an area of $614.43 km^2$ and the total number of pixels is $411,045$.

The kernel-based proposal algorithm uses Gaussian kernel K_1 for pixel intensities and the multiple kernel-based proposal algorithm uses Gaussian kernel K_1 and polynomial kernel K_2 for pixel intensities. Thus, u_{ij} and V values can be calculated according to the equations 3.12 and 3.13 in SKFCM-F and u_{ij}, β_j, ω_k, d_{ij} according to the equations 3.14, 3.18 and 3.19 in Algorithm SMKFCM. The labeled data corresponds to around 5% to 10% for each class.

We use two bands No. 3 and No. 4 to compute NDVI index which is

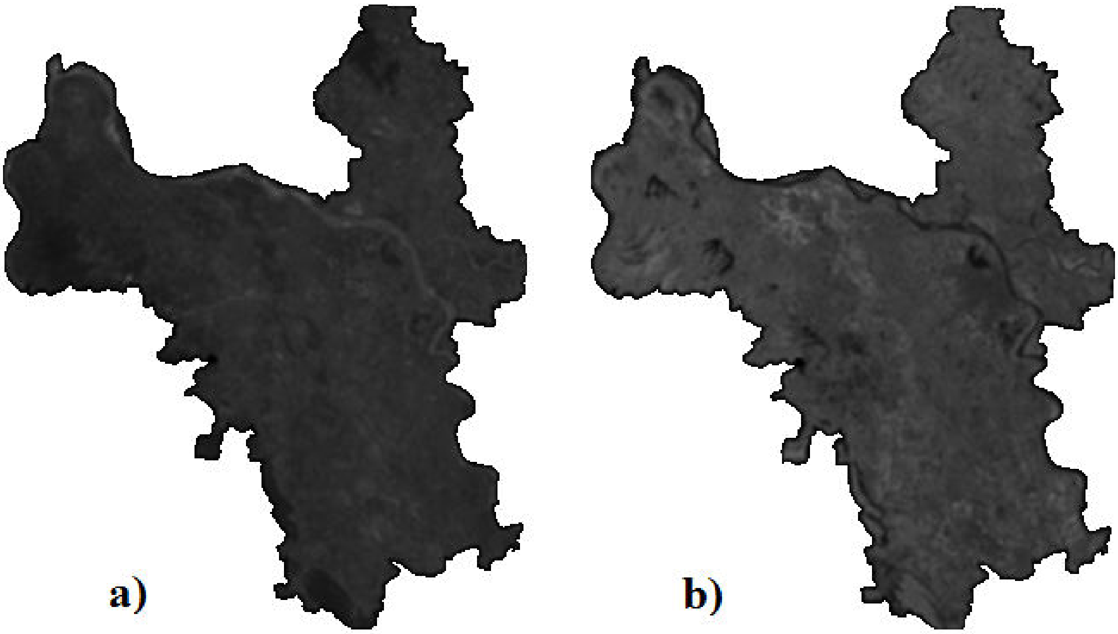


Figure 3.1: Landsat-7 ETM+ satellite image of Hanoi capital: a) Band 3 (RED); b) Band 4 (NIR)

the most common measurement to assess the growth and distribution of the vegetation on the earth's surface.

$$NDVI = \frac{N\mathrm{IR} - RED}{N\mathrm{IR}{+}RED} \tag{3.21}$$

In which, NIR (Near-Infrared) and RED (Visible Red) corresponding to band No. 3 and band No. 4 in the 7-bands of Landsat-7 ETM+ imagery, respectively. The value of NDVI index of a pixel assumes in the range $[-1, 1]$. The no-vegetable pixel takes an amount around zero. The value of almost 1.0 means the highest density of vegetables. Low values of NDVI (around 0.1) correspond to barren areas of rock, sand. Moderate values represent shrub and grassland (0.2 to 0.3), while high values indicate temperate and tropical rainforests (0.6 to 0.8). Conveniently, NDVI data is converted to the pixel-based image by the following equation:

$$P\mathrm{ixel}_{value} = (NDVI + 1) * 127 \tag{3.22}$$

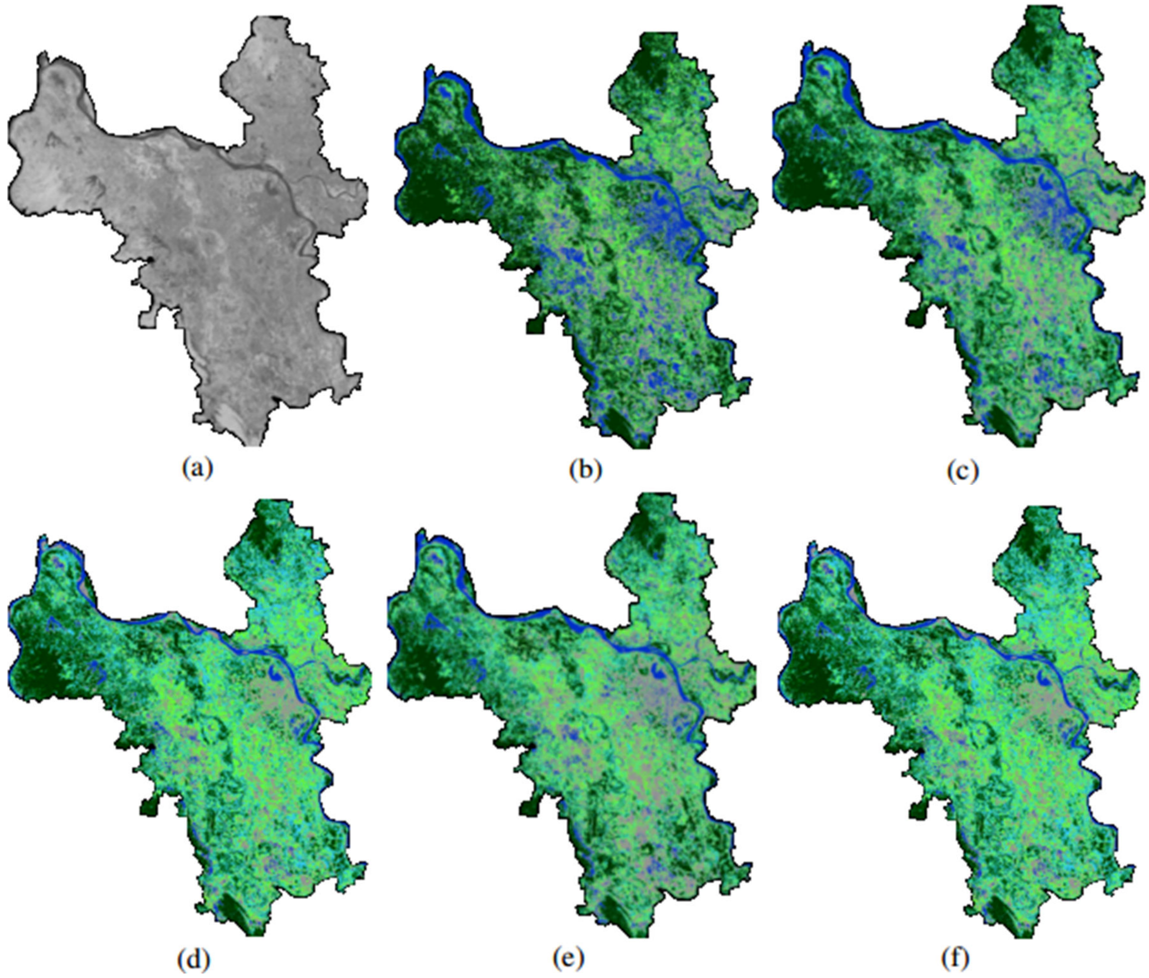


Figure 3.2: Land-cover classification results of Hanoi capital (a) NDVI Image; (b) SFCM; (c) S2KFCM; (d) PS3VM; (e) SKFCM-F; (f) SMKFCM.

We have implemented classification on the different algorithms such as SFCM, S2KFCM, PS3VM, SKFCM-F and SMKFCM.

To estimate the performance, we considered several validity indexes such as the PC-I, D-I, S-I, XB-I, and CE-I [8], [14], [110], [55]. Note that the validity indexes are proposed to evaluate the quality of clustering. Algorithms producing better results are associated with smaller values of D-I, S-I, CE-I, XB-I and the larger value of PC-I.

The experimental results are shown in Fig. 3.2 in which (a), (b), (c), (d), (e) and (f) are NDVI image, the classification results of SFCM,

Table 3.2: Land-cover classification result of Hanoi area by SMKFCM algorithm

Class	*The number of pixels*	*Percentage (%)*	*Square (hectares)*
1	211 491	6.021	19034.211
2	663 663	18.894	59729.678
3	888 081	25.283	79927.249
4	689 270	19.623	62034.268
5	652 001	18.562	58680.125
6	408 054	11.617	36724.869

S2KFCM, PS3VM, SKFCM-F, and SMKFCM, respectively. Table 3.2 and Fig. 3.3 shows the detailed classification produced by SMKFCM algorithm according to the number of pixels, the percentage and the area of the individual classes.

Table 3.3 shows the comparative results between SFCM, S2KFCM, PS3VM, SKFCM-F and SMKFCM and the data of the Vietnam National Remote Sensing Center (VNRSC) on each class (in percentage %). The significant difference between the algorithm SFCM, S2KFCM, PS3VM, SKFCM-F and SMKFCM is utilized to determine the area of regions; the largest difference is around 11%. Comparing the experimental results with VNRSC data, we also consider the accuracy by estimating the percentage of the class with the lowest deviation, the largest deviation and the average deviation of six layers (in percentage %). The largest differences produced by SFCM, S2KFCM, SKFCM-F, PS3VM are 10.78%, 7.743%, 3.77%, 1.925%, respectively. Meanwhile, the value obtained by SMKFCM algorithm is below 1.0% (see Table 3.4).

Besides, the validity indices in Table 3.4 show that the proposed algorithm obtains better results than other algorithms in most cases. The validity indices produced by SMKFCM algorithm produces better values than the ones run by different algorithms, i.e., the PC-I, IQI, CE-I, XB-

Table 3.3: Land-cover classification results of Hanoi area by some algorithms and VNRSC data

Class	*VNRSC*	*SMKFCM*	*SKFCM-F*	*PS3VM*	*S2KFCM*	*SFCM*
1	5.833%	6.021%	8.361%	6.389%	9.784%	11.234%
2	19.665%	18.894%	16.178%	18.169%	13.175%	10.763%
3	25.041%	25.283%	27.883%	25.517%	32.784%	35.821%
4	19.857%	19.623%	17.234%	19.273%	14.768%	12.428%
5	17.702%	18.562%	21.472%	19.627%	22.313%	23.274%
6	11.903%	11.617%	8.874%	11.026%	7.176%	6.481%

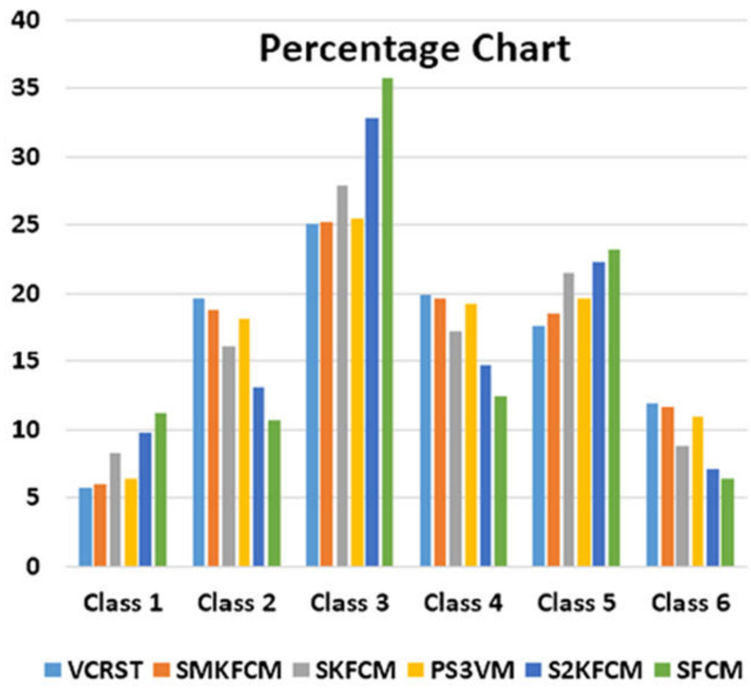


Figure 3.3: Hanoi area: Land-cover classification results by percentage (VNRSC data, SMKFCM, SKFCM-F, PS3VM, S2KFCM and SFCM)

Table 3.4: The various validity indexes for Hanoi area

Algorithm	*SFCM*	*S2KFCM*	*PS3VM*	*SKFCM-F*	*SMKFCM*
PC-I	0.5821	0.6314	0.7216	0.6464	**0.7376**
D-I	0.1125	0.2548	0.1093	0.5422	**0.5711**
IQI	0.2398	0.4877	0.6982	0.6745	**0.8759**
CE-I	0.9973	0.9287	0.6241	0.8872	**0.5098**
XB-I	3.8734	2.0831	1.7034	1.9652	**1.6912**
CS-I	2.7658	**1.0346**	1.6983	1.7685	1.0645
MSE	12.4676	9.7728	6.9822	5.8271	**5.8269**
SSE	37.8758	29.8721	25.9833	24.8856	**18.8734**
Min deviation	5.401	3.951	0.476	2.528	**0.188**
Max deviation	10.780	7.743	1.925	3.770	**0.860**
Average deviation	7.2510	5.4352	0.9857	3.0463	**0.4302**

I, MSE, and SSE reach values of 0.7376, 0.8759, 0.5098, 1.6912, 5.8269, and 18.8734, respectively. The PC-I value is lower and higher values of IQI, CE-I and XB-I correspond to PS3VM, SKFCM-F, S2KFCM, and SFCM. While, the value D-I index obtains the largest value of 0.5711 by running SMKFCM algorithm, followed by SFCM, SKFCM-F, S2KFCM, and PS3VM with values of 0.1125, 0.2548, 0.5422, 0.1093, respectively. CS-I index achieves the best value of 1.0346 with S2KFCM algorithm.

Table 3.5: Land-cover classification results for Bao Loc area

Class	***VNRSC***	***SMKFCM***	***SKFCM-F***	***PS3VM***	***S2KFCM***	***SFCM***
1	2.08%	2.13%	2.37%	2.49%	3.00%	4.37%
2	10.26%	11.23%	13.83%	11.10%	15.36%	17.50%
3	18.76%	19.81%	21.09%	21.09%	22.76%	25.07%
4	26.84%	27.82%	30.15%	27.94%	31.90%	36.64%
5	23.20%	21.53%	15.31%	18.63%	14.14%	9.95%
6	18.87%	17.48%	17.25%	18.76%	12.83%	6.47%

Table 3.6: The various validity indexes on the Landsat-7 images in Bao Loc

Algorithm	***SFCM***	***S2KFCM***	***PS3VM***	***SKFCM-F***	***SMKFCM***
PC-I	0.4896	0.6482	0.7653	0.6976	**0.7784**
D-I	0.1061	0.3652	0.1076	0.4982	**0.6851**
IQI	0.5895	0.7196	**0.7972**	0.6298	0.7872
CE-I	0.9938	0.9492	0.7862	0.9273	**0.4762**
XB-I	4.0982	2.8723	1.6982	2.0874	**1.3723**
CS-I	1.3464	1.0873	1.0374	0.9983	**0.8927**
MSE	11.8777	8.2712	8.0724	7.9801	**7.8185**
SSE	40.9842	41.4526	32.7365	30.1553	**27.8274**
Min deviation	2.297	0.928	0.112	0.298	**0.054**
Max deviation	13.248	9.061	4.574	7.888	**1.667**
Average deviation	8.551	5.034	1.562	3.170	**1.019**

The results produced from the experiment on dataset No. 2 are demonstrated in Tables 3.5 and 3.6. Table 3.5 shows the percentages of the individual classes from algorithms of SFCM, S2KFCM, PS3VM, SKFCM-F and SMKFCM, and VNRSC, in which, the differences between VNRSC and SMKFCM assume the smallest values. In Table 3.6, the validity indices produced by SMKFCM algorithm are also the best values i.e, the larger values of PC-I, IQI, CE-I, and XB-I. While, the value D-I index obtains the largest value of 0.6851 by running SMK-FCM algorithm, followed by SKFCM-F, PS3VM, S2KFCM and SFCM with values of 0.4982, 0.3652, 0.1076, and 0.1061, respectively.

Table 3.7: Land-cover classification results for Thai Nguyen area

Class	***VNRSC***	***SMKFCM***	***SKFCM-F***	***PS3VM***	***S2KFCM***	***SFCM***
1	2.631%	2.722%	3.778%	3.247%	3.918%	5.702%
2	14.894%	15.592%	18.036%	15.989%	20.028%	22.813%
3	24.458%	25.253%	27.495%	25.979%	29.685%	31.939%
4	17.559%	17.469%	15.813%	16.723%	14.310%	12.827%
5	23.428%	22.869%	20.724%	21.937%	18.436%	15.251%
6	17.029%	16.095%	14.153%	16.125%	13.623%	11.467%

Table 3.8: The various validity indexes on the Landsat-7 images in Thai Nguyen

Algorithm	***SFCM***	***S2KFCM***	***PS3VM***	***SKFCM-F***	***SMKFCM***
PC-I	0.5287	0.5872	0.7987	0.6098	**0.8763**
D-I	0.1098	0.2987	0.1265	0.3987	**0.4272**
IQI	0.6898	0.7981	0.7982	**0.8733**	0.8721
CE-I	0.9652	0.8869	0.5091	0.7898	**0.4827**
XB-I	5.8936	4.0673	1.9033	2.6732	**1.2871**
CS-I	2.3827	1.5877	1.8724	0.8727	**0.3875**
MSE	14.8752	9.1763	9.7686	6.9924	**5.9982**
SSE	35.4982	31.9566	27.1536	26.1635	**13.8375**
Min deviation	3.071	1.286	0.616	1.147	**0.090**
Max deviation	8.177	5.227	1.521	3.142	**0.934**
Average deviation	6.157	3.882	1.077	2.442	**0.528**

In the experiment on dataset No. 3, the smallest differences of the individual classes also come with SMKFCM in Table 3.7. The indicators CS-I, MSE, SSE, CE-I and XB-I for SMKFCM algorithm obtain the lowest value of 0.3875, 5.9982, 13.8375, 0.4827 and 1.2871, respectively, in Table 3.8 and the PC-I also produce the best result of 0.8763. Deviation of percentage area from six classes, when compared to data collected from VNRSC, shows that the smallest average difference of only 0.528% for SMKFCM algorithm. In contrast, the differences of algorithms PS3VM, SKFCM-F, S2KFCM and SFCM are 1.077%, 2.442%, 3.882%, and 6.157%, respectively.

These results demonstrate that SMKFCM produces a better clustering solution than the other algorithms such as PS3VM, SKFCM-F, S2KFCM, and SFCM. With spatial resolution of $30m$, classification results can be accepted in the quick assessment of land covers, which is more cost-effective than the traditional methods. These results not only make predictions about the land cover changes but also support urban planning, natural resources management, etc.

A small amount of labeled data can improve the quality of the classification results. This section describes an approach on semi-supervised fuzzy clustering for satellite images using kernel technique and centroid information retrieved from the labeled data part.

The kernel techniques used, involve two cases: single kernel function for all data features and multiple-kernel functions for data features, i.e. spatial features and Landsat-band valued features. The proposed method improves the clustering results and overcomes the drawbacks of the conventional clustering algorithms. The experiments were conducted on the well-known datasets (ULC, LSDS, FTM) and RS image clustering on three datasets (Hanoi, Bao Loc and Thai Nguyen) in Vietnam.

3.3 Hybrid method of fuzzy clustering and PSO

3.3.1 Proposed method

Usually, the supervised clustering technique requires large amounts of labeled data for training. In cases where labeled data is limited; the method often used is a semi-supervised clustering method.

A_i is the set of pixels that have been labeled for the i^{th} cluster, with

$i = 1, ..., c$. Calculation c centroids by the following formula:

$$v_i^* = \sum_{j=1}^{|A_i|} p_j(A_i) / |A_i| \tag{3.23}$$

In which, $|A_i|$ is the number of labeled pixels for the i^{th} cluster, p_j is the j^{th} pixel in A_i.

The objective function J_m of FCM algorithm is changed as follows:

$$J_m = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^m [\mathrm{d}^2(\mathrm{v}_i,\mathrm{x}_k) + \mathrm{d}^2(\mathrm{v}_i, \mathrm{v}_i^*)], 1 < m < \infty \tag{3.24}$$

With $d(v_i, x_k)$ is the Euclidean distance between the pixel x_k and the cluster centroid v_i and $d(v_i, v_i^*)$ is the distance between the calculated cluster centroid and the desired cluster centroid, cluster results are good when this distance is small.

Minimize the objective function J_m, based on the Lagrange method:

$$J_m = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^m [\mathrm{d}^2(\mathrm{v}_i,\mathrm{x}_k) + \mathrm{d}^2(\mathrm{v}_i, \mathrm{v}_i^*)] + \sum_{k=1}^{n} \lambda_k \sum_{i=1}^{c} (1 - u_{ik}) \tag{3.25}$$

Minimize Lagrange function by computation of derivatives u_{ik} and v_i, we have:

$$u_{\mathrm{ik}} = \left[\frac{1/(d^2(v_i, x_k) + d^2(v_i, v_i^*))}{\sum\limits_{j=1}^{c} [1/(d^2(v_i, x_k) + d^2(v_i, v_i^*))]^{1/(m-1)}} \right]^{1/(m-1)} \tag{3.26}$$

$$v_i = \frac{\sum\limits_{k=1}^{n} u_{ik}^m (v_i^* + x_k)}{2 \sum\limits_{k=1}^{n} u_{ik}^m} \tag{3.27}$$

Subject to $0 < \sum\limits_{k=1}^{n} u_{\mathrm{ik}} < n; 0 \leq u_{\mathrm{ik}} \leq 1; \sum\limits_{i=1}^{c} u_{\mathrm{ik}} = 1; 1 \leq k \leq n; 1 \leq i \leq c.$

In clustering data, one of the criteria for evaluating results is the distance between the cluster centers. In this study, we propose a criterion for the minimum distance between cluster centers $\min_{i \neq j}\{d^2(v_i, v_j)\}$. A large value indicates that the clusters are more separated from each other.

Therefore, the dissertation proposes an objective function as follows:

$$F = \frac{\sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^{m}[\mathrm{d}^2(\mathrm{v}_i,\mathrm{x}_k)+\mathrm{d}^2(\mathrm{v}_i, \mathrm{v}_i^*)]}{\min_{i \neq j}\{\mathrm{d}^2(\mathrm{v}_i, \mathrm{v}_j)\}} \tag{3.28}$$

Equation 3.28 is used instead of equation 3.25. Clusters are useful when the numerator is small, and the denominator is large.

In PSO algorithm, particles never die (this is different from GA). Particles can be viewed as simple agents, passing through the search space and recording the best solution they discover. The optimization process of PSO can be accomplished through several steps as follows: Create an initial swarm, initialize location and velocity of particles; evaluate particles; update the location and velocity of the particles. An impor-

m	V_1	V_2	V_3	...	V_c

$\Downarrow$

m	v_{11}	v_{21}	v_{31}	...	v_{c1}
	v_{12}	v_{22}	v_{32}	...	v_{c2}
	....	....	....	....	....
	v_{1b}	v_{2b}	v_{3b}	...	v_{cb}

$\Downarrow$

m	v_1	v_2	v_3	...	v_{c*b}

Figure 3.4: The matrix represents the particles

tant point to consider is how particles is initialized, so it is necessary to define the structure of the particles. For multi-spectral image including b bands ($b = 3$ for color image), the number of cluster is c: $V_1; V_2; ...; V_c$

with $V_i = (v_{ij}), i = 1;...;c; j = 1;...;b$, the components are described in figure 3.4 following the conditions $m > 1$, $v_{min} \leq v_{ij} \leq v_{max}$ to limit the search space ($v_{min} = 0, v_{max} = 255$ for 8 bit image or $v_{max} = 65536$ for 16 bit image, etc). In the case of parameter fuzzy $m, 1 \leq m \leq 4$. After each update step of the algorithm, if $v_{ij} > v_{max}$ then $v_{ij} = v_{max}$, if $v_{ij} < v_{min}$ then $v_{ij} = v_{min}$. With $b * c$ components and fuzzy parameter m, the number of particles to be initialized is $b * c + 1$, see Fig. 3.4.

Typically, the position of the particles will be randomly generated in the search space, and the algorithm will perform a finite number of iterations of velocity and position updates. Updating the position adds velocity value. The velocity value represents the speed of movement of the particles. If velocity is too high, particles can move out of the search space. Conversely, if velocity is too small, particles are limited, and the optimum solution hence may not be achieved. Let vt_{max} and vt_{min} be the velocity limits of the particles, in which vt_{max} value and vt_{min} value are selected by experience. Velocity values are limited from vt_{min} to vt_{max}:

$$vt_{\max} = \frac{v_{\max} - v_{\min}}{2}, vt_{\min} = -\frac{v_{\max} - v_{\min}}{2} \tag{3.29}$$

A constraint is given, if $vt_i > vt_{max}$ then $vt_i = vt_{max}$, if $vt_i < vt_{min}$ then $vt_i = vt_{min}$, with $i = 1; 2; ...; c * b + 1$.

Two values which need consideration are P_{ibest} and G_{ibest}. P_{ibest} is the best solution that i^{th} particle has discovered so far. G_{ibest} is the best global solution, which means that G_{ibest} is the best solution found by the whole swarm. These values will be updated based on the optimization of the objective function F, and the process of moving the particles will change the value of the objective function F. In each iteration, if the movement of the particles optimizes the objective function F (the smaller objective function), then the location of the particle will be saved by P_{ibest}; the particle that causes the objective function F to reach the smallest value then the location of that particle will be saved by G_{ibest}.

An important issue in PSO algorithm is the selection of parameters. Parameters c_1 and c_2 represent the influence of the best particle solution and the best global solution. These two parameters are normally set to 2.05, as suggested in the original document of PSO algorithm [38]. The parameter ω is the inertial coefficient. This value indicates the rate of change in the velocity of the particle during moving. Common values range from zero to one. And $r_1; r_2$ are the random numbers in the range of $(0, 1)$.

Details of implementation steps of the hybrid approach of semi-supervised fuzzy clustering and particle swarm optimization method for RS imagery analysis (SFCM-PSO) are presented in algorithm 3.3.

Note that if the objective function 3.25 is optimized, SFCM-PSO algorithm becomes FCM-PSO algorithm. Compared to FCM algorithm, in SFCM-PSO algorithm, the calculation in steps 2.2, 2.4 and 2.5 is quite simple. The computing complexity of SFCM-PSO algorithm is similar to that of the FCM algorithm.

3.3.2 Experiments

The proposed method is tested on Landsat-8 OLI and SPOT-5 images. Experiments are also carried out on algorithms FCM [7], SFCM [51], FCM-PSO and SFCM-PSO. For the PSO algorithm, $c_1 = c_2 = 2.05$; $\omega =$ 0.9 and decrease to 0.1 when the maximum number of loops (generation

Algorithm 3.3 Semi-supervised fuzzy c-means algorithm (SFCM-PSO)

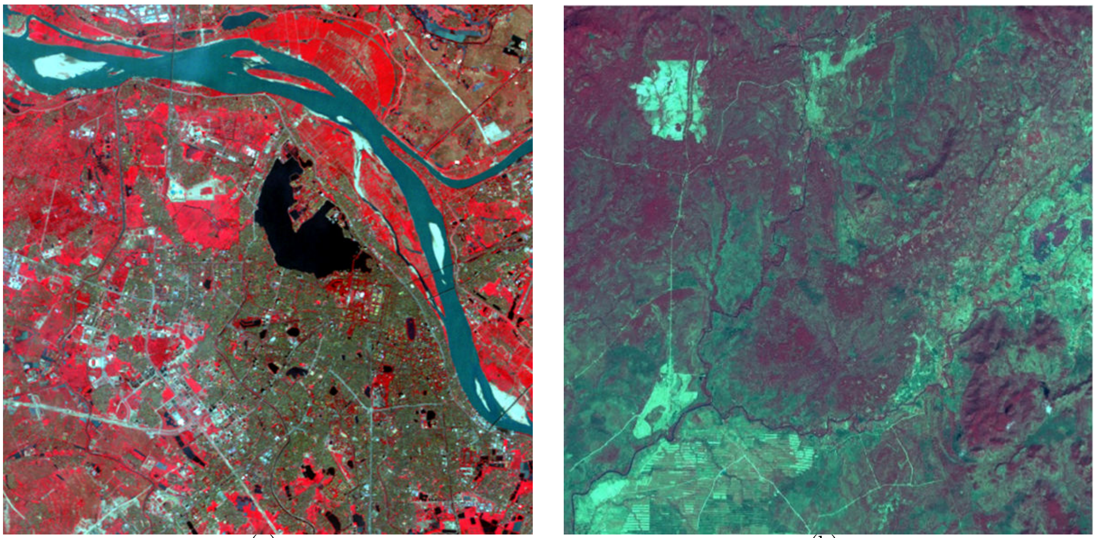
Input: Given a set of n samples $X = \{\mathrm{x}_i\}_{i=1}^{n}$, where $A = A_1 \cup A_2 \cup ... \cup A_c$ is the set of labeled data samples, $A_i, i = \overline{1,c}$ is a set of labeled data samples for class i.
Output: $U = [u_{ik}]$
Step 1: Initialize swarm
1.1 Calculation c centroids: $V^* = [\mathrm{v}_1, \mathrm{v}_2, ..., \mathrm{v}_c]$ by Equation 3.23.
1.2 Set the constants: Maximum loop number $T, t = 0, c_1, c_2, \omega, r_1, r_2, \varepsilon$.
1.3 Create random locations of particles $v_1^{(0)}; v_2^{(0)}; ...; v_{c*b}^{(0)}$ and $v_{c*b+1}^{(0)}$ $(m^{(0)})$ within the limits from v_{min} to v_{max}.
1.4 Create random velocity of particles: $vt_1^{(0)}; vt_2^{(0)}; ...; vt_{c*b}^{(0)}$ and $vt_{c*b+1}^{(0)}$ $(vt_m^{(0)})$ within the limits from vt_{min} to vt_{max}.
1.5 Calculate the value of U by Equation 3.26.
Step 2: Hybrid algorithm of semi-supervised fuzzy clustering and PSO
2.1 $t = t + 1$
2.2 $v_i^{(t)} = v_i^{(t)} + vt_i^{(t)}, i = 1, ..., c * b + 1$
2.3 Update F by Equation 3.28.
2.4 Update P_{ibest} and G_{ibest}.
2.5 $vt_i^{(t+1)} = \omega * vt_i^{(t)} + c_1 * r_1 * (P_{ibest} - v_i^{(t)}) + c_2 * r_2 * (G_{ibest} - v_i^{(t)}), i = 1, ..., c * b + 1$
2.6 Update the value of U by Equation 3.26.
2.7 If $\max(\left\| u_{ik}^{(t+1)} - u_{ik}^{(t)} \right\|) < \varepsilon$ or $(t > T)$ then go to step 3 else go to step 2.1.
Step 3: Finished
3.1 Given $U = [u_{ik}]$.
3.2 Defuzzification and assign pixels to the cluster: if $u_{ik} > u_{jk}$ for $j = 1; 2; ...; c$ and then x_k is assigned to cluster i.

number) is reached $T = 10000$. With FCM and SFCM, the maximum number of loops is set to $100, m = 2$ and experimental results were averaged over 10 runs of the algorithm.

RS imagery is the Landsat and Spot image. Fig. 3.5(a, b) displays the original images. These are two distinctive areas of land-cover, one of which is the city centre and the other is a mountainous area.

The data is clustered into 6 classes as follows: Class 1: Surface water ; Class 2: Bare land ; Class 3: Grass, shrubs ; Class 4: Planted forests, low woods ; Class 5: Perennial tree crops ; Class 6: Dense vegetation (see Algorithm 1.6).

The clustering results have been evaluated by some validity indexes, including PC-I, D-I, CE-I, MSE, CS-I, XB-I and IQI. Significant values are with indexes PC-I, IQI and D-I for good clustering results while



(a) (b)

Figure 3.5: Study datasets (a. Hanoi center area, b. Chu Prong area)

small values with indexes DB-I, CE-I, CS-I and XB-I for good clustering results [8], [14], [110], [55].

Experiment 1

Experimental data from Landsat-8 OLI image is the regional centre of Hanoi, Vietnam (see Figure 3.5a) with eight image bands, so the number of particles is 49. The size of each image band is $512x512$, and the number of pixels is $262,144$. The number of samples labeled is $7982; 327; 298; 309; 412$ and 278 for class 1; 2; 3; 4; 5 and 6 respectively. Test results on SFCM-PSO algorithm show that $m = 2.18642$, on FCM-PSO algorithm show that $m = 2.08265$ corresponding to the minimum value of the function F.

Fig. 3.6(a,b,c,d) shows land-cover classification results for Hanoi area by four algorithms including FCM, SFCM, FCM-PSO and SFCM-PSO, respectively. Detailed statistical data are shown in Table 3.9 and Table 3.10.

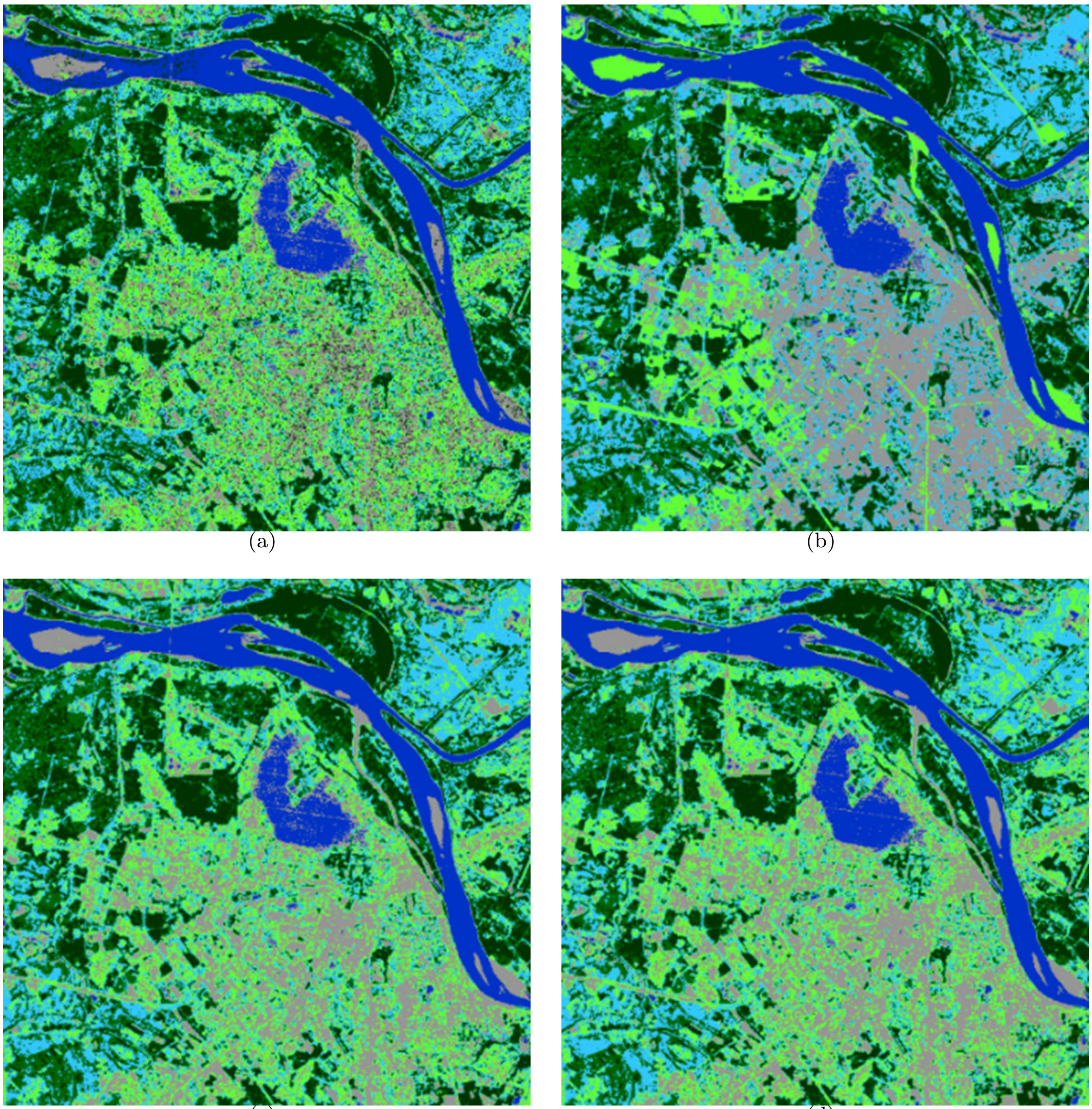

(a) (b) (c) (d)

Figure 3.6: Land-cover classification results of Hanoi city center

Table 3.9 shows that SFCM-PSO has better quality clustering than FCM, SFCM, FCM-PSO algorithms in most cases. SFCM algorithm gives the best clustering result at IQI index with a value of 0.877036; SFCM-PSO algorithm is 0.876242. SFCM-PSO algorithm gives better clustering results than other algorithms in the indexes PC-I, D-I, DB-I, CE-I, CS-I, MSE, and XB-I.

Table 3.10 shows the correct classification rate on labeled pixels. The

Table 3.9: Validity indices obtained for Hanoi area

Method	*XB-I*	*PC-I*	*CE-I*	*D-I*	*IQI*	*CS-I*	*MSE*
FCM	0.175231	0.687263	0.562283	0.198275	0.672641	0.037862	11.37661
SFCM	0.187632	0.779824	0.498472	0.276914	**0.877036**	0.088651	6.982757
FCM-PSO	0.157295	0.576231	0.389745	0.321874	0.862874	0.108743	6.257482
SFCM-PSO	**0.128746**	**0.782632**	**0.319768**	**0.348723**	0.876242	**0.128743**	**3.877244**

Table 3.10: Land-cover classification results by percentage of Hanoi area

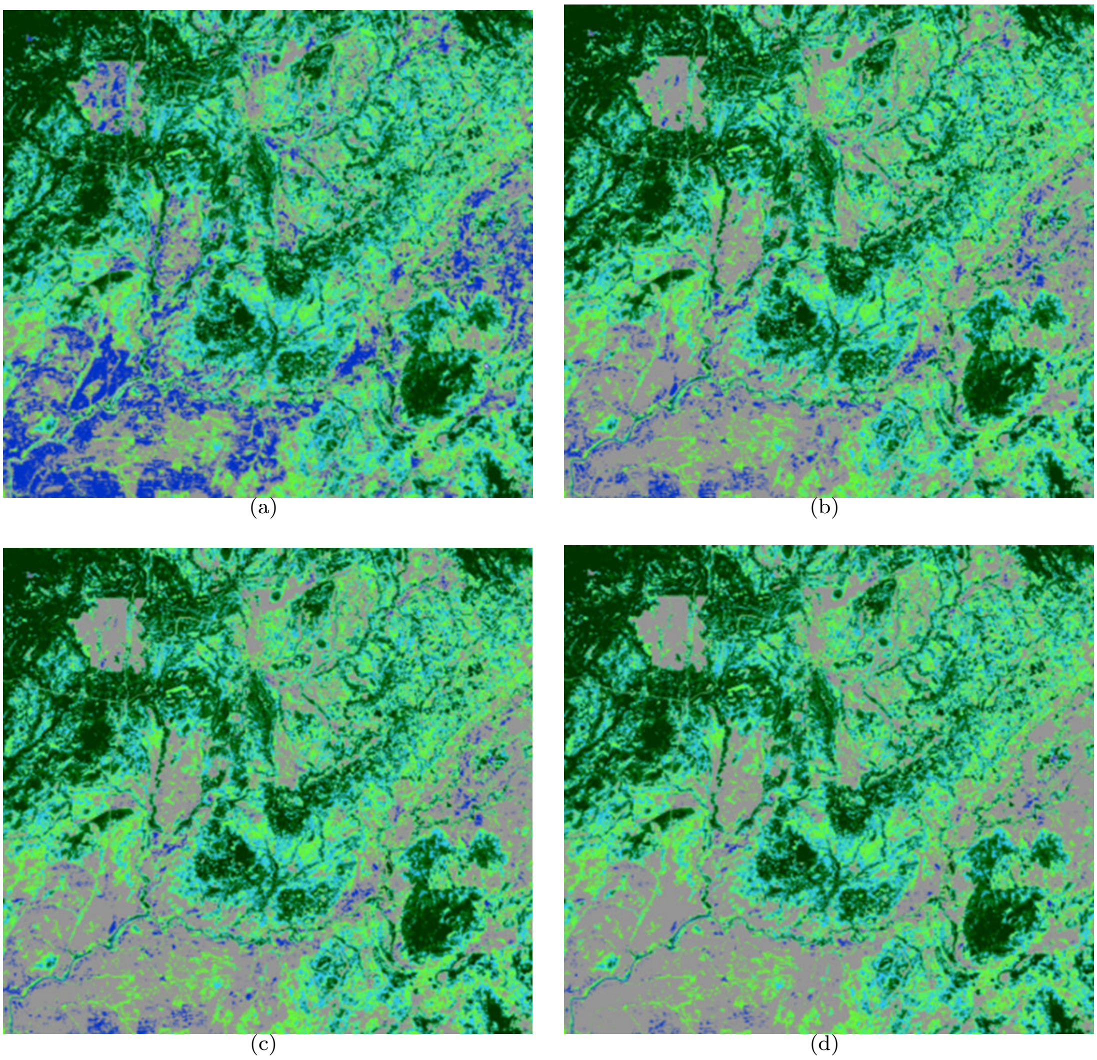
Class	Samples	FCM (True/%)	SFCM (True/%)	FCM-PSO (True/%)	SFCM-PSO (True/%)
Class 1	7982	7343/91.994%	7954/99.649%	7783/97.507%	7978/99.949%
Class 2	327	284/86.850%	313/95.719%	297/90.826%	320/97.247%
Class 3	298	253/84.899%	264/88.591%	276/92.617%	288/96.644%
Class 4	309	288/93.204%	293/94.822%	289/93.528%	295/95.469%
Class 5	412	347/84.223%	382/92.718%	378/91.748%	389/94.417%
Class 6	278	239/85.971%	260/93.525%	263/94.604%	271/97.482%
Sum	9606	8754/91.131%	9466/98.543%	9286/96.669%	9539/99.303%

results show that the proposed method (SFCM-PSO) produces the highest accuracy, especially with class 1 (Surface water) with an accuracy of 99.949% while the lowest accuracy of 94.417% is reached with class 5 (Perennial tree crops). This indicates confusion between planted forests, low woods; perennial tree crops; and dense vegetation. The average accuracy of the total number of pixels labeled is 99.303% for SFCM-PSO algorithm, 96.669% for FCM-PSO algorithm, 98.543% for SFCM algorithm and 91.131% for FCM algorithm.

Experiment 2

The second experiment is performed with images taken of the area of Chu Prong district, Gia Lai province (Central highlands of Vietnam, see Figure 3.5b) with three bands of SPOT image, so the number of particles is 19. RS data used in the classification is the SPOT-5 multispectral image. The number of samples labeled is 261; 129; 172; 82; 102 and 93 for class 1; 2; 3; 4; 5; 6 respectively. Test results on SFCM-PSO algorithm show that $m = 2.37864$, on FCM-PSO algorithm show that $m = 1.94764$ corresponding to the minimum value of the function F.

Fig. 3.7(a, b, c, d) displays the clustering images obtained when running each of the algorithms on the images of Chu Prong area. The results in Table 3.11 show that SFCM-PSO has better quality clustering than algorithms FCM, SFCM, FCM-PSO.



(a) (b) (c) (d)

Figure 3.7: Land-cover classification results of Chu Prong area

In table 3.11, FCM-PSO algorithm gives the best clustering result at CS-I index with value 0.476524; SFCM-PSO algorithm is 0.468753 while the other clusters show SFCM-PSO algorithm for better clustering.

Table 3.12 shows the correct classification rate on labeled pixels. The results show that the proposed method (SFCM-PSO) has the highest accuracy rate, especially with class 1 (Surface water) with the accuracy of 99.617%, while with class 5 (perennial tree crops), the lowest accuracy of 96.078% is reached. The average accuracy of the total number of pixels labeled is 98.093% for SFCM-PSO algorithm, 93.455% for FCM-PSO algorithm, 97.139% for SFCM algorithm and 84.148% for FCM algorithm.

Table 3.11: Validity indices obtained for Chu Prong area

Method	***XB-I***	***PC-I***	***CE-I***	***D-I***	***IQI***	***CS-I***	***MSE***
FCM	0.463852	0.338713	0.376192	0.089362	0.752082	0.187465	23.589621
SFCM	0.277341	0.397653	0.327897	0.168431	0.897409	0.366524	**11.824844**
FCM-PSO	0.221844	0.427562	0.297846	0.148914	0.762653	**0.476524**	15.974978
SFCM-PSO	**0.187651**	**0.538762**	**0.276122**	**0.187235**	**0.897431**	0.468753	**11.824844**

Table 3.12: Land-cover classification results by percentage of Chu Prong area

Class	Samples	FCM (True/%)	SFCM (True/%)	FCM-PSO (True/%)	SFCM-PSO (True/%)
Class 1	261	242/92.720%	259/99.234%	258/98.851%	260/99.617%
Class 2	129	110/85.271%	121/93.798%	117/90.697%	126/97.674%
Class 3	172	127/73.837%	167/97.093%	160/93.023%	170/98.837%
Class 4	82	72/87.805%	81/98.780%	77/93.902%	80/97.561%
Class 5	102	80/78.431%	97/95.098%	89/87.255%	98/96.078%
Class 6	93	74/79.570%	89/95.699%	83/89.247%	90/96.774%
Sum	839	706/84.148%	815/97.139%	787/93.445%	825/98.093%

Through two above experiments, based on the indicators XB-I, PC-I, CE-I, D-I, IQI, MSE and CS-I, in most cases, the proposed algorithm SFCM-PSO produce better results than other algorithms, namely: FCM-PSO, SFCM, and FCM. Furthermore, based on the labeled data, the results classified by SFCM-PSO algorithm for accuracy 99.608% at Hanoi area and 98.093% at Chu Pong area. This percentage is lower for SFCM,

FCM-PSO and FCM algorithms.

Fig. 3.8 shows the changes in the value of the function F by the number of iterations with images of two areas: Hanoi and Chu Prong.

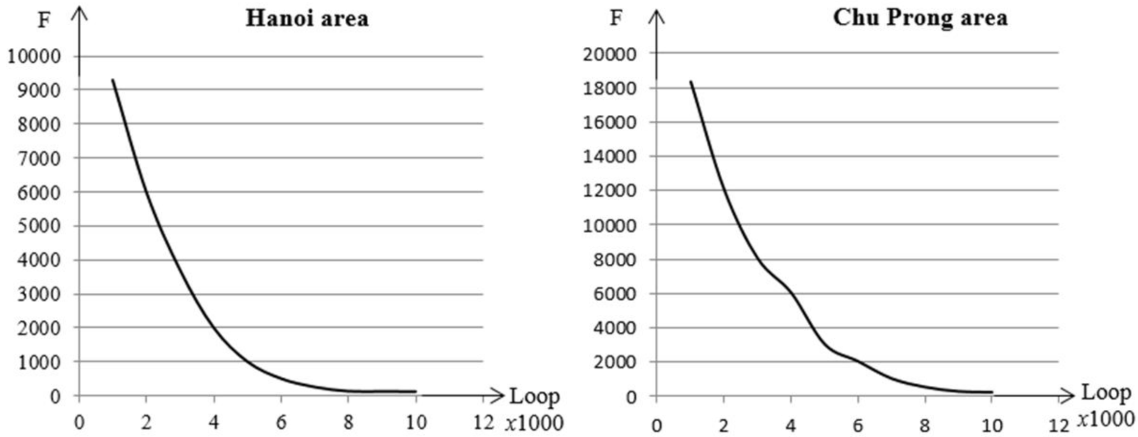


Figure 3.8: The values of the objective function F

Two experimental areas showed that the hybrid method of semi-supervised fuzzy clustering and particle swarm optimization method for RS imagery analysis (SFCM-PSO) achieve higher accuracy FCM-PSO, SFCM and FCM algorithms. The above results have suggested that using optimization techniques can improve the accuracy of semi-supervised clustering algorithms.

In the next section, the dissertation will present a hybrid method between interval type-2 semi-supervised PFCM and PSO technique.

3.4 Hybrid method of interval type-2 SPFCM and PSO

3.4.1 General Semi-supervised PFCM

The section proposes a general semi-supervised PFCM clustering (GSPFCM) algorithm to improve the clustering quality of PFCM. Our proposed method can solve problems in which labeled data is much less than unlabeled data.

In this part, we present a general semi-supervised algorithm based on PFCM algorithm. Consider data $X = \{\mathrm{x}_k, \mathrm{x}_k \in \mathrm{R}^d, \mathrm{k} = 1,..., \mathrm{n}\}$, with $\mathrm{X} = \mathrm{X}_1 \cup \mathrm{X}_2$, $X_1 = [x_1^*, x_2^*, ..., x_L^*]$ is a labeled data set and $X_2 = [x_{L+1}, x_{L+2}, ..., x_n]$ is an unlabeled data set ($|\mathrm{X}_1| << |\mathrm{X}_2|$).

From the labeled data set, the centroid constraints $V^* = [v_1^*, v_2^*, ..., v_c^*]$ will be calculated by averaging, c is the number of clusters.

The constraint of fuzzy MF $U^* = [\mu_{ik}^*]$ is calculated by equation:

$$\mu_{ik}^* = 1/\sum_{z=1}^{c} (\frac{x_k - v_i^*}{x_k - v_z^*})^{2/(m-1)} \tag{3.30}$$

In equation 1.12, the T value is a constant defined by the user, but in the study [41], Krishnapuram and Keller also suggest using fuzzy MF as a good way to initialize the parameter T according to the following formula:

$$\gamma_i = K\sum_{k=1}^{n} (\mu_{ik})^\eta d_{ik}^2 / \sum_{k=1}^{n} (\mu_{ik})^\eta \tag{3.31}$$

Where μ_{ik} is the fuzzy MF value from the results of the equation 3.30, K is a user-defined constant (usually set to 1). The constraint of possibilistic MF $T^{(*)} = [\tau_{ik}^{(*)}]$ is calculated by equation 3.31 and 1.12.

From 3 constraints on fuzzy MF $U^* = [\mu_{ik}^*]$, possibilistic MF $T^{(*)} = [\tau_{ik}^{(*)}]$ and cluster centroids $V^* = [v_1^*, v_2^*, ..., v_c^*]$, we propose a new objective function $J_{m,\eta}(U, T, V, X, \gamma)$ as follows:

$$\begin{aligned} J_{m,\eta} = \sum_{i=1}^{c}\sum_{k=1}^{n}(a\|\mu_{ik}-\mu_{ik}^*\|^m + b\|\tau_{ik}-\tau_{ik}^*\|^\eta)(\|v_i - x_k\|^2 \\ +\delta\|v_i - v_i^*\|^2) + \sum_{i=1}^{c}\gamma_i\sum_{k=1}^{n}(1-\tau_{ik})^\eta \end{aligned} \quad (3.32)$$

With the constraints:

$$0 \le \mu_{ik}, \tau_{ik} \le 1; \sum_{i=1}^{c}\mu_{ik} = 1; \sum_{k=1}^{n}\tau_{ik} = 1; 1 \le i \le c; 1 \le k \le n \quad (3.33)$$

Parameters a, b and δ are user-defined constants, representing the importance of constraints, $m, \eta > 1; a, b > 0; \delta \ge 0$. $\delta = 0$ when v_i^* does not exist.

Set $D_{ik}^2 = \|v_i - x_k\|^2 + \delta\|v_i - v_i^*\|^2$.

GSPFCM algorithm is expressed as follows: $X = \{\mathrm{x}_k, \mathrm{x}_k \in \mathrm{R}^d, \mathrm{k} = 1,..., \mathrm{n}\}$, X contains at least c distinct data points. With the constraint 3.33 then $J_{m,\eta}(U, T, V, X, \gamma)$ may minimize if only:

$$\mu_{ik} = \mu_{ik}^* + \frac{(1 - \sum_{i=1}^{c}\mu_{ik}^*)[1/D_{ik}^2]^{1/(m-1)}}{\sum_{i=1}^{c}[1/D_{ik}^2]^{1/(m-1)}} \quad (3.34)$$

$$\tau_{ik} = \begin{cases} \left(\tau_{ik}^* + \left[\frac{\gamma_i}{bD_{ik}^2}\right]^{\frac{1}{\eta-1}}\right) / \left(1 + \left[\frac{\gamma_i}{bD_{ik}^2}\right]^{\frac{1}{\eta-1}}\right) & \tau_{ik} \ge \tau_{ik}^* \\ \left(\tau_{ik}^* - \left[\frac{\gamma_i}{bD_{ik}^2}\right]^{\frac{1}{\eta-1}}\right) / \left(1 - \left[\frac{\gamma_i}{bD_{ik}^2}\right]^{\frac{1}{\eta-1}}\right) & else \end{cases} \quad (3.35)$$

$$v_i = \frac{\sum_{k=1}^{n}(a\|\mu_{ik}-\mu_{ik}^*\|^m + b\|\tau_{ik}-\tau_{ik}^*\|^\eta)(x_k + v_i^*)}{\sum_{k=1}^{n}(a\|\mu_{ik}-\mu_{ik}^*\|^m + b\|\tau_{ik}-\tau_{ik}^*\|^\eta)(1+\delta)} \quad (3.36)$$

Equation 3.34 can be achieved by using the Lagrange multiplier with fixed T and V by minimum problem:

$$\min\left\{\sum_{i=1}^{c}\sum_{k=1}^{n}(a\|\mu_{ik}-\mu_{ik}^*\|^m)(\|v_i - x_k\|^2 + \delta\|v_i - v_i^*\|^2)\right\} \quad (3.37)$$

It can be seen that equation 3.34 is independent of the constant a and depends only on v_i and v_i^*. When $\mu_{ik}^* = 0$ (μ_{ik}^* does not exist or not use), if the distance D_{ik} is considered similar to the distance d_{ik} then equation 3.34 is similar the to fuzzy membership in FCM algorithm.

Equation 3.35 is achieved by handling the minimum problem for the objective function 3.32, with V and U fixed by the minimum problem:

$$\min\left\{(a\|\mu_{ik} - \mu_{ik}^*\|^m + b\|\tau_{ik} - \tau_{ik}^*\|^\eta)D_{ik}^2 + \gamma_i(1-\tau_{ik})^\eta\right\} \tag{3.38}$$

When $\tau_{ik}^* = 0$ (τ_{ik}^* does not exist or is not used), if the distance D_{ik} is considered similar to the distance d_{ik} then equation 3.35 is similar to the possibilistic membership in PCM algorithm.

Similarly, equation 3.36 is achieved by minimizing the following objective function with fixed U and T:

$$\min\left\{\sum_{k=1}^{n}(\|\mu_{ik} - \mu_{ik}^*\|^m + \|\tau_{ik} - \tau_{ik}^*\|^\eta)D_{ik}^2\right\} \tag{3.39}$$

If v_i^* is not used or does not exist then $\delta = 0$. In equation 3.36, if additional information (v_i^*, τ_{ik}^*, μ_{ik}^*) is not used, they will become the equation 1.10 in PFCM.

Without reducing the generality, the additional information μ_{ik}^*, τ_{ik}^*, v_i^* can be achieved by different methods. May be from labeled data, experts' experience or results from other methods. The calculation of μ_{ik}^*, τ_{ik}^*, v_i^* in this study is only one of them.

Computational complexity: GSPFCM algorithm will execute a conditional loop, when either of the conditions $max(\left\|U^{(t+1)} - U^{(t)}\right\| + \left\|T^{(t+1)} - T^{(t)}\right\|) \leq \varepsilon$ or $t > T_{max}$ comes first, the algorithm will stop and give the classification result. Each loop will calculate V, U and T according to equations

Algorithm 3.4 General semi-supervised possibilistic fuzzy c-means algorithm (GSPFCM)

Input: A dataset $X = X_1 \cup X_2$, $X_1 = [x_1^*, x_2^*, ..., x_L^*]$, $X_2 = [x_{L+1}, x_{L+2}, ..., x_n]$ ($|X_1| << |X_2|$), the number of clusters c ($1 < c < n$), fuzzifiers $m, \eta > 1$, $T_{\max}, t = 0$, $a, b > 0; \delta \geq 0$.
Output: The membership matrix U, T and the centroid matrix V.
Step 1: Compute $V^{(*)} = [v_i^{(*)}], V^{(*)} \in R^{dxc}$ from X_1.
Step 2: Compute $U^{(*)} = [\mu_{ik}^{(*)}]$ by using Equation 3.30.
Step 3: Compute $T^{(*)} = [\tau_{ik}^{(*)}]$ by using Equation 3.31 and Equation 1.12.
Step 4: Initialize the centroid matrix $V^{(0)}$ and fuzzy MF $U^{(0)}$ by running FCM algorithm on dataset X.
Step 5: Compute $T^{(0)}$ by using Equation 3.31 and Equation 1.12.
Step 6: Loop
6.1 $t = t + 1$
6.2 Compute $V^{(t)} = [v_1^{(t)}, v_2^{(t)}, ..., v_c^{(t)}]$ by using Equation 3.36.
6.3 Compute $U^{(t)} = [\mu_{ik}^{(t)}]$ by using Equation 3.34.
6.4 Compute $T^{(t)} = [\tau_{ik}^{(t)}]$ by using Equation 3.31 and 3.35.
6.5 Check if $max(\left\|U^{(t+1)} - U^{(t)}\right\| + \left\|T^{(t+1)} - T^{(t)}\right\|) \leq \varepsilon$ or $t > T_{max}$ then stop else go to Step 6.1.
Defuzzification: Assign data x_k to the i^{th} cluster if $u_{ik} \geq u_{jk}, j = 1, ..., c; j \neq c$.

3.36, 3.34, 3.31 and 3.35. The algorithm stops at the T_{max} loop, the computational complexity of the algorithm will be $O(4ndcT_{max})$. When n is large, the computational complexity of GSPFCM and PFCM algorithm is the same.

3.4.2 General Interval type-2 Semi-supervised PFCM

In this section, the dissertation presents a hybrid algorithm of the semi-supervised PFCM clustering and PSO technique based on IT2FS. Typically, supervised methods require a large amount of labeled data for training; this method can be applied in cases where there is very little labeled data.

Dataset $X = \{\mathrm{x}_k, \mathrm{x}_k \in \mathrm{R}^d, \mathrm{k} = 1, ..., \mathrm{n}\}$ with $X = X_1 \cup X_2$, $X_1 = [\mathrm{x}_1^*, \mathrm{x}_2^*, ..., \mathrm{x}_L^*]$ is the labeled dataset and $X_2 = [\mathrm{x}_{L+1}, \mathrm{x}_{L+2}, ..., \mathrm{x}_n]$ is the unlabeled dataset ($|X_1| << |X_2|$). Let c be the number of clusters, calculation c centroids $v_1^*, v_2^*, ..., v_c^*$ from the labeled pixel dataset and $V^* = [v_1^*, v_2^*, ..., v_c^*]$ is the set of additional cluster centroids, which is averaged from the labeled data as follows:

$$v_i^* = \sum_{s=1}^{m_i} P_{\text{is}}/N_i \tag{3.40}$$

Where P_{is} is the s^{th} labeled pixel on the i^{th} cluster, N_i is the number of labeled pixels on the i^{th} cluster, $s = 1, 2, ..., N_i$; $i = 1, 2, ..., c$. The additional fuzzy MF is calculated based on a set of additional centroid V^* by FCM algorithm:

$$\mu_{ik}^* = 1/\sum_{z=1}^{c} (\frac{x_k - v_i^*}{x_k - v_z^*})^{2/(m-1)} \tag{3.41}$$

With γ_i is calculated according to equation 1.7. The additional possibilistic MF is calculated based on a set of additional centroid V^* by PCM algorithm:

$$\tau_{ik}^* = 1/\left(1 + (b\,\|v_i^* - x_k\|)^{1/(\eta-1)}/\gamma_i\right) \tag{3.42}$$

We propose a new objective function by adding additional information including the MFs μ_{ik}^*, τ_{ik}^* and cluster centroids v_i^*. A new objective function is as follows:

$$\min\left\{\begin{array}{l} J_{m,\eta}(U,T,V,X,\gamma) = \sum\limits_{i=1}^{c}\sum\limits_{k=1}^{n}(a\|\mu_{ik} - \mu_{ik}^*\|^m + b\|\tau_{ik} - \tau_{ik}^*\|^\eta)(\|v_i - x_k\|^2 + \\ \quad +\delta\|v_i - v_i^*\|^2) \;\; + \sum\limits_{i=1}^{c}\gamma_i\sum\limits_{k=1}^{n}(1-\tau_{ik})^\eta \end{array}\right\} \tag{3.43}$$

Subject to the constraints:

$$0 \le \mu_{ik}, \tau_{ik} \le 1; \sum_{i=1}^{c}\mu_{ik} = 1; \sum_{k=1}^{n}\tau_{ik} = 1; 1 \le i \le c; 1 \le k \le n; m, \eta > 1; a, b > 0 \tag{3.44}$$

Set

$$D_{ik}^2 = \|v_i - x_k\|^2 + \delta\|v_i - v_i^*\|^2 \tag{3.45}$$

Where $\delta \geq 0$ is a user-defined constant that represents the role of additional centroid value in the objective function and $\delta = 0$ when v_i^* does not exist. To minimize the objective function 3.43 we use the Lagrange operator we get U, T, V for the new GSPFCM algorithm, the implementation steps are similar to PFCM algorithm.

Expanding the equation 3.43 by using two fuzziness parameters m_1, m_2 and two possibilistic parameters η_1, η_2. These parameters are conducted to make FOU corresponding upper and lower values of fuzzy clustering and possibilistic clustering. The new algorithm is the general interval type-2 semi-supervised PFCM clustering (GIT2SPFCM).

The use of m_1, m_2 and η_1, η_2 gives different objective functions to be minimized as follows:

$$\begin{aligned} J_{m_1,\eta_1} &= \sum_{i=1}^{c}\sum_{k=1}^{n}(a\|\mu_{ik}-\mu_{ik}^*\|^{m_1}+b\|\tau_{ik}-\tau_{ik}^*\|^{\eta_1})D_{ik}^2+\sum_{i=1}^{c}\gamma_i\sum_{k=1}^{n}(1-\tau_{ik})^{\eta_1} \\ J_{m_2,\eta_2} &= \sum_{i=1}^{c}\sum_{k=1}^{n}(a\|\mu_{ik}-\mu_{ik}^*\|^{m_2}+b\|\tau_{ik}-\tau_{ik}^*\|^{\eta_2})D_{ik}^2+\sum_{i=1}^{c}\gamma_i\sum_{k=1}^{n}(1-\tau_{ik})^{\eta_2} \end{aligned} \tag{3.46}$$

Subject to the constraints:

$$\begin{aligned} & m_1, \eta_1, m_2, \eta_2 > 1; a, b > 0; \delta \geq 0; 0 \leq \mu_{ik}, \tau_{ik} \leq 1; \\ & \sum_{i=1}^{c}\mu_{ik} = 1; \sum_{k=1}^{n}\tau_{ik} = 1; 1 \leq i \leq c; 1 \leq k \leq n \end{aligned} \tag{3.47}$$

Theorem 3.1. *For* $X = \{\mathrm{x}_k, \mathrm{x}_k \in \mathrm{R}^M, \mathrm{k} = 1,...,\mathrm{n}\}$*,* $m, \eta > 1; c \geq 1,\ \delta \geq 0$ *and* X *contains at least* c *distinct data points. With the constraints 3.47 and the equation 3.46 then* $J_{m_1,\eta_1}(U, T, V, X, \gamma)$ *and* $J_{m_2,\eta_2}(U, T, V, X, \gamma)$

may minimize if only:

$$\mu_{ik}^{(1)} = \begin{cases} \mu_{ik}^{*} + \frac{(1-\sum_{i=1}^{c}\mu_{ik}^{*})\left[1/D_{ik}^{2}\right]^{1/(m_1-1)}}{\sum_{i=1}^{c}\left[1/D_{ik}^{2}\right]^{1/(m_1-1)}} & if \frac{1}{\sum_{j=1}^{C}(D_{ik}/D_{jk})} < \frac{1}{c} \\ \mu_{ik}^{*} + \frac{(1-\sum_{i=1}^{c}\mu_{ik}^{*})\left[1/D_{ik}^{2}\right]^{1/(m_2-1)}}{\sum_{i=1}^{c}\left[1/D_{ik}^{2}\right]^{1/(m_2-1)}} & otherwise \end{cases} \tag{3.48}$$

$$\mu_{ik}^{(2)} = \begin{cases} \mu_{ik}^{*} + \frac{(1-\sum_{i=1}^{c}\mu_{ik}^{*})\left[1/D_{ik}^{2}\right]^{1/(m_1-1)}}{\sum_{i=1}^{c}\left[1/D_{ik}^{2}\right]^{1/(m_1-1)}} & if \frac{1}{\sum_{j=1}^{C}(D_{ik}/D_{jk})} \geq \frac{1}{c} \\ \mu_{ik}^{*} + \frac{(1-\sum_{i=1}^{c}\mu_{ik}^{*})\left[1/D_{ik}^{2}\right]^{1/(m_2-1)}}{\sum_{i=1}^{c}\left[1/D_{ik}^{2}\right]^{1/(m_2-1)}} & otherwise \end{cases} \tag{3.49}$$

Where

$$\begin{aligned} \bar{\mu}_i(x_k) &= max\{\mu_{ik}^{(1)},\mu_{ik}^{(2)}\} \\ \underline{\mu}_i(x_k) &= \min\{\mu_{ik}^{(1)},\mu_{ik}^{(2)}\} \end{aligned} \tag{3.50}$$

$$\tau_{ik}^{(1)} = \begin{cases} \left(\tau_{ik}^{*} + [\gamma_i/bD_{ik}^{2}]^{1/(\eta_1-1)}\right)/\left(1 + [\gamma_i/bD_{ik}^{2}]^{1/(\eta_1-1)}\right) & \tau_{ik} \geq \tau_{ik}^{*} \\ \left(\tau_{ik}^{*} - [\gamma_i/bD_{ik}^{2}]^{1/(\eta_1-1)}\right)/\left(1 - [\gamma_i/bD_{ik}^{2}]^{1/(\eta_1-1)}\right) & else \end{cases} \tag{3.51}$$

$$\tau_{ik}^{(2)} = \begin{cases} \left(\tau_{ik}^{*} + [\gamma_i/bD_{ik}^{2}]^{1/(\eta_2-1)}\right)/\left(1 + [\gamma_i/bD_{ik}^{2}]^{1/(\eta_2-1)}\right) & \tau_{ik} \geq \tau_{ik}^{*} \\ \left(\tau_{ik}^{*} - [\gamma_i/bD_{ik}^{2}]^{1/(\eta_2-1)}\right)/\left(1 - [\gamma_i/bD_{ik}^{2}]^{1/(\eta_2-1)}\right) & else \end{cases} \tag{3.52}$$

Where

$$\begin{aligned} \bar{\tau}_i(x_k) &= max\{\tau_{ik}^{(1)},\tau_{ik}^{(2)}\} \\ \underline{\tau}_i(x_k) &= \min\{\tau_{ik}^{(1)},\tau_{ik}^{(2)}\} \end{aligned} \tag{3.53}$$

Because each pattern has a membership interval as the upper $\bar{\mu}, \bar{\tau}$ and the lower $\underline{\mu}, \underline{\tau}$, each cluster centroid is represented by the interval between v^L and v^R.

For possibilistic membership grades:

$$\tau_i(x_k) = (\bar{\tau}_i(x_k) + \underline{\tau}_i(x_k))/2; i = 1, ..., c; k = 1, ..., n \tag{3.54}$$

Proof. The equations 3.48 and 3.49 are similar to the equations 1.17, 1.18 in IT2FCM algorithm achieved by using the Lagrange multiplier and providing additional information. When $\mu_{ik}^* = 0$ (μ_{ik}^* does not exist or is not used) and the distance D_{ik} is considered similar to the distance d_{ik} then the equations 3.48 and 3.49 become the equations 1.17, 1.18 in IT2FCM algorithm.

Equations 3.51 and 3.52 are achieved by handling the minimum problem for the objective function 3.46, with V and U fixed by minimum problem:

$$min\{J_{m_1,\eta_1}(T) = (a\|\mu_{ik} - \mu_{ik}^*\|^{m_1} + b\|\tau_{ik} - \tau_{ik}^*\|^{\eta_1})D_{ik}^2 + \gamma_i(1-\tau_{ik})^{\eta_1}\} \tag{3.55}$$

and

$$min\{J_{m_2,\eta_2}(T) = (a\|\mu_{ik} - \mu_{ik}^*\|^{m_2} + b\|\tau_{ik} - \tau_{ik}^*\|^{\eta_2})D_{ik}^2 + \gamma_i(1-\tau_{ik})^{\eta_2}\} \tag{3.56}$$

When $\tau_{ik}^* = 0$ (τ_{ik}^* does not exist or is not used), the distance D_{ik} is considered similar to the distance d_{ik} then the equation 1.26 becomes the equation 1.6 in PCM algorithm or the equation 1.11 in PFCM algorithm, but with two possibilistic parameters η_1, η_2.

Because each pattern has both membership interval values as the upper $\bar{\mu}, \bar{\tau}$ and the lower $\underline{\mu}, \underline{\tau}$; each cluster centroid is represented by an interval between v^L and v^R. Using Algorithm 1.1 and 1.2 to find the centroids v^L and v^R, where $V = [\mathrm{v}_i]$ are computed in the same way of IT2FCM in equation 1.20.

In Equations 3.48, 3.49, 3.51, 3.52 and 3.46, we can use both the MFs μ_{ik}^*, τ_{ik}^* and the centroid v_i^*, or use one of them depending on the additional information that has been obtained. It is apparent that, when $\mu^*_{ik}, \tau^*_{ik}, v^*_i$ are not used ($\mu^*_{ik} = 0, \tau^*_{ik} = 0, v^*_i = 0$), GIT2SPFCM algorithm becomes IT2PFCM algorithm (PFCM based on IT2FS). □

Theorem GIT2SPFCM holds exactly as it does in IT2FCM algorithm. The implementation steps of GIT2SPFCM algorithm are similar to IT2FCM, details of the steps are as follows:

Algorithm 3.5 General interval type-2 semi-supervised possibilistic fuzzy c-means algorithm (GIT2SPFCM)

Input: Dataset $X = \{\mathrm{x}_k, \mathrm{x}_k \in \mathrm{R}^d,\ \mathrm{k} = 1,...,\mathrm{n}\}$, the labeled data set $X^* = \{\mathrm{P_{is}}, \mathrm{P_{is}} \in \mathrm{R}^d; \mathrm{s} << \mathrm{n}; \mathrm{i} = 1,...,\mathrm{c}\}$, the number of clusters $c(1 < c < n)$, fuzzifier parameters $m_1, m_2, m, \eta_1, \eta_2, \eta$, and T_{max}, $t = 0$.
Output: The membership matrix U, T and the centroid matrix V.
Step 1: Compute the additional cluster centroids $V^* = [\mathrm{v}^*_i]$ by using Equation 3.40, the additional fuzzy MF $U^* = [\mu^*_{ik}]$ by using Equation 3.41, the additional possibilistic MF $T^* = [\tau^*_{ik}]$ by using Equations 1.7, 3.42.
Step 2: Initialize the centroid matrix $V^{(t)} = [v^{(t)}_i], V^{(t)} \in R^{dxc}$ by choosing randomly from the input dataset X.
Step 3: Compute $U^{(t)}$ by using Equations 3.48, 3.49, 3.50, 1.21, 1.22, 1.23.
Step 4: Compute $T^{(t)}$ by using Equations 1.7, 3.51, 3.52, 3.53, 3.54.
Step 5: Repeat
5.1 $t = t + 1$
5.2 Compute the centroids v^R and v^L use Equation 3.46 and the algorithm 1.1 and 1.2.
5.3 Update the centroid matrix $V^{(t)} = [v^{(t)}_1, v^{(t)}_2, ..., v^{(t)}_C]$ by using Equation 1.20.
5.4 Update $U^{(t)}$ by using Equations 3.48, 3.49, 3.50, 1.21, 1.22, 1.23.
5.5 Update $T^{(t)}$ by using Equations 1.7, 3.51, 3.52, 3.53, 3.54.
5.6 Assign data x_k to the i^{th} cluster if $u_{ik} \geq u_{jk}, j = 1, ..., c; j \neq c$.
5.7 Check if $max(\left\|U^{(t+1)} - U^{(t)}\right\| + \left\|T^{(t+1)} - T^{(t)}\right\|) \leq \varepsilon$. If yes then stop and go to Output else go to Step 5.1.

Defuzzification: Assign the data pattern x_k to the i^{th} cluster if $u_{ik} \geq u_{jk}, j = 1, ..., c; j \neq c$.

Similar to algorithms PFCM, GT2SPFCM will execute a conditional loop, when either of the conditions $t \geq T_{\max}$ or $max(\left\|U^{(t+1)} - U^{(t)}\right\| + \left\|T^{(t+1)} - T^{(t)}\right\|) \leq \varepsilon$ comes first, the algorithm will stop and give the classification result.

Computational complexity: The computational complexity of the proposed algorithm is mainly in Step 5.2, calculate the right and left centroid according to the algorithms 1.1 and 1.2, this algorithm performs the process of sorting n patterns on each of d features in ascending order: $x_1 \leq x_2 \leq ... \leq x_n$ then execute d loop to find v^R and v^L. Using the quick-sort algorithm, the computational complexity of algorithms 1.1 and 1.2 are $O(dcnlogn + dcn)$. In each loop there is the computational complexity $O(dcnlogn)$.

When n is large, and the number of iterations is T_{max}, the computational complexity of GIT2SPFCM algorithm is $O(dcnlognT_{max})$. The computational complexity of GIT2SPFCM algorithm is equivalent to that of IT2FCM algorithm.

3.4.3 Hybrid method of GIT2SPFCM and PSO

In real life applications, users usually have difficulty initializing the parameters when using the above-proposed algorithms. These parameters are usually not fixed but adjusted according to the characteristics of each data set. This means that the parameters may be suitable for one data set, but unsuitable for another. In this section, we propose a method to find parameters using PSO technique [38]. This algorithm has the advantage of simple installation and fast convergence, which is suitable for large data sets.

As a starting point, it is important point to note that PSO algorithm is the initialization of particles. Typically, in satellite image classification, the number of clusters is determined by the user based on the number of land covers on the image. The parameters of GSPFCM algorithm need to be set at optimal values including centroid of clusters and the parameters $m, m_1, m_2, \eta, \eta_1, \eta_2, a, b$. With satellite image data has M

spectrum bands ($d = 3$ with RGB color image), the number of clusters is c. Therefore, the total number of particles initialized is $d * c + 8$ (see 3.57).

$$\underbrace{v_{11}, v_{12}, ...v_{1d}}_{V_1} \ \underbrace{v_{21}, v_{22}, ...v_{2d}}_{V_2} \ \cdots \ \underbrace{v_{c1}, v_{c2}, ...v_{cd}}_{V_c} \ \underbrace{m, m_1, m_2, \eta, \eta_1, \eta_2, a, b}_{\text{parameters}} \tag{3.57}$$

Where $v_i = [\mathrm{v_{ij}}]$ is cluster centroids ($i = 1, ..., c; j = 1, ..., d$) and $m, m_1, m_2, \eta, \eta_1, \eta_2$ are fuzzy and possibilistic parameters, and a, b are user-defined parameters.

Let $P = (p_1, p_2, ..., p_{c*d}, p_{c*d+1}, ..., p_{c*d+8})$ be the set of all particles position, $p_1, p_2, ..., p_{c*d}$ represents cluster centroids, $p_{c*d+1}, p_{c*d+2}, p_{c*d+3}$ represent fuzzy parameters,$p_{c*d+4}, p_{c*d+5}, p_{c*d+6}$ represent posibilistic parameters, p_{c*d+7}, p_{c*d+8} represent a, b parameters. With each particle, there will be position and movement velocity. The position of a particle is usually randomly generated in the search space. Each particle will include the following information: p_i is the current position of i^{th} particle; vel_i is the current velocity of i^{th} particle; $pBest_i$ is the personal best position of i^{th} particle.

With the objective function F, the personal best position of a particle at the time t is updated as:

$$pBest_i^{(t+1)} = \begin{cases} pBest_i^{(t)} & \text{if} \quad F(p_i^{(t+1)}) \geq F(pBest_i^{(t)}) \\ p_i^{(t+1)} & \text{if} \quad F(p_i^{(t+1)}) < F(pBest_i^{(t)}) \end{cases} \tag{3.58}$$

With the entire population, the best position of the population is denoted by g_{Best}:

$$\begin{aligned} &gBest^{(t)} = \{pBest_i^{(t)} | F(pBest_i^{(t)}) \\ &= \min\{F(pBest_1^{(t)}), F(pBest_2^{(t)}), ..., F(pBest_{c*d+4}^{(t)})\}\} \end{aligned} \tag{3.59}$$

For each iteration of PSO algorithm, p_i and vel_i are updated as follows:

$$vel_i^{(t+1)} = \omega * vel_i^{(t)} + c_1 * r_1 * (pBest_i^{(t)} - p_i^{(t)}) + c_2 * r_2 * (gBest^{(t)} - p_i^{(t)})$$
$$p_i^{(t+1)} = p_i^{(t)} + vel_i^{(t+1)} \tag{3.60}$$

An important issue in PSO algorithm is the selection of parameters. Parameters c_1 and c_2 represent the influence of the best particle solution and the best global solution. These two parameters are normally set to 2.05 as suggested in the original paper [38]. Parameter ω is the inertia parameter, which indicates the rate of change in velocity of the particle during moving, common values range from zero to one. And r_1, r_2 are the random number in the range $(0, 1)$. In PSO algorithm loops, every particle must always be in the search space under the conditions $p_{\min} \le p_i \le p_{\max}, i = 1, ..., c * d$, $p_{c*d+1}, ..., p_{c*d+6} > 1$ and $p_{c*d+7}, p_{c*d+8} > 0$. If $p_i < p_{\min} \;\; then \;\; p_i = p_{\min}$, if $p_i > p_{\max} \;\; then \;\; p_i = p_{\max}$ $(i = 1, ..., c * d + 8)$. Same for the velocity of particles, set $vel_{\min} \le vel_i \le vel_{\max}, \forall i$ be the velocity limits of the particles, in which $vel_{\min}, vel_{max}$ are selected according to the user's experiences. A constraint is given, if $vel_i < vel_{\min} \;\; then \;\; vel_i = vel_{\min}$ and if $vel_i > vel_{\max} \;\; then \;\; vel_i = vel_{\max}$.

Similar to GIT2SPFCM algorithm, it is necessary to define an objective function for GIT2PFCM-PSO algorithm. The hybrid algorithm between GIT2SPFCM and PSO is considered to be minimum the objective function following:

$$F_{m_1,\eta_1,m_2,\eta_2}(U,T,V,X,\gamma) = \frac{F_{m_1,\eta_1}(U,T,V,X,\gamma) + F_{m_2,\eta_2}(U,T,V,X,\gamma)}{\min\limits_{i,j=1,\ldots,c;i\neq j} \left\| v_i - v_j \right\|^2} \tag{3.61}$$

Steps to implement hybrid algorithm between GIT2SPFCM and PSO are as follows:

Algorithm 3.6 The hybrid algorithm between GIT2SPFCM and PSO (GIT2SPFCM-PSO)

Input: Dataset $X = \{\mathrm{x}_k, \mathrm{x}_k \in \mathrm{R}^d, \mathrm{k} = 1, ..., \mathrm{n}\}$, the labeled data set $X^* = \{\mathrm{P_{is}}, \mathrm{P_{is}} \in \mathrm{R}^d; \mathrm{s} << \mathrm{n}; \mathrm{i} = 1, ..., \mathrm{c}\}$, the number of clusters $c(1 < c < n)$, fuzzifier parameters ϵ, and T_{max}, $t = 0$, $c_1, c_2, r_1, r_2, \omega$.
Output: The membership matrix U, T and the centroid matrix V.
Step 1: Compute the additional cluster centroids $V^* = [\mathrm{v}_i^*]$ by using Equation 3.40, the additional fuzzy MF $U^* = [\mu_{ik}^*]$ by using Equation 3.41, the additional possibilistic MF $T^* = [\tau_{ik}^*]$ by using Equations 1.7, 3.42.
Step 2: Initialization
2.1 Initialize the centroid matrix $V^{(0)} = [v_i^{(0)}], V^{(0)} \in R^{dxc}$ by using FCM algorithm.
2.2 Initialize the location particles $P^{(0)} = (p_1^{(0)}, p_2^{(0)}, ..., p_{c*d}^{(0)}, p_{c*d+1}^{(0)}, ..., p_{c*d+8}^{(0)})$ by using $V^{(0)} = [v_i^{(0)}]$ and the random values $m, m_1, m_2, \eta, \eta_1, \eta_2, a, b$ within limits of the search space.
2.3 Create the random velocity of particles: $vel_1^{(0)}, vel_2^{(0)}, ..., vel_{c*d}^{(0)}, vel_{c*d+1}^{(0)}, ..., vel_{c*d+8}^{(0)}$ within limits of the velocity.
2.4 Compute $U^{(0)}$ by using Equations 3.48, 3.49, 3.50, 1.21, 1.22, 1.23.
2.5 Compute $T^{(0)}$ by using Equations 1.7, 3.51, 3.52, 3.53, 3.54.
2.6 Compute $F_{m_1,\eta_1,m_2,\eta_2}^{(0)}$ by using Equation 3.61.
2.7 Let $pBest_i^{(0)} = p_i^{(0)}$, $gBest^{(0)}$ by using Equation 3.59.
Step 3: Hybrid algorithm of GIT2SPFCM and PSO
3.1 $t = t + 1$
3.2 For each particle i.
+ Compute the velocity of particles: $vel_i^{(t+1)} = \omega * vel_i^{(t)} + c_1 * r_1 * (pBest_i^{(t)} - p_i^{(t)}) + c_2 * r_2 * (gBest^{(t)} - p_i^{(t)})$
+ Compute the location of particles $p_i^{(t+1)} = p_i^{(t)} + vel_i^{(t+1)}$.
+ Compute the objective function $F_{m_1,\eta_1,m_2,\eta_2}^{(t)}$ by Equation 3.61.
+ Update $pBest_i^{(t)}$ by using Equation 3.58.
+ Update the cluster centroids $V^{(t)} = [v_i^{(t)}]$ and the parameters $m, m_1, m_2, \eta, \eta_1, \eta_2, a, b$ (if change).
3.3 Find the global best solution $gBest^{(t)}$ by using Equation 3.59.
3.4 Update $U^{(t)}$ by using Equations 3.48, 3.49, 3.50, 1.21, 1.22, 1.23.
3.5 Update $T^{(t)}$ by using Equations 1.7, 3.51, 3.52, 3.53, 3.54.
3.6 Check if $t > T_{\max}$ then go to Output else go to step 3.1.
Output: $V^{(t)}$, $U^{(t)}$, $T^{(t)}$, $m, m_1, m_2, \eta, \eta_1, \eta_2, a, b$.
Defuzzification: Assign data x_k to the i^{th} cluster if $u_{ik} \geq u_{jk}, j = 1, ..., C; j \neq C$.

Computational complexity: The computational complexity of the proposed algorithm is mainly in step 3. Each iteration will include: step 3.2, the computational complexity will be calculated by the equation 3.61 $O(ndc(dc + 8))$. Step 3.3 is $O(dc + 8)$, step 3.4 and step 3.5 the computational complexity by the equations 3.48, 3.49, 3.50, 1.21, 1.22, 1.23, 1.7, 3.51, 3.52, 3.53, 3.54 is $O(11ndc)$. So, in each loop in step 3 there is the computational complexity $O(nd^2c^2)$.

When n is large and the number of iterations of is T_{max}, the computational complexity of GIT2SPFCM-PSO algorithm is $O(nd^2c^2T_{max})$. The computational complexity analysis of algorithms PFCM, IT2FCM, GIT2SPFCM, and GIT2SPFCM-PSO shows that the computational complexity of the IT2FCM and GIT2SPFCM is similar and is the largest compared to other algorithms. When n is large, it can be stated that the computational complexity of algorithms PFCM and GIT2SPFCM-PSO is similar and is smaller than the computational complexity of algorithms IT2FCM and GIT2SPFCM.

3.4.4 Experiments

A. Initialize parameters and evaluation methods

We selected some datasets at different locations including (city, delta and mountain forest) for testing. Multi-spectral satellite images have been used, including Landsat-5 TM, Landsat-7 ETM+, Landsat-8, and Sentinel-2A. The image data is clustered to six classes as follows: Class 1: Surface water ; Class 2: Bare land ; Class 3: Grass, shrubs ; Class 4: Planted forests, low woods ; Class 5: Perennial tree crops ; Class 6: Dense vegetation (see Algorithm 1.6).

For a multi-spectral image with d bands, each pixel will be characterized by d components on d gray bands which are described as follows $X = [\mathrm{x}_1,\mathrm{x}_2,\ldots\mathrm{x}_n]$ with $x_i = (b_{i1}, b_{i2}, ..., b_{id})$.

Experimental algorithms include SFCM [51], GSPFCM, SPFCM-W, SPFCM-SS, SMKFCM, SIIT2FCM [68], SFCM-PSO, GIT2SPFCM, and GIT2SPFCM-PSO. Additional information in semi-supervised algorithms is calculated from the labeled data. The algorithms are executed for a maximum 1000 iterations, $\varepsilon = 10^{-6}$. For all algorithms, we first run FCM algorithm ($m = 2$) to determine the initial centroids. With the algorithms GSPFCM, SPFCM-W, SPFCM-SS, GIT2SPFCM and GIT2SPFCM-PSO, select $K = 1$ to calculate the value γ_i by using the equation 1.7.

The parameters of algorithms SFCM, SPFCM-W, SPFCM-SS, GSPFCM, SMKFCM and SIIT2FCM are selected according to the original papers.

The parameters of GIT2SPFCM algorithm are selected as follows: $m = \eta = 2$, $m_1 = \eta_1 = 1.5$, $m_2 = \eta_2 = 3.5$, $a = b = \delta = 1$.

The parameters of GIT2SPFCM-PSO algorithm are selected as follows: $m = 2$, $(p_{\min}; p_{\max}) = (0; 255)$ for 8 bits gray image or $(p_{\min}; p_{\max}) = (0; 65536)$ for 16 bits gray image with particles generated from cluster centroids information, $(p_{\min}; p_{\max}) = (1; 5)$ for other parameters. $(vel_{\min}; vel_{\max}) = (-\frac{p_{\max}-p_{\min}}{2}; \frac{p_{\max}-p_{\min}}{2})$ for cluster centroids and $(vel_{\min}; vel_{\max}) = (-2.5; 2.5)$ for other parameters. The parameters $c_1 = c_1 = 2.05$ are suggested in the original paper [38]. Let $\omega = 0.9$ and decrease to 0.1 when the maximum number of loops (the generation number) is reached.

The classification results are compared with the statistical data of the Vietnam National RS Center (VNRSC). The labeled data is taken directly from the satellite image data according to 6 land-cover classes. Besides, the clustering results have been evaluated by some validity indexes, including PC-I, D-I, CE-I, XB-I, $\tau - I$, CS-I, IQI, and MSE, which are introduced to assess the degree of characteristic separation between pixels and cluster centroids. Semi-supervised algorithms use the distance D_{ik}, otherwise the distance d_{ik} is used.

B. Land cover classification

Experiment 1: Landsat-7 ETM+ image

Experimental data include images of central area of Hanoi capital, Vietnam on $30 September 2012$ ($105^0 26' 47.1714" E$, $21^0 10' 15.7519" N$, and $106^0 12' 38.5927" E$, $20^0 52' 33.2849" N$) with 7 spectral bands and spatial resolution of $30m$ (see Figure 3.9).

Figure 3.9: RGB color image: Hanoi capital central area

Table 3.13: Parameters achieved when implementing GIT2SPFCM-PSO algorithm for Hanoi central area

Algorithm	m	m_1	m_2	η	η_1	η_2	a	b
GIT2SPFCM-PSO	2.21364	1.36534	3.26513	2.19874	1.47635	3.07366	0.52752	0.52463

Figure 3.10 shows the classification results according to six land-covers by algorithms a) SFCM; b) SFCM-PSO; c) SPFCM-W; d) SPFCM-SS; e) GSPFCM; f) SMKFCM; g) SIIT2FCM; h) GIT2SPFCM; i) GIT2SPFCM-PSO. In table 3.13, the parameter values are achieved by the GIT2SPFCM-

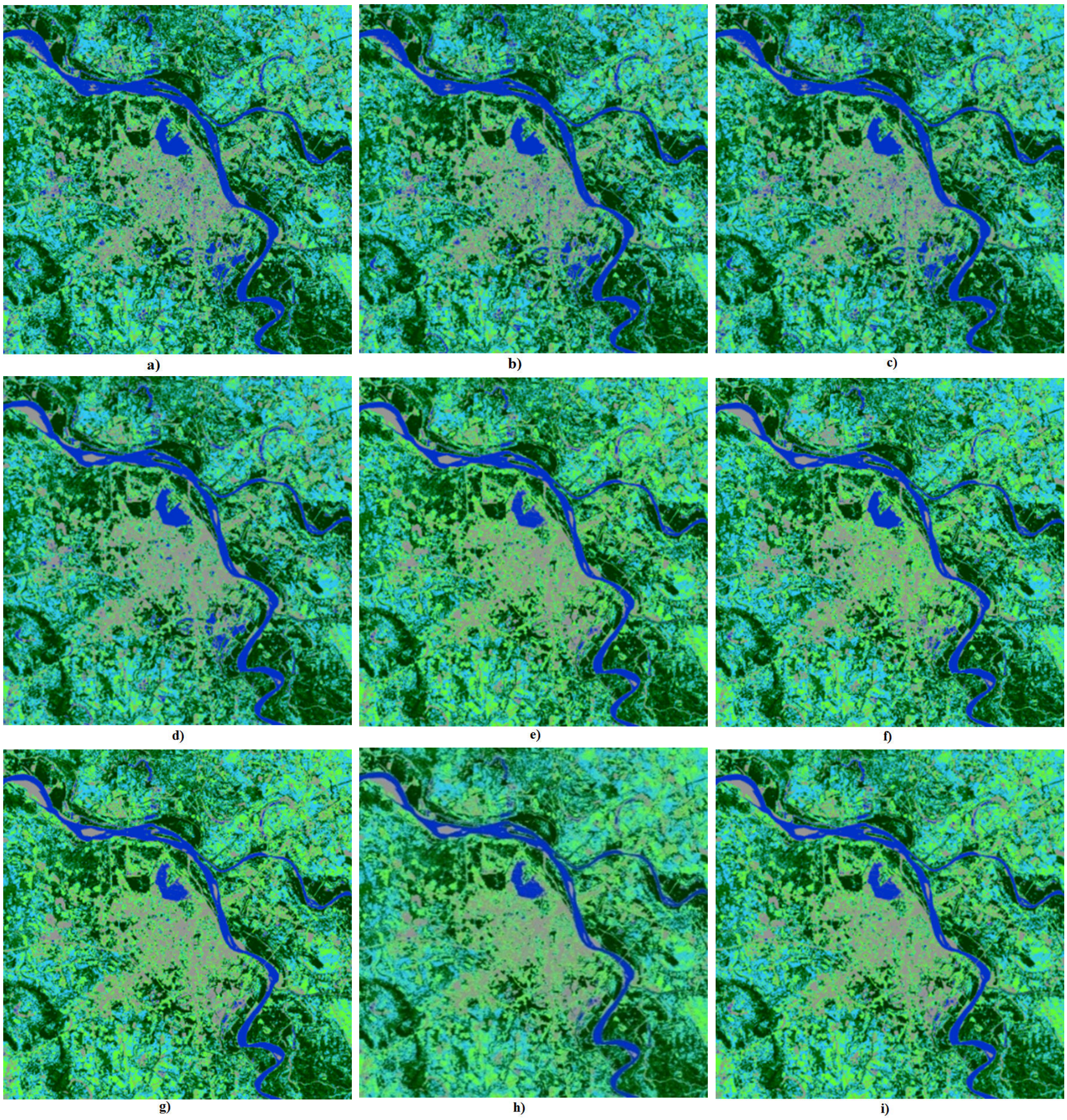


Figure 3.10: Land cover classification results of Hanoi central area: a) SFCM; b) SFCM-PSO; c) SPFCM-W; d) SPFCM-SS; e) GSPFCM; f) SMKFCM; g) SIIT2FCM; h) GIT2SPFCM; i) GIT2SPFCM-PSO

PSO algorithm implementation.

Table 3.14 shows the correct classification rate (%) according to each land cover based on the number of labeled pixels. GIT2SPFCM-PSO algorithm gives the best classification results with the correct classification rate reaching over 99% for labeled pixels, while the lowest rate is 87.77%

Table 3.14: Correct classification rate for Hanoi central area by labeled data (%)

Algorithm	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Class 6*	*Total*
SFCM	93.21	91.23	88.65	87.37	91.25	86.28	89.67
SFCM-PSO	95.16	93.67	92.46	89.89	92.23	94.32	92.96
SPFCM-W	95.34	93.76	93.22	91.26	92.65	91.89	93.02
SPFCM-SS	94.46	94.10	93.71	91.76	93.13	92.46	93.27
GSPFCM	99.23	96.36	98.32	96.32	94.33	95.48	96.67
SMKFCM	98.45	97.18	93.23	95.55	95.99	92.65	95.51
SIIT2FCM	98.43	97.09	95.88	93.23	96.28	94.33	95.87
GIT2SPFCM	99.11	98.33	98.37	97.42	98.48	97.56	98.21
GIT2SPFCM-PSO	100	98.11	99.15	99.08	98.43	99.32	**99.02**

by PFCM algorithm. This correct classification rate significantly increases when using the semi-supervised method. GIT2SPFCM algorithm has the correct classification rate of 98.21%, lower than GIT2SPFCM-PSO algorithm, while the computational complexity is higher than that of GIT2SPFCM-PSO algorithm.

Table 3.15: Land-cover classification results and VNRSC data (km^2) for Hanoi central area

Algorithm	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Class 6*
VNRSC	43.84	124.68	174.70	250.57	219.96	129.97
SFCM	57.27	133.06	131.19	252.31	217.08	152.81
SFCM=PSO	42.36	124.18	196.22	268.52	202.27	110.18
SPFCM-W	42.18	125.80	194.24	268.34	202.45	110.63
SPFCM-SS	42.72	124.90	186.14	278.20	227.27	102.50
GSPFCM	42.27	125.98	193.61	251.47	219.59	110.09
SMKFCM	51.30	128.40	163.70	230.43	217.35	152.54
SIIT2FCM	47.26	121.26	174.70	230.61	228.53	141.36
GIT2SPFCM	42.72	125.80	194.24	231.04	220.05	129.88
GIT2SPFCM-PSO	44.99	123.53	174.61	250.48	208.66	141.45

Table 3.15 and Table 3.16 show the area of six land-covers and VNRSC statistics data. It shows that GIT2SPFCM-PSO algorithm gives the classification results with the lowest deviation of 2.676% compared to VNRSC statistics data, or in other words, the classification accuracy produced by GIT2SPFCM-PSO is 97.324%. While deviation is 4.401% with GIT2SPFCM algorithm and 4.954% with SIIT2FCM algorithm,

the largest value comes from SFCM algorithm with 9.831%. This is an unsupervised algorithm in algorithms used for testing.

Table 3.16: Land-cover classification results and VNRSC data (%) for Hanoi central area

Algorithm	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Class 6*	*Differrence*
VNRSC	4.646	13.211	18.512	26.552	23.308	13.772	0.000
SFCM	6.068	14.100	13.901	26.736	23.003	16.192	9.831
SFCM-PSO	4.488	13.159	20.792	28.454	21.433	11.675	8.364
SPFCM-W	4.488	13.159	20.792	28.454	21.433	11.675	8.364
SPFCM-SS	4.526	13.235	19.724	29.479	24.082	10.861	7.968
GSPFCM	4.479	13.350	20.515	26.646	23.269	11.665	4.549
SMKFCM	5.436	13.606	17.346	24.418	23.031	16.163	7.153
SIIT2FCM	5.008	12.849	18.512	24.437	24.216	14.979	4.954
GIT2SPFCM	4.526	13.331	20.582	24.481	23.318	13.762	4.401
GIT2SPFCM-PSO	4.767	13.089	18.502	26.542	22.111	14.988	**2.676**

In the above algorithms, using labeled data can produce an accuracy of over 90%. Moreover, when interval type-2 fuzzy sets are used, the accuracy can further increase to over 95%. The hybrid algorithms with PSO can help find the appropriate parameters resulting in a significant increase in efficiency compared to the same algorithm without using PSO technique.

Table 3.17: The various validity indexes for Hanoi central area

Algorithm	*PC-I*	*CE-I*	*D-I*	*XB-I*	$\tau - I$	*MSE*	*CS-I*	*IQI*
SFCM	0.5821	0.5973	0.2654	2.1734	NaN	17.3674	1.1672	0.6199
SFCM-PSO	0.7424	0.3684	0.3652	1.1945	NaN	12.8769	0.8942	0.7099
SPFCM-W	0.6873	0.4742	0.2985	1.0986	0.1438	16.9823	0.7832	0.7519
SPFCM-SS	0.7084	0.3587	0.3492	0.8764	0.1499	13.8751	0.6381	0.7645
GSPFCM	0.8967	**0.2279**	0.5767	0.1789	0.0928	10.8692	0.6268	0.7647
SMKFCM	0.8776	0.4098	0.4798	0.6912	NaN	14.7862	0.5767	**0.8276**
SIIT2FCM	0.8893	0.2317	0.4938	0.1986	NaN	14.1893	0.5798	0.7988
GIT2SPFCM	0.9165	0.2985	**0.5981**	0.1783	0.1078	10.1621	0.2948	0.8052
GIT2SPFCM-PSO	**0.9243**	**0.2279**	0.5979	**0.1697**	**0.0921**	**9.8276**	**0.2764**	**0.8276**

In Table 3.17, GIT2SPFCM-PSO algorithm gives the best results in most indicators, followed by GIT2SPFCM algorithm. However, in a few cases, algorithms GSPFCM and SMKFCM gives better results. Specifically, the CE-I index by algorithms GSPFCM and GIT2SPFCM-PSO is 0.2279 better than the rest of the algorithms, the IQI index by algorithms SMKFCM and GIT2SPFCM-PSO is 0.8276 better than the rest of the algorithms.

Experiment 2: Landsat-8 image

In this experiment, Landsat-8 multispectral image with the spatial resolution of $30m$ (bands 2, 3, 4, 5, 6, and 7) acquired from 2016 in Nghe An province, Vietnam was used ($104^0 44' 11.4162" E$, $19^0 29' 1.3803" N$, $105^0 34' 33.0556" E$, $19^0 09' 32.1210" N$).

Figure 3.11: RGB color image: Quy Hop district, Nghe An province in Vietnam

Table 3.18: Parameters achieved when implementing GIT2SPFCM-PSO algorithm for Quy Hop area

Algorithm	m	m_1	m_2	η	η_1	η_2	a	b
GIT2SPFCM-PSO	2.2876	1.4764	3.4565	2.1876	1.3768	3.3764	0.3764	0.3798

In Table 3.18, the parameters achieved by GIT2SPFCM-PSO algo-

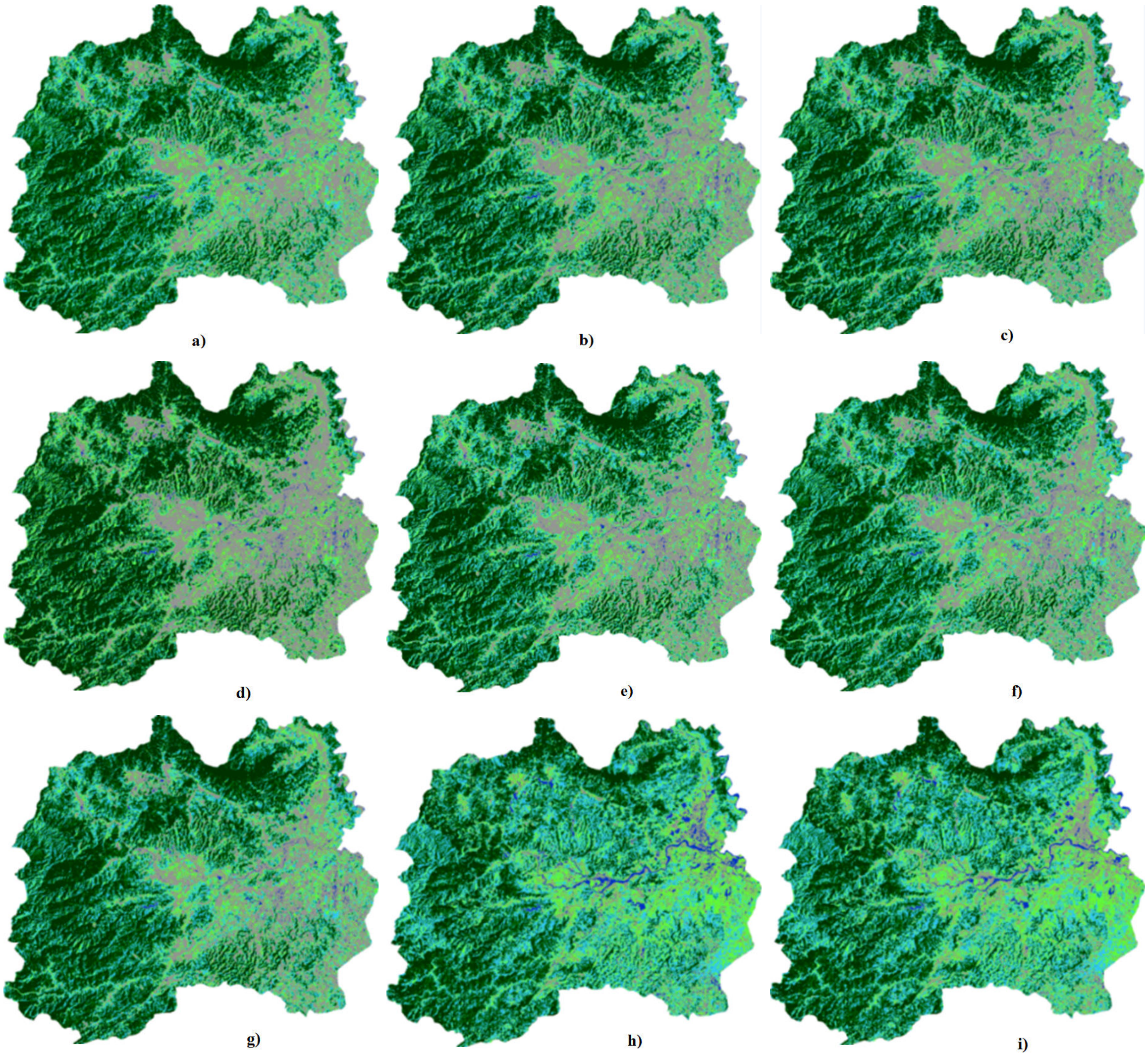


Figure 3.12: Land cover classification results of Quy Hop area: a) SFCM; b) SFCM-PSO; c) SPFCM-W; d) SPFCM-SS; e) GSPFCM; f) SMKFCM; g) SIIT2FCM; h) GIT2SPFCM; i) GIT2SPFCM-PSO

rithm show that there are small differences between the parameters m, η and a, b. Figure 3.12 is the result of land cover classification by using algorithms a) SFCM; b) SFCM-PSO; c) SPFCM-W; d) SPFCM-SS; e) GSPFCM; f) SMKFCM; g) SIIT2FCM; h) GIT2SPFCM; i) GIT2SPFCM-PSO.

Table 3.19 shows the correct classification rate according labeled pixels. GIT2SPFCM-PSO algorithm gives the best classification results

Table 3.19: Correct classification rate for Quy Hop area by labeled data (%)

Algorithm	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Class 6*	*Total*
SFCM	92.14	89.49	86.32	89.34	85.38	86.28	88.16
SFCM-PSO	93.35	93.27	91.21	92.26	89.48	87.91	91.25
SPFCM-W	92.48	92.34	91.77	90.33	89.99	86.76	90.61
SPFCM-SS	93.89	92.11	89.46	90.89	90.45	88.69	90.92
GSPFCM	99.31	98.56	97.49	98.87	95.92	94.11	97.38
SMKFCM	98.54	96.82	94.22	92.88	94.56	93.65	95.11
SIIT2FCM	98.28	95.89	96.46	94.82	96.32	91.33	95.52
GIT2SPFCM	99.23	97.39	95.87	96.65	98.79	98.07	97.67
GIT2SPFCM-PSO	99.84	99.12	98.87	97.64	98.79	98.93	**98.87**

with the correct classification rate reaching 98.87%, while the lowest rate is 88.16% by the SFCM algorithm. GIT2SPFCM algorithm has a correct classification rate of 97.67%, while the figures for GSPFCM, SIIT2FCM, and SMKFCM algorithms are 97.38%, 95.52%, and 95.11%, respectively.

Table 3.20: Land-cover classification results and VNRSC data (km^2) for Quy Hop area

Algorithm	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Class 6*
VNRSC	2.65	174.01	137.81	196.58	228.58	203.41
SFCM	6.18	191.25	166.95	155.67	204.23	218.77
SFCM-PSO	5.10	206.74	141.73	184.46	207.79	197.19
SPFCM-W	5.28	191.25	148.95	165.57	219.52	212.46
SPFCM-SS	5.19	188.64	161.19	173.66	216.79	197.56
GSPFCM	3.48	181.55	133.97	183.03	249.60	191.39
SMKFCM	4.92	198.43	142.04	182.05	221.20	194.41
SIIT2FCM	4.38	199.19	138.96	192.10	228.72	179.69
GIT2SPFCM	3.19	195.52	140.13	190.03	229.62	185.63
GIT2SPFCM-PSO	2.74	183.61	144.36	189.34	228.90	194.09

Table 3.20 and Table 3.21 show the classification result according to the area (km^2) of the land covers and VNRSC data. In this experiment, only GIT2SPFCM-PSO algorithm produces an the accuracy of over 95%, followed by the interval type-2 fuzzy clustering algorithms GIT2SPFCM and SIIT2FCM with the deviations of 5.274% and 5.980% respectively. Meanwhile, SFCM algorithm achieves an accuracy of less than 90%.

Table 3.21: Land-cover classification results and VNRSC data (%) for Quy Hop area

Algorithm	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Class 6*	*Differrence*
VNRSC	0.281	18.452	14.613	20.845	24.239	21.570	0.000
SFCM	0.655	20.280	17.703	16.507	21.656	23.198	13.841
SFCM-PSO	0.541	21.923	15.030	19.561	22.035	20.911	8.295
SPFCM-W	0.560	20.280	15.795	17.557	23.278	22.530	8.498
SPFCM-SS	0.550	20.004	17.092	18.415	22.989	20.949	8.601
GSPFCM	0.369	19.252	14.207	19.409	26.468	20.295	6.234
SMKFCM	0.522	21.042	15.061	19.304	23.456	20.615	6.558
SIIT2FCM	0.465	21.122	14.735	20.370	24.253	19.055	5.980
GIT2SPFCM	0.338	20.733	14.860	20.151	24.348	19.684	5.274
GIT2SPFCM-PSO	0.290	19.470	15.308	20.077	24.273	20.581	**3.513**

Table 3.22 shows some indicators, as can be seen, GIT2SPFCM-PSO algorithm gives the best clustering results in many indicators, except the $\tau - I$ index. SIIT2FCM algorithm achieves the best results with the PC-I index along with GIT2SPFCM-PSO algorithm.

Table 3.22: The various validity indexes for Quy Hop area

Algorithm	*PC-I*	*CE-I*	*D-I*	*XB-I*	$\tau - I$	*MSE*	*CS-I*	*IQI*
SFCM	0.4986	0.8753	0.3652	1.7824	NaN	12.9876	0.7632	0.5641
SFCM-PSO	0.5589	0.8378	0.4092	1.0914	NaN	10.3871	0.7049	0.6365
SPFCM-W	0.5437	0.8593	0.3968	1.2874	0.1322	11.7842	0.6254	0.6806
SPFCM-SS	0.5519	0.8468	0.3981	0.1789	0.1299	10.9682	0.5680	0.6898
GSPFCM	0.6537	0.6754	0.5078	0.1427	0.0567	8.9827	0.3442	0.6799
SMKFCM	0.6173	0.7863	0.4762	0.3278	NaN	9.5737	0.3267	0.8753
SIIT2FCM	**0.8763**	0.7382	0.4918	0.3148	NaN	8.5642	0.2986	0.8878
GIT2SPFCM	0.8651	0.6529	0.5289	0.1328	0.0899	8.5601	0.4152	0.8759
GIT2SPFCM-PSO	**0.8763**	**0.6281**	**0.5473**	**0.1147**	**0.0668**	**7.3654**	**0.2531**	**0.9073**

Experiment 3: Sentinel-2A image

In this experiment, four bands of Sentinel 2A image (bands 2, 3, 4, and 8) with the spatial resolution of $10m$, acquired in the mountainous area of Vinh Phuc province on $20 December 2017$ were used.

Table 3.23: Parameters achieved when implementing GIT2SPFCM-PSO algorithm for Vinh Phuc area

Algorithm	$\boldsymbol{m}$	m_1	m_2	η	η_1	η_2	$\boldsymbol{a}$	$\boldsymbol{b}$
GIT2SPFCM-PSO	2.2653	1.4762	3.0984	2.1987	1.6872	2.9875	0.7653	0.7759

Figure 3.13: RGB color image: the mountainous area of Vinh Phuc province

Table 3.24: Correct classification rate for Vinh Phuc area by labeled data (%)

Algorithm	***Class 1***	***Class 2***	***Class 3***	***Class 4***	***Class 5***	***Class 6***	***Total***
SFCM	94.65	92.42	89.31	88.08	90.27	89.46	90.70
SFCM-PSO	96.19	93.21	91.43	89.88	90.90	91.73	92.22
SPFCM-W	96.65	94.22	90.56	91.95	89.81	92.95	92.69
SPFCM-SS	96.73	95.29	93.64	89.81	90.18	91.87	92.92
GSPFCM	97.56	96.40	94.62	95.78	92.34	93.48	95.03
SMKFCM	98.39	95.37	96.37	93.44	92.54	94.36	95.08
SIIT2FCM	98.22	97.28	95.99	96.43	94.77	93.10	95.97
GIT2SPFCM	99.28	99.34	98.26	98.49	98.21	99.43	98.84
GIT2SPFCM-PSO	100.00	99.67	99.11	99.24	99.55	98.78	**99.39**

Figure 3.14 is the result of classification by six land-covers by algorithms a) SFCM; b) SFCM-PSO; c) SPFCM-W; d) SPFCM-SS; e) GSPFCM; f) SMKFCM; g) SIIT2FCM; h) GIT2SPFCM; i) GIT2SPFCM-PSO. Table 3.23 shows the parameters that were achieved when GIT2SPFCM-PSO algorithm was completed. Table 3.24 shows the correct classification rate for each land cover based on labeled data. GIT2SPFCM-PSO

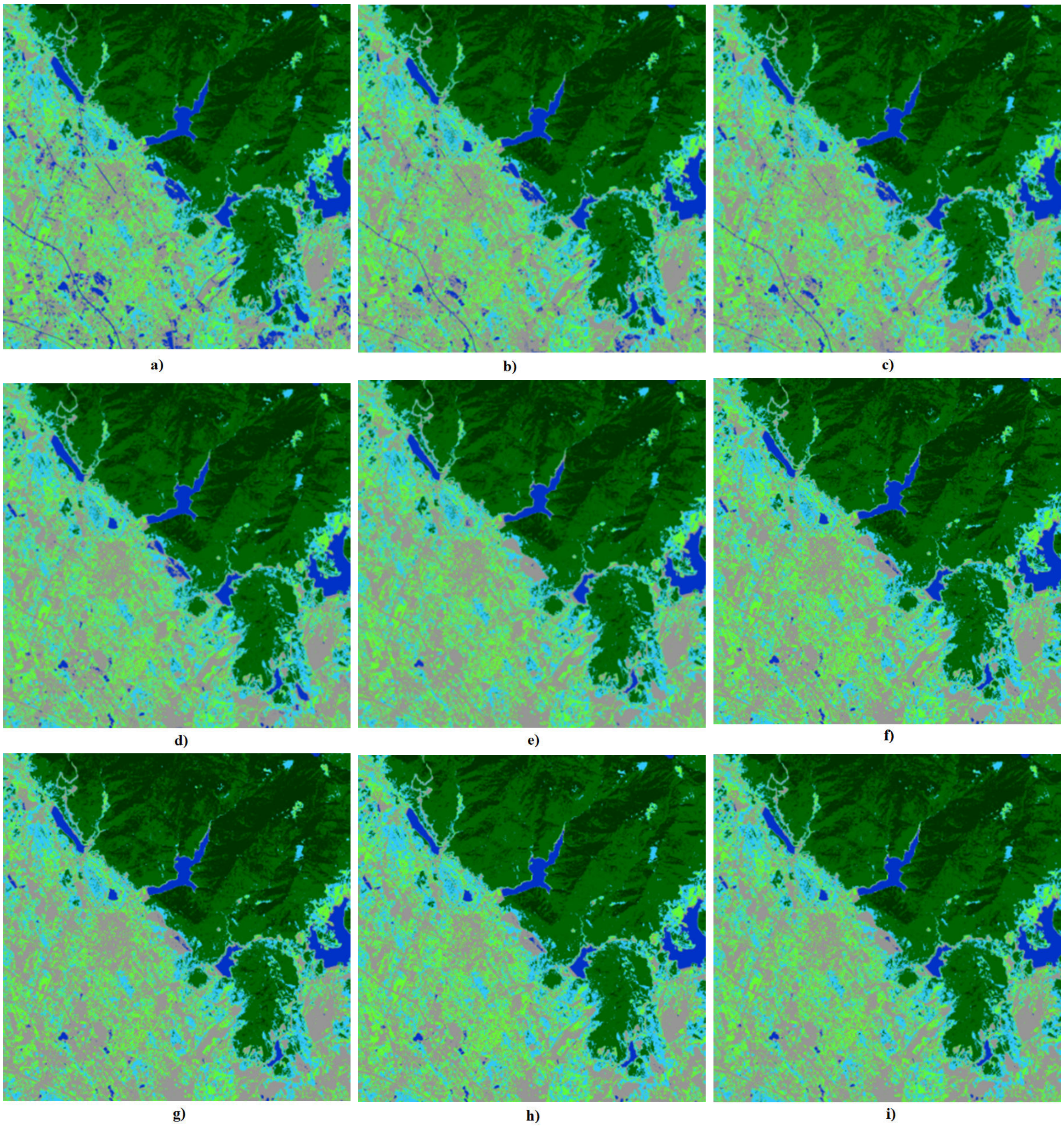


Figure 3.14: Land cover classification results of Vinh Phuc area: a) SFCM; b) SFCM-PSO; c) SPFCM-W; d) SPFCM-SS; e) GSPFCM; f) SMKFCM; g) SIIT2FCM; h) GIT2SPFCM; i) GIT2SPFCM-PSO

algorithm gives the best classification results, with the correct classification rate of the labeled pixels reaching 99.39%, while the lowest rate is 90.70% by SFCM algorithm. It can be seen that all algorithms give an accuracy of over 90%, while the algorithms GSPFCM, SMKFCM, SIIT2FCM, GIT2SPFCM, and GIT2SPFCM-PSO achieve an accuracy of over 95%.

Table 3.25: Land-cover classification results and VNRSC data (km^2) for Vinh Phuc area

Algorithm	***Class 1***	***Class 2***	***Class 3***	***Class 4***	***Class 5***	***Class 6***
VNRSC	13.38	198.56	162.12	146.29	274.38	94.89
SFCM	16.17	218.49	150.10	135.23	261.69	107.94
SFCM-PSO	14.28	205.18	161.22	139.66	263.94	105.33
SPFCM-W	15.18	218.33	151.52	135.32	260.80	108.47
SPFCM-SS	10.68	222.23	159.76	134.87	256.98	105.09
GSPFCM	11.58	221.33	161.56	133.07	258.78	103.29
SMKFCM	14.01	215.57	160.30	134.83	268.08	99.44
SIIT2FCM	12.48	204.28	167.81	139.37	262.14	103.53
GIT2SPFCM	13.74	198.20	166.13	142.27	270.78	90.39
GIT2SPFCM-PSO	12.75	198.65	168.83	139.57	275.28	93.99

Table 3.26: Land-cover classification results and VNRSC data (%) for Vinh Phuc area

Algorithm	***Class 1***	***Class 2***	***Class 3***	***Class 4***	***Class 5***	***Class 6***	***Differrence***
VNRSC	1.504	22.319	18.223	16.444	30.842	10.667	0.000
SFCM	1.818	24.560	16.872	15.200	29.416	12.133	8.042
SFCM-PSO	1.606	23.064	18.122	15.699	29.669	11.840	4.039
SPFCM-W	1.707	24.542	17.032	15.211	29.316	12.193	7.902
SPFCM-SS	1.201	24.980	17.959	15.161	28.886	11.813	7.613
GSPFCM	1.302	24.879	18.161	14.958	29.089	11.611	7.007
SMKFCM	1.585	24.232	18.019	15.156	30.134	11.177	4.704
SIIT2FCM	1.403	22.963	18.863	15.667	29.467	11.637	4.507
GIT2SPFCM	1.545	22.279	18.675	15.993	30.437	10.161	1.895
GIT2SPFCM-PSO	1.434	22.329	18.978	15.689	30.943	10.566	**1.792**

Table 3.25 and Table 3.26 are areas (km^2) and percentages (%) of land covers compared to VNRSC statistics data. The classification results by GIT2SPFCM-PSO algorithm have the highest accuracy with the difference of 1.792; followed by GIT2SPFCM algorithm with 1.895. Meanwhile, the three algorithms SFCM-PSO, SMKFCM and SIIT2FCM, have differences of 4.039, 4.704 and 4.507, respectively. The remaining algorithms all have a differences of over 7%. The largest is 8.042 by SFCM algorithm.

Table 3.27 shows the various validity indexes. It can be seen that the

Table 3.27: The various validity indexes for Vinh Phuc area

Algorithm	*PC-I*	*CE-I*	*D-I*	*XB-I*	$\tau - I$	*MSE*	*CS-I*	*IQI*
SFCM	0.6986	0.3167	0.3269	0.4389	NaN	7.7836	0.6549	0.8134
SFCM-PSO	0.7826	0.5172	0.4019	2.6731	NaN	7.2389	0.5087	0.8102
SPFCM-W	0.7659	0.1687	0.2098	0.2587	0.1689	7.3519	0.5143	0.8358
SPFCM-SS	0.7981	0.1631	0.2911	0.2684	0.1573	7.1673	0.4907	0.8571
GSPFCM	0.8872	0.2789	0.6762	1.2874	0.0783	5.1738	0.4873	0.8619
SMKFCM	0.8276	0.1388	0.5781	0.2517	NaN	6.8937	0.4396	0.8885
SIIT2FCM	0.8457	0.1287	0.5987	0.2074	NaN	5.4872	**0.3869**	**0.9167**
GIT2SPFCM	0.8635	0.0938	**0.8524**	0.1983	0.0653	5.0982	0.4188	0.9078
GIT2SPFCM-PSO	**0.8902**	**0.0842**	**0.8524**	**0.1729**	**0.0376**	**4.8278**	**0.3869**	0.9166

proposed algorithms produce better results than previous algorithms. GIT2SPFCM-PSO algorithm provides the classification results with the highest accuracy, while the computational complexity is lower than that of GIT2SPFCM algorithm. Tables 3.13, 3.18, 3.23 show that the values of the parameters are different from the above experiment. The experiment on the Sentinel-2A image is for the highest accuracy, due to the fact that Sentinel-2A image has a higher resolution ($10m$) than those of Landsat-7 ETM+ and Landsat-8 images ($30m$).

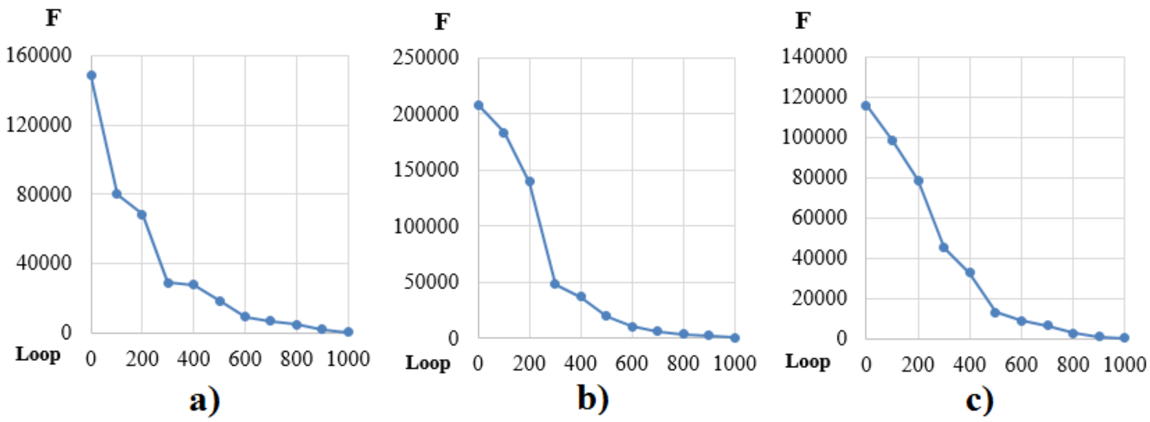


Figure 3.15: The graph of the objective function value change of the GIT2SPFCM-PSO algorithm

Figure 3.15 shows the change in the objective function's value according to the number of loops by GIT2SPFCM-PSO algorithm. Specifically, the figures 3.15.a, 3.15.b, and 3.15.c show the value of F function for Hanoi capital central area, Quy Hop district, and Vinh Phuc area, respectively. The process of finding the optimal parameter is also the process of minimizing the objective function F. It can be seen that the F function value decreases very quickly in the first 300 iterations. Then the rate decreases and reaches the smallest value when the maximum number of iterations is reached.

Table 3.28: The accuracy of the proposed algorithms on three experimental areas

Dataset	Hanoi area			Quy Hop area			Vinh Phuc area		
Algorithm	***TPR***	***FPR***	***ACC***	***TPR***	***FPR***	***ACC***	***TPR***	***FPR***	***ACC***
Erdas	89.52%	1.36%	89.18%	91.61%	1.09%	90.99%	88.76%	0.87%	89.14%
DFCM	92.67%	1.21%	92.67%	91.18%	0.87%	91.13%	92.09%	1.02%	92.01%
IFCM	93.71%	1.09%	93.39%	93.26%	1.12%	93.25%	94.64%	0.68%	94.32%
SMKFCM	98.23%	0.87%	98.11%	97.58%	0.98%	97.54%	98.45%	0.75%	98.42%
SFCM-PSO	95.48%	0.99%	95.21%	96.83%	0.79%	96.84%	95.83%	0.89%	95.78%
GIT2SPFCM-PSO	99.08%	0.58%	99.02%	98.97%	0.52%	98.77%	99.15%	0.69%	99.13%

Table 3.28 shows the accuracy of the proposed algorithms calculated by the TPR, FPR, and ACC indicators compared with Erdas software on three experimental datasets. Overall, it can be observed that the accuracy of the landcover classification using Erdas software is lower than using five proposed algorithms on all data sets. The accuracy of two unsupervised algorithms (DFCM and IFCM) is lower than that of three semi-supervised algorithms (SMKFCM, SFCM-PSO, and GIT2SPFCM-PSO). On all three datasets, GIT2SPFCM-PSO algorithm has the highest accuracy with TPR, ACC higher than 98.77%, and FPR less than 0.69%. Next, the SMKFCM algorithm, with TPR, ACC is higher than 97.54% and FPR is less than 0.98%. As can be seen, GIT2SPFCM-PSO algorithm gives the highest accuracy, followed by SMKFCM, SFCM-

PSO, IFCM, and DFCM algorithms, respectively.

Table 3.29: Implementation time (s) of the proposed algorithms on three datasets

Algorithm	Hanoi area	Quy Hop area	Vinh Phuc area
DFCM	235.465	196.254	188.264
IFCM	623.982	558.785	719.482
SMKFCM	285.522	209.095	276.541
SFCM-PSO	115.981	147.472	132.753
GIT2SPFCM-PSO	398.164	318.498	387.907

Table 3.29 provides the average values produced by ten running times of the five proposed algorithms on three datasets. SFCM-PSO algorithm has the lowest computation time, followed by DFCM, SMKFCM, GIT2SPFCM-PSO, and IFCM algorithms, respectively. Although GIT2SPFCM-PSO algorithm has the highest accuracy, the calculation time is only about half that of IFCM algorithm.

From the above experiments, the proposed method not only has reduced computational complexity but also gives higher accuracy than other algorithms. The results also show that the choice of parameters and the use of labeled data can significantly improve the accuracy of the classification algorithm.

3.5 Application in landcover change detection

In this section, RS image in Bac Binh district, Binh Thuan province from 1988 to 2017 is used to assess the land cover change including Landsat-5 TM, Landsat-7 ETM+ and Landsat-8 ($107^0 39'22.5529" E$, $11^0 33'10.9214$ and $109^0 03'54.7794" E$, $10^0 58'25.7780" N$). This is one of the two districts most severely affected by drought and desertification in the south-central coast region of Vietnam. Figure 3.16 shows RGB color image of Bac Binh district at six different times, and the detailed information on satellite image data by years is described in Table 3.30.

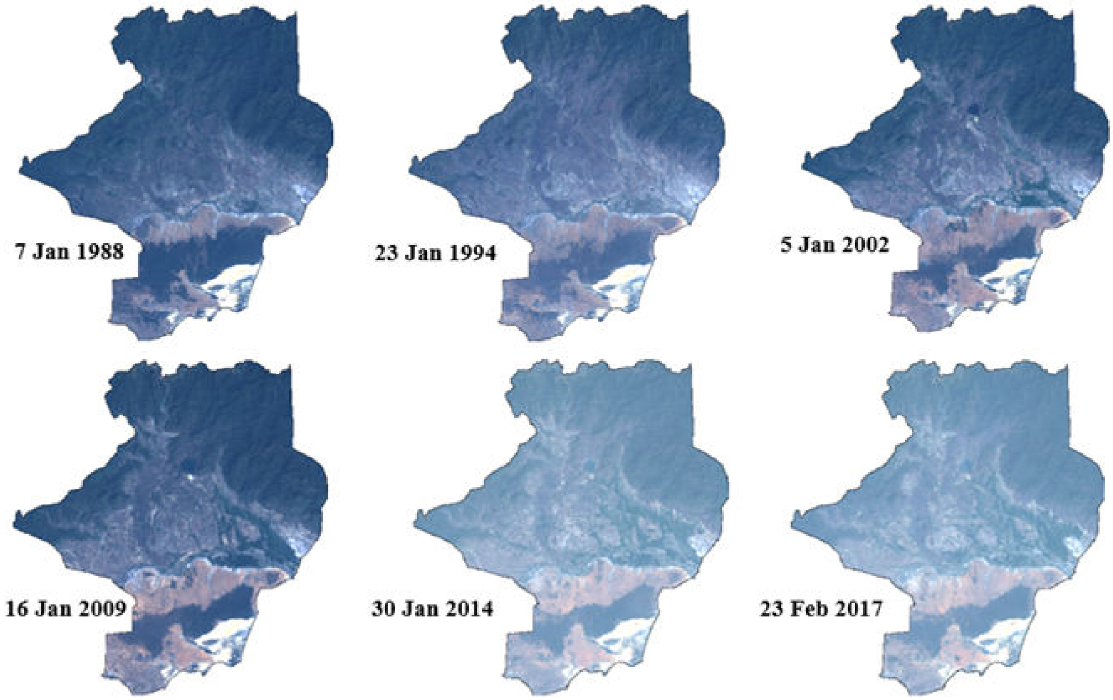


Figure 3.16: RGB color images: Bac Binh district, Binh Thuan province, Vietnam

Table 3.30: Satellite image data of Bac Binh district, Binh Thuan province, Vietnam

No.	*Image data*	*Date*	*Number of bands*	*Number of bands used*	*Spatial resolution*	*Number of bits per pixel*
1	Landsat-5 TM	07 Jan 1988	7	6	30m	8 bits
2	Landsat-5 TM	23 Jan 1994	7	6	30m	8 bits
3	Landsat-7 ETM+	05 Jan 2002	8	6	30m	8 bits
4	Landsat-7 ETM+	16 Jan 2009	8	6	30m	8 bits
5	Landsat-8	30 Jan 2014	11	6	30m	16 bits
6	Landsat-8	23 Feb 2017	11	6	30m	16 bits

Figure 3.17 shows the classification results according to six land-covers by years; a significant change in the land cover distribution can be recognized. The difference of the land cover area here is mainly the loss of forest cover to give way to agricultural land, residential land, and desertification from 1988 to 2017 shown in the soil, rock, and construction land.

Legend: Class 1: Surface water ; Class 2: Bare land ; Class

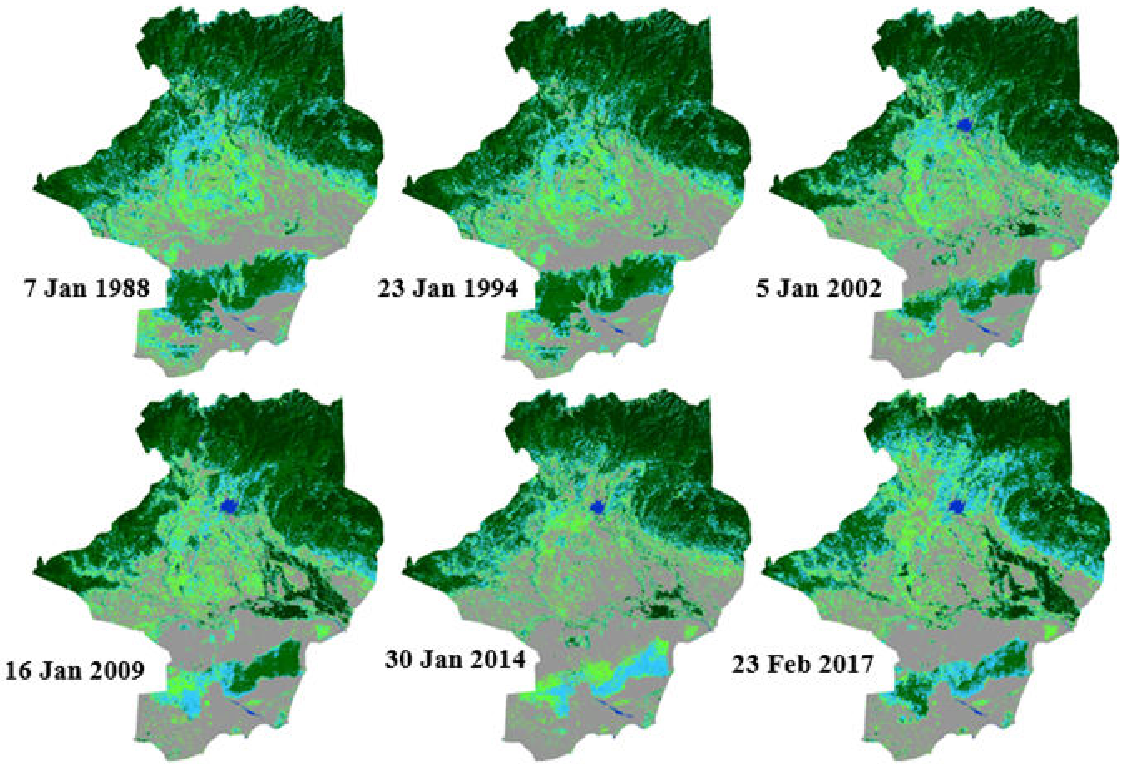


Figure 3.17: Classification results: Bac Binh district, Binh Thuan province, Vietnam

3: Grass, shrubs ; Class 4: Planted forests, low woods ; Class 5: Perennial tree crops ; Class 6: Dense vegetation .

The land cover classification result using GIT2SPFCM-PSO algorithm into six classes by percentage (%) is shown in Table 3.31 and Figure 3.18. In general, while the area of most land cover is reduced or unchanged, the area of soil, rock, and construction land class (Class 2) increase dramatically from 1988 to 2017.

Firstly, with the rivers, ponds, lakes class occupying the smallest area has increased from 0.06% to 0.325%, this is due to the appearance of a lake near the district centre. Meanwhile, the area of soil, rock, and construction land class area shows the most robust increase from 23.943% in 1988 to 37.518% in 2017. The process of urbanization is almost constant over the years and is extended to the north of Bac Binh district.

Table 3.31: Land cover classification results using GIT2SPFCM-PSO

Class/Year	*1988*	*1994*	*2002*	*2009*	*2014*	*2017*
Class 1	0.06(%)	0.09(%)	0.336(%)	0.428(%)	0.383(%)	0.325(%)
Class 2	23.943(%)	24.697(%)	28.531(%)	28.227(%)	32.19(%)	37.518(%)
Class 3	14.000(%)	13.217(%)	14.348(%)	14.024(%)	16.003(%)	14.106(%)
Class 4	17.150(%)	15.422(%)	14.236(%)	13.729(%)	16.252(%)	14.806(%)
Class 5	28.402(%)	26.981(%)	21.605(%)	24.406(%)	21.926(%)	17.628(%)
Class 6	16.444(%)	19.593(%)	20.944(%)	19.186(%)	13.806(%)	15.616(%)

Secondly, the field and grass class show almost no change in the area, although there is a change in distribution, this happens because local people here still live mainly on agriculture. Similarly, the area of dense vegetation class also changed very little from 16.444% in 1988 to 15.616% in 2017, and it can be seen that the area of dense vegetation in the North is well preserved.

Thirdly, two classes of plant forests, low woods, and perennial forest were all significantly reduced in the area, especially perennial forest class decreased significantly from 28.402% in 1988 to 17.628% in 2017, while planted forests, low woods class which decreased from 17.150% in 1988 to 14.806% in 2017. The area of the planted forests, low woods, and perennial forest classes decreased due to urbanization and the extraction of forest products by the people.

Analysis of the results showed that the general trend of land cover changes in Bac Binh district (Binh Thuan province) is an increase in the area of rocks, bare soil, construction land and a decline in forest area, including plantation forest, perennial forest, and dense vegetation. Meanwhile, types of cover such as surface water, fields and grass have not changed significantly during the period $1988 - 2017$.

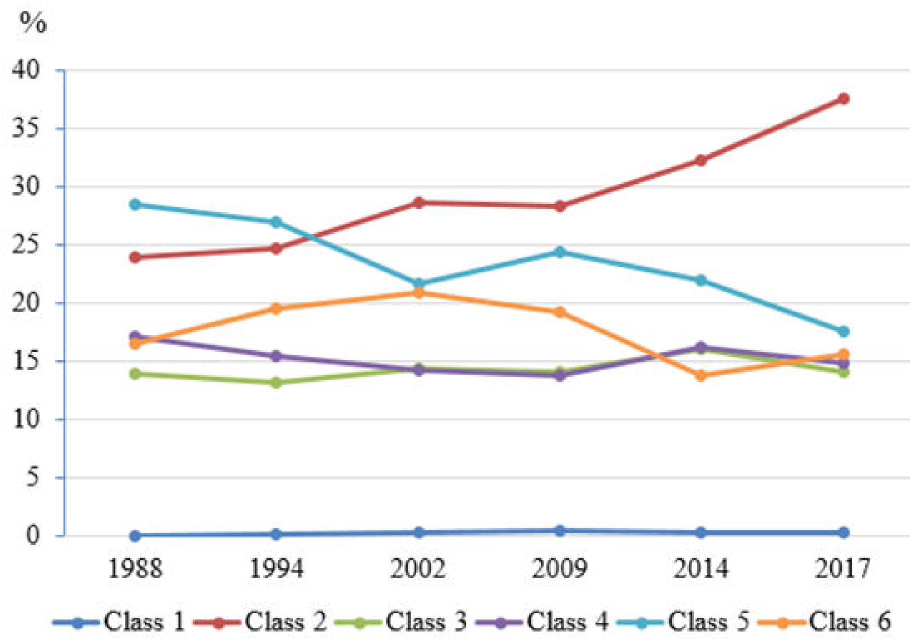


Figure 3.18: The diagram shows the land cover change by years from 1988 to 2017

The class of "rocks, bare soil, construction land" shows the most significant increase, from 23.943% of the total area of the district in 1988 to 37.518% in 2017, equivalent to 0.468% of the total area of the district per year, in which the growth rate in the period from 2009 to 2017 (1.161% per year) is much higher than the period 1988 – 2009 (0.204% per year). The increase in the area of "rocks, bare soil, construction land" in Bac Binh district can be explained by the expansion of residential areas as well as the impact of desertification here.

In the opposite direction, the strongest decline was recorded in the "perennial forest" class, with a decline rate of about 0.513% per year. The rate of decline of the "perennial forest" was also low during the period 1988 – 2009 (0.190% per year) and gained momentum in the period of 2009 – 2017 (0.753% of the total area of the district per year).

The results of land-cover classification from the proposed methods are compared with the classification results from the Erdas software (Version 2014).

Table 3.32: Land-cover classification results by the Erdas software, DFCM, IFCM, SMKFCM, SFCM-PSO, and GIT2SPFCM-PSO

Class	***Erdas***	***DFCM***	***IFCM***	***SMKFCM***	***SFCM-PSO***	***GIT2SPFCM-PSO***
Class 1	94.32(%)	98.08(%)	98.11(%)	99.19(%)	98.45(%)	**99.43(%)**
Class 2	94.25(%)	97.35(%)	96.42(%)	**98.88(%)**	97.65(%)	98.63(%)
Class 3	92.33(%)	95.76(%)	97.29(%)	97.60(%)	99.32(%)	**99.45(%)**
Class 4	96.16(%)	96.88(%)	96.34(%)	97.98(%)	97.78(%)	**99.25(%)**
Class 5	93.91(%)	97.21(%)	95.81(%)	**99.14(%)**	98.49(%)	98.76(%)
Class 6	91.79(%)	94.29(%)	97.89(%)	98.47(%)	96.23(%)	**99.05(%)**
Total	93.55(%)	95.83(%)	96.66(%)	98.64(%)	97.71(%)	**99.13(%)**

Table 3.32 shows the accuracy of the proposed algorithms based on the labeled data. It can be observed that, GIT2SPFCM-PSO algorithm gives the highest accuracy of over 99%, followed by algorithms SMK-FCM and SFCM-PSO. The unsupervised algorithms IFCM and DFCM gave worse results than the semi-supervised algorithms. However, they still give classification results with higher accuracy than those produced by Erdas software. The GIT2SPFCM-PSO algorithm produces the highest accuracy of four per six land-covers, while the SMKFCM algorithm achieves the highest accuracy of two per six remaining classes.

In summary, from the classification results, it is possible to show the land cover change over the years. The proposed method also achieves the highest accuracy when compared to labeled data, while the computational complexity of GIT2SPFCM-PSO algorithm is lower than that of GIT2SPFCM algorithm.

Experiments also show that higher resolution image data leads to higher accuracy on the same algorithm. Moreover, the semi-supervised method used in the proposed algorithm can improve efficiency, stability and reduce the risk of falling into local optimization.

The accuracy of the classification results by GIT2SPFCM-PSO algorithm is above 95% for all experiments, which indicates that the appropriateness of the parameters in clustering algorithms is very important. According to the classification results, when using some indicators to assess cluster quality, GIT2SPFCM-PSO algorithm gives the best results in most cases.

3.6 Chapter summary

This chapter presents three semi-supervised fuzzy clustering algorithms including SMKFCM, SFCM-PSO and GIT2SPFCM-PSO:

+ SMKFCM algorithm describes an new approach based on semi-supervised method for satellite images classification using kernel technique with initial centroid information retrieved from the labeled data part. The proposed methods improve the clustering results and overcome the drawbacks of the conventional clustering algorithms.

+ SFCM-PSO is a hybrid algorithm between semi-supervised method and PSO optimization technique. The PSO technique is used to find the optimal parameter for the FCM algorithm. Furthermore, the labeled data can help improve the accuracy of the proposed algorithm.

+ Meanwhile, PSO optimization technique is used in GIT2SPFCM-PSO to optimize the centroid of clusters and fuzzy parameters. The semi-supervised method is also used by adding labeled data information to the clustering process.

The classification results on some satellite images (Landsat-5 TM, Landsat-7 ETM+, Landsat-8, and Sentinel-2A) show that it is possible for the proposed method to produce higher accuracy than several previous algorithms do.

The proposed methods in this chapter were published in the Journal of Science and Technology (**2018**) [Pub2], Engineering Applications of Artificial Intelligence journal (**2018, SCIE, Q1, IF=4.2**) [Pub8], and some international conferences NICS (**2018**) [Pub4], SMC (**2018**) [Pub5], KSE (**2019**) [Pub6] and Information Sciences journal (**2020, SCI, Q1, IF=5.9**) [Pub9].

The proposed methods can significantly improve accuracy compared with some other methods. By using PSO techniques, we can achieve lower or equivalent computational complexity than algorithms that do not use them. However, they still have some limitations, such as the knowledge gained from the labeled data is only used in the proposed algorithm. The parameters of the algorithms after being found may not be useful on other data sets. This happens because surface objects are continually changing in shape, size, and color. The image data of the same object at different times may be different.

Hybrid studies with other optimization techniques to evaluate the advantages and disadvantages of each method for remote sensing image analysis problem class will be studied in the next time.

CONCLUSIONS

1. Conclusions

The dissertation has presented several robust classification models to overcome the disadvantages of current methods and apply these models to land cover classification of RS image data. The proposed method can be applied to many types of RS images (radar, optics) and spatial resolutions ($10m$, $30m$). In this dissertation, some main contributions can be summarized as follows:

The dissertation proposes two unsupervised fuzzy clustering algorithm which extended from FCM including DFCM [Pub7] and IFCM [Pub1], [Pub3]. DFCM algorithm proposes to use density information to select initial centroids for FCM algorithm. IFCM algorithm proposed the use of spectral clustering as a preprocessing step to map the original data space to a new space based on the main components.

The dissertation also develops three semi-supervised fuzzy clustering algorithms including SMKFCM [Pub8], SFCM-PSO [Pub2] and GIT2SPFCM-PSO [Pub9] which integrate the semi-supervised fuzzy clustering method [Pub4], [Pub5], [Pub6] and PSO technique. SMKFCM algorithm proposes the multiple-kernel technique to improve data separation. Moreover, the proposed method uses labeled data to adjust the focus during clustering; so the algorithm to run with greater stability. For algorithms SFCM-PSO and GIT2SPFCM-PSO, PSO technique is used for finding the optimal parameters.

The proposed algorithms all produce higher accuracy than the original algorithms. From the experimental results of the algorithms proposed in Chapter 2 and Chapter 3, some recommendations are provided as follows:

- When all data is unlabeled, DFCM and IFCM algorithms should be used. The land-cover classification results by IFCM algorithm provide better accuracy than DFCM algorithm, while DFCM algorithm has smaller computational complexity than IFCM algorithm.

- When very little data is labeled, SMKFCM, SFCM-PSO, and GIT2SPFCM-PSO algorithms should be used. GIT2SPFCM-PSO algorithms give the highest accuracy, while SFCM-PSO is suitable for large data cases because they have lower computational complexity than GIT2SPFCM-PSO and SMKFCM algorithms. The GIT2SPFCM-PSO algorithm can work well with highly uncertain data, while SMKFCM works well with overlapping data.

Experiments in the dissertation have shown that the proposed methods can overcome some disadvantages and produce higher accuracy in most cases than several other methods. They still have some limitations, such as:

- In principle, the proposed methods can work with any dimensional image data, but in fact, it has not been applied to hyperspectral image data. Applications for hyperspectral image often requires a massive amount of calculations, which is only feasible when a parallel computing model or high-performance computing based on graphics processing units (GPUs) is employed.

- The parameters of the algorithms established in the above experiments may not be useful on other data sets. This is due to the fact that surface objects are continually changing in shape, size, and color. Image data of the same object in different periods may be different.

2. Future works

Although the proposed methods in the dissertation can overcome disadvantages and give better results than several previous approaches. Most algorithms still face difficulty working with large data and multidimensional data. The author believes that further research in this direction can succeed in speeding up calculations and optimizing parameters for algorithms, reducing data dimensions and learning based on deep learning.

- Speed up the calculation: With the explosion of information and data, most algorithms have difficulty facing "big data". Several approaches, including parallel processing, high-performance computing based on GPU architecture, are suggested for this research direction.

- Dimensional reduction: RS image data is often characterized by many dimensions and large capacity, especially hyperspectral RS image; the number of dimensions can be up to hundreds or more. Therefore, reducing the size to eliminate unnecessary attributes (features) will help the algorithms work more effectively.

- Deep learning: For supervised classification problem, it requires a large amount of labeled data for training. While traditional learning algorithms are ineffective, deep learning can solve this problem well. Therefore, this might be a good research direction for the remote sensing image analysis problem for now and in the future.

PUBLICATIONS

Pub 1. **Mai D.S**, Ngo T.L, Trinh L.H, (2018). Spatial-spectral fuzzy k-Means clustering for remote sensing image segmentation. *Vietnam Journal of Science and Technology, VAST*, 56(2), pp.257–272.

Pub 2. **Mai D.S**, Ngo T.L, Trinh L.H, (2018). A hybrid approach of fuzzy clustering and Particle Swarm Optimization method for Landcover classification. *Journal of Science and Technology, Section on Information and Communication Technology, Le Quy Don Technical University*, No. 12, pp.48–63.

Pub 3. **Mai D.S**, Ngo T.L, Trinh L.H, (2018). Satellite Image Classification based Spatial-Spectral Fuzzy Clustering Algorithm. *The* 10^{th} *Asian Conference on Intelligent Information and Database Systems (ACIIDS), Springer LNAI 10752*, pp. 505—518.

Pub 4. **Mai D.S**, Ngo T.L, (2018). Semi-supervised method with Spatial weights based Possibilistic fuzzy C-means clustering for Land-cover Classification. *The* 6^{th} *NAFOSTED Conference on Information and Computer Science (NICS), IEEE*, pp. 406–411.

Pub 5. **Mai D.S**, Ngo T.L, Trinh L.H, (2018). Advanced Semi-supervised Possibilistic Fuzzy C-means Clustering using Spatial-Spectral distance for Land-cover Classification. *The IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 4375–4380.

Pub 6. **Mai D.S**, Ngo T.L, (2019). General Semi-supervised Possibilistic Fuzzy c-Means Clustering for Land-cover Classification, *The* 11^{th} *IEEE International Conference on Knowledge and Systems Engineering (KSE)*, pp. 1–6.

Pub 7. Trinh L.H, **Mai D.S**, (2019). Classification of remote sensing imagery based on density and fuzzy c-means algorithm. *International Journal of Fuzzy System Applications*, Vol.8 (2), pp. 1–15, (**Scopus, Q2**).

Pub 8. **Mai D.S**, Ngo T.L, (2018). Multiple Kernel Approach to Semi-Supervised Fuzzy Clustering Algorithm for Land-Cover Classification. *Engineering Applications of Artificial Intelligence*, Vol.68, pp. 205—213, (**SCIE, Q1, IF=4.2**).

Pub 9. **Mai D.S**, Ngo T.L, Trinh L.H, Hani Hagras, (2020). Hybrid algorithm of Interval Type-2 Semi-supervised Possibilistic Fuzzy c-Means clustering and Particle Swarm Optimization for Satellite Image Analysis, *Information Sciences*, Vol. 548, pp. 398-422 (**SCI, Q1, IF=5.9**).

BIBLIOGRAPHY

[1] Abraham, A., (2005). Adaptation of Fuzzy Inference System Using Neural Learning. *Studies in Fuzziness and Soft Computing*, 181, pp. 53–83.

[2] Askari S., Montazerin N., Zarandi M.H. and Hakimi E., (2017). Generalized entropy based possibilistic fuzzy C-means for clustering noisy data and its convergence proof, *Neurocomputing*, 219, pp. 186–202.

[3] Ayad, A.M., Fendy, S., Matthew, A.G, and Sreenatha, G.A., (2019). An Intelligent Control of an Inverted Pendulum Based on an Adaptive Interval Type-2 Fuzzy Inference System, *FUZZ-IEEE*.

[4] Bandyopadhyay S., (2005). Satellite image classification using genetically guided fuzzy clustering with spatial information, *International Journal of Remote Sensing*, 26(3), 579–593.

[5] Benmouiza, K., and Cheknane, A., (2016). Density-based spatial clustering of application with noise algorithm for the classification of solar radiation time series. *International Conference on Modelling, Identification and Control*, pp. 279–283.

[6] Bezdek J.C., (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. *New York: Academic.*

[7] Bezdek, J.C., Ehrlich, R., & Full, W., (1984). FCM: The fuzzy c-means clustering algorithm. *Computer & Geoscience* 10(2): 191-–203.

[8] Bezdek, J., Pal, N., (1998). Some new indexes of cluster validity. *IEEE Trans. Syst. Man Cybern.* 28(3), 301—315.

[9] Buckley P.J.J, Feuring T., (2000). Evolutionary algorithm solution to fuzzy problems: Fuzzy linear programming. *Fuzzy Sets and System*, 109(1), pp. 35—53.

[10] Chao, C., Robert, J., Jamie, T., Jonathan, M.G., (2016). An Extended ANFIS Architecture and its Learning Properties for Type-1 and Interval Type-2 Models, *FUZZ-IEEE*, pp. 602–609.

[11] Chen, D., Yan, Y., and Wang, D. (2014). Density Clustering Based on Border Expanding. *International Conference on Natural Computation*, 670–674.

[12] Chen, L., Philip Chen, C.L., (2012). A multiple-kernel fuzzy c-means algorithm for image segmentation. *IEEE Transaction on Systems, Man and Cybernetics*, 41(5), 1263—1274.

[13] Chen Z. and Wang B., (2015). Semi-supervised Spectral–Spatial Classification of Hyperspectral Imagery with Affinity Scoring, *IEEE Geoscience and Remote Sensing Letters*, Vol. 12(8), pp. 1710– 1714.

[14] Chou, C.H., Su, M.C, Lai, E., (2004). A new cluster validity measure and its application to image compression. *Pattern Anal Applic*, 7(2), pp 205–220.

[15] Das, A.K., Subramanian, K., Suresh, S., (2015). An Evolving Interval Type-2 Neuro-Fuzzy Inference System and Its Meta-Cognitive Sequential Learning Algorithm. *IEEE Transactions on Fuzzy Systems*, 23(6), pp. 2080–2093.

[16] Emanuel, O.R., Patricia, M., (2019). A hybrid design of shadowed type-2 fuzzy inference systems applied in diagnosis problems. *Engineering Applications of Artificial Intelligence* 86, pp. 43-–55.

[17] Emanuel, O.R., Patricia, M., Castillo, O., (2019). Relevance of Polynomial Order in Takagi-Sugeno Fuzzy Inference Systems Applied in Diagnosis Problems, *FUZZ-IEEE*.

[18] Eduardo, R., Patricia, M., German P.A., (2019). Hybrid model based on neural networks, type-1 and type-2 fuzzy systems for 2-lead cardiac arrhythmia classification, *Expert Systems with Applications*, 126, pp. 295—307.

[19] Fernando, G., Patricia, M., Fevrier, V., Juan, R.C., (2016). Optimization with Genetic Algorithm and Particle Swarm Optimization of Type-2 Fuzzy Integrator for Ensemble Neural Network in Time Series, *FUZZ-IEEE*.

[20] Ganesh, M., Palanisamy, V., (2012). Multiple-kernel fuzzy c-means algorithm for satellite image segmentation. *Eur. J. Sci. Res.* 83(2), pp. 255—263.

[21] Genitha, C.H., Vani, K., (2013). Classification of satellite images using new fuzzy cluster centroid for unsupervised classification algorithm. *2013 IEEE Conference on Information and Communication Technologies*. pp. 203-–207.

[22] Ghosh, A., Mishra, N.S., Ghosh, S., (2011). Fuzzy clustering algorithms for unsupervised change detection in remote sensing images. *Information Sciences* 181, 699—715.

[23] Graves, D., Pedrycz, W., (2007). Fuzzy c-means, GustafsonKessel FCM, and kernel-based FCM: a comparative study. *Adv. Soft Comput.* 41, 140—149.

[24] Graves, D., Pedrycz, W., (2010). Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study. *Fuzzy Sets and System.* 161(4), 522-–543.

[25] Girolami, M., (2002). Mercer kernel-based clustering in feature space. *IEEE Trans. Neural Network* 13(3), 780-–784.

[26] Guo J. & Huo H., (2017). An Enhanced IT2FCM* Algorithm Integrating Spectral Indices and Spatial Information for Multi-Spectral Remote Sensing Image Clustering. *Remote Sensing*, 9(9), 960.

[27] Hathaway R.J, Bezdek J.C., Huband J.M., (2005). Kernelized non-Euclidean relational c-means algorithms. *Neural Parallel Sci. Comput.* 13, 305—326.

[28] Hu W., Huang Y., Wei L., Zhang F., and Li H., (2015). Deep convolutional neural networks for hyperspectral image classification, *Sensors*, pp. 1–12.

[29] Huang H.C., Chuang Y.Y., and Chen C.S., (2012). Multiple Kernel Fuzzy Clustering, *IEEE Transactions on Fuzzy Systems*, 20(1), pp.120–134.

[30] Hwang C. and Rhee F.C.H., (2007). Uncertain Fuzzy clustering: Interval Type-2 Fuzzy Approach to C-Means, *IEEE Transactions on Fuzzy Systems*, 15(1), pp.107–120.

[31] Huo, H., Guo, J. and Li, Z.L., (2018). Hyperspectral Image Classification for Land Cover Based on an Improved Interval Type-2 Fuzzy C-Means Approach, *Sensors*, 18, 363.

[32] Jang, J.S. R., (1993). ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transaction on Systems, Man and Cybernetics* 23, pp. 665—684.

[33] Ji, Z., Xia, Y., Sun, Q., Cao, G., (2014). Interval-valued possibilistic fuzzy C-means clustering algorithm. *Fuzzy Sets and System* 253, pp. 138–156.

[34] John R., (1998). Type 2 fuzzy sets: An appraisal of theory and applications, *Int. J. Uncertainty, Fuzziness, Knowledge Based System*, 6(6), pp. 563–576.

[35] Karnik N., Mendel J.M, (2001). Centroid of a type-2 fuzzy set. *Information Sciences* 132, pp. 195–220.

[36] Karnik N., Mendel J., and Liang Q., (1999). Type-2 fuzzy logic systems, *IEEE Transactions Fuzzy System*, 7, pp. 643-–658.

[37] Karnik N., Mendel J.M., (2001). Operations on Type-2 Fuzzy Sets, *Fuzzy Sets and Systems*, 122, pp.327–348.

[38] Kennedy, J., Eberhart, R., (1995). Particle Swarm Optimization. *IEEE International Conference on Neural Networks*, pp. 1942–1948.

[39] Khan, K., Rehman, S. U., Aziz, K., Fong, S., Sarasvady, S. (2014). DBSCAN: Past, present and future. *International Conference on the Applications of Digital Information and Web Technologies*, 232–238.

[40] Krishnapuram R. and Keller J., (1993). A possibilistic approach to clustering, *IEEE Transactions Fuzzy System*, 1, pp. 98–110.

[41] Krishnapuram, R. and Keller, J., (1996). The possibilistic c-Means algorithm: Insights and recommendations, *IEEE Transactions Fuzzy System*, Vol. 4(3), pp. 385—393.

[42] Li, J., Yang, L., Fu, X., Chao, F., Qu, Y., (2018). Interval Type-2 TSK+ Fuzzy Inference System, *FUZZ-IEEE*.

[43] Li H., Zhang S., Ding X., Zhang C. & Cropp R., (2016). A novel unsupervised bee colony optimization (UBCO) method for remote-sensing image classification: a case study in a heterogeneous marsh area, *International Journal of Remote Sensing*, 37(24), 5726–5748.

[44] Lin, Y.Y., Liao, S.H., Chang, J.Y., Lin, C.T., (2014). Simplified Interval Type-2 Fuzzy Neural Networks, *IEEE Transactions on Neural Networks and Learning Systems*, 25(5), pp. 959–969.

[45] Liang Q., Mendel J.M., (2000). Interval Type-2 Fuzzy Logic Systems: Theory and Design, *IEEE Transactions on Fuzzy Systems*, 8(5), pp.535–550.

[46] Linda, O. and Manic, M., (2012). General Type-2 Fuzzy C-Means Algorithm for Uncertain Fuzzy Clustering, *IEEE Transactions on Fuzzy Systems*, 20(5), pp. 883–897.

[47] Lilin J., Weidong L., Zheng S., Shasha T., (2017). Hybrid fuzzy clustering methods based on improved self-adaptive cellular genetic algorithm and optimal-selection-based fuzzy c-means, *Neurocomputing*, 249, pp. 140–156.

[48] Liu, C.-A., Guo, Z., Liu, C., Zhou, H. (2011). An image-segmentation method based on improved spectral clustering algorithm. *In: Qi, L. (ed.) ISIA. CCIS*, Vol. 86, pp. 178–184.

[49] Liu, H., Zhao, F., Jiao, L., (2012). Fuzzy spectral clustering with robust spatial information for image segmentation. *Appl. Soft Comput.* 12, 3636—3647.

[50] Liu, Y., Zhang, B., Wang, L., Wang, N., (2013). A self-trained semi-supervised SVM approach to the remote sensing land cover classification. *Computers & Geosciences* 59, 98-–107.

[51] Mai D.S., Ngo L.T., (2015). Semi-supervised fuzzy c-means clustering for change detection from multispectral satellite image. *FUZZ-IEEE*, pp.1–8.

[52] Mai D.S, Ngo L.T, (2015). Interval Type-2 Fuzzy C-Means Clustering with Spatial Information for Land-Cover Classification, *ACIIDS, Springer LNAI 9011*, pp.387–397.

[53] Maciel, L., Ballini, R., (2019). A fuzzy inference system modeling approach for interval-valued symbolic data forecasting, *Knowledge-Based Systems*, 164, pp. 139–149.

[54] Maboudi M., Amini J., Hahn M. & Saati M., (2016). Objectbased road extraction from satellite images using ant colony optimization, *International Journal of Remote Sensing*, 38(1), 179–198.

[55] Maulik, U., and Bandyopadhyay, S., (2002). Performance Evaluation of Some Clustering Algorithms and Validity Indices, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12), pp.1650–1654.

[56] Mendel J., (2017). Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. *Second Edition, Springer.*

[57] Mendel J.M. and John R.I., (2002). Type-2 fuzzy set made simple, *IEEE Transactions on Fuzzy System*, 10(2), pp.117–127.

[58] Mendel J.M., John R.I., Liu F., (2006). Interval Type-2 Fuzzy Logic Systems Made Simple, *IEEE Transactions on Fuzzy Systems*, 14(6), pp. 808–821.

[59] Mitchell, M. (1998). An Introduction to Genetic Algorithms, *MIT Press.*

[60] Melin P. & Castillo O., (2013). A review on the applications of type-2 fuzzy logic in classification and pattern recognition. *Expert Systems with Applications*, 40(13), pp. 5413—5423.

[61] Mishra, N.S., Ghosh, S, Ghosh, A., (2012). Fuzzy clustering algorithms incorporating local information for change detection in remotely sensed images. *Appl. Soft Comput.* 12, 2683—2692.

[62] Ng, A., Jordan, M., Weiss, Y, (2002). On spectral clustering: analysis and an algorithm. Advances in Neural Information Processing Systems, vol. 14. MIT Press.

[63] Nguyen D.D., Ngo L.T., Watada J., (2014). A genetic type-2 fuzzy C-means clustering approach to M-FISH segmentation. *Journal of Intelligent & Fuzzy Systems*, 27(6), pp. 3111–3122.

[64] Nguyen D.D., Ngo L.T., Pham L.T., Pedrycz W., (2015). Towards hybrid clustering approach to data classification: Multiple kernels based interval-valued Fuzzy C-Means algorithms, *Fuzzy Sets and Systems*, 279, 17–39.

[65] Nguyen D.D., Ngo L.T., Pham L.T., (2013). Multiple Kernel Interval Type-2 Fuzzy C-Means Clustering, *FUZZ-IEEE*.

[66] Nguyen L.G, and et al (2020). Novel Incremental Algorithms for Attribute Reduction from Dynamic Decision Tables using Hybrid Filter–Wrapper with Fuzzy Partition Distance. *IEEE Transactions on Fuzzy Systems*, Vol. 28(5), pp. 858–873.

[67] Nikhil R.P, Kuhu P., Keller J.M., and Bezdek J.C., (2005). A Possibilistic Fuzzy c-Means Clustering Algorithm, *IEEE Transactions on Fuzzy Systems*, 13(4), pp. 517–530.

[68] Ngo L.T., Mai D.S. and Pedrycz W., (2015). Semi-supervising interval type-2 fuzzy c-means clustering with spatial information for multi-spectral satellite image classification and change detection, *Computers & Geosciences*, 83, 1–16.

[69] Ngo L.T., Nguyen D.D., (2012). Land cover classification using interval type-2 fuzzy clustering for multi-spectral satellite imagery, *IEEE SMC*, 2371–2376.

[70] Ngo L.T., Dang T.H., Pedrycz W., (2018). Towards interval-valued fuzzy set-based collaborative fuzzy clustering algorithms, *Pattern Recognition*, 81, 404–416.

[71] Ngo H.H., Nguyen C.H., Nguyen V.Q, (2019). Multichannel image contrast enhancement based on linguistic rule-based intensificators, *Applied Soft Computing*, 76, 744–762.

[72] Olivas, F., Angulo, L.A, Perez, J., Caraveo, C., Valdez, F. and Castillo, O., (2017). Comparative Study of Type-2 Fuzzy Particle Swarm, Bee Colony and Bat Algorithms in Optimization of Fuzzy Controllers. *Algorithms*, 10, pp.1–27.

[73] Peherstorfer, B., Pflüger, D., and Bungartz, H. J. (2012). Clustering Based on Density Estimation with Sparse Grids. *Advances in Artificial Intelligence*, 7526, 131–142.

[74] Phong P.A., Khang T.D., Dong D.K., (2016). A fuzzy rule-based classification system using Hedge Algebraic Type-2 Fuzzy Sets, *IEEE NAFIPS*, 1–6.

[75] Pham V.N, Ngo L.T, Pedrycz W., (2016). Interval-valued fuzzy set approach to fuzzy co-clustering for data classification, *Knowledge-Based Systems*, 107, 1–13.

[76] Pham N.V., Pham L.T, Nguyen T.D., Ngo L.T., (2018). A new cluster tendency assessment method for fuzzy co-clustering in hyperspectral image analysis, *Neurocomputing*, 307, 213–226.

[77] Roy, M., Ghosh, S., Ghosh, A., (2014). A novel approach for change detection of remotely sensed images using semi-supervised multiple classifier system. *Information Sciences*. 269, 35-–47.

[78] Shawe-Taylor, J., Cristianini, N., (2004). Kernel Methods for Pattern Analysis. *Cambridge University Press*.

[79] Shi, J., Malik, J., (2000). Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 22(8), 888–905.

[80] Shihabudheen, K.V., Mahesh, M., Pillai, G.N., (2018). Particle Swarm Optimization based Extreme Learning Neuro-Fuzzy System for regression and classification, *Expert Systems with Applications*, 92, pp. 474–484.

[81] Son L.H., Tuan T.M., (2016). A cooperative semi-supervised fuzzy clustering framework for dental X-ray image segmentation. *Expert Systems with Applications*, 46, pp.380–393.

[82] Sun, L., Jing, L., Xia, X., (2006). A New Proximal Support Vector Machine for Semi-supervised Classification. *International Symposium on Neural Networks. Lecture Notes in Computer Science*, 3971, pp. 1076–1082.

[83] Sumati, V., Patvardhan, C., (2018). Interval Type-2 Mutual Subsethood Fuzzy Neural Inference System (IT2MSFuNIS), *IEEE Transactions on Fuzzy Systems*, 26(1), pp. 203–215.

[84] Truong H.Q, Ngo L.T, Pedrycz W., (2016). Advanced Fuzzy Possibilistic C-means Clustering based on Granular Computing, *IEEE-SMC*, 2576–2581.

[85] Tzortzis, G., Likas, A., (2009). The global kernel k-means algorithm for clustering in feature space. *IEEE Transactions on Neural Network*. 20(7), 1181—1194.

[86] Vargas, D.M., Funes, F.J.G., Silva, A.J.R., (2013). A fuzzy clustering algorithm with spatial robust estimation constraint for noisy color image segmentation. *Pattern Recognition*. Lett. 34, 400–413.

[87] Varma, M., Babu, B.R., (2009). More generality in efficient multiple kernel learning. *26th Annu. Int. Conf. Machine Learning*. pp. 1065-–1072.

[88] Vuppuluri, S., Patvardhan, C., Paul, S., Singh, L., Swarup, V.M., (2019). Hybrid Model of Interval Type-2 Neural Fuzzy Inference System and Mutual Subsethood with Applications, *FUZZ-IEEE*.

[89] Xiang T., and Gong S., (2008). Spectral clustering with eigenvector selection, *Pattern Recognition*, 41(3), pp. 1012-–1029.

[90] Yang M.S. and Nataliani Y., (2017). A feature-Reduction Fuzzy Clustering Algorithm Based on Feature-Weighted Entropy, *IEEE Transactions on Fuzzy Systems*, Issue: 99.

[91] Yasunori, E., Yukihiro, H., Makito, Y., & Sadaaki, M., (2009). On semi-supervised fuzzy c-means clustering, *FUZZ-IEEE*, 1119–1124.

[92] Yin X., Shu T., Huang Q., (2012). Semi-supervised fuzzy clustering with metric learning and entropy regularization, *Knowledge-Based Systems*, 35, 304—311.

[93] Yu, C., Li, Y., Liu, A., Liu, J., (2011). A novel modified kernel fuzzy c-means clustering algorithm on image segmentation. *The 14th International Conference on Computational Science and Engineering*. pp. 621-–626.

[94] Yu, X., Zhou, W., & He, H., (2014). A method of remote sensing image auto classification based on interval type-2 fuzzy c-means. *FUZZ-IEEE*, pp. 223–228.

[95] Yuan, Y., Haobo L., Lu, X., (2015). Semi-supervised change detection method for multi-temporal hyperspectral images. *Neurocomputing* 148, 363—375.

[96] Yugander, P., Babu, J.S., Sunanda, K., Susmitha, E., (2012). Multiple kernel fuzzy c-means algorithm with ALS method for satellite and medical image segmentation. *Int. Conf. on Devices, Circuits and Systems (ICDCS)*. pp. 244—248.

[97] Zadeh L.A., (1965). Fuzzy Sets, *Information and Control*, 8, pp.338–353.

[98] Zhang, D.Q., Chen, S.C., (2003). Kernel-based fuzzy and possibilistic c-means clustering. *International Conference on Artificial Neural Networks*, pp. 122-–125.

[99] Zhang, D., Tan, K., Chen, S., (2004). Semi-supervised Kernel-Based Fuzzy C-Means. *ICONIP. Lecture Notes in Computer Science*, 3316, pp. 1229–1234.

[100] Zhang J.S. and Leung Y.W., (2004). Improved Possibilistic C-Means Clustering Algorithms, *IEEE Transactions on Fuzzy Systems*, 12(2), pp. 209–217.

[101] Zhang H., Lu J., (2009). Semi-supervised fuzzy clustering: A kernel-based approach, *Knowledge-Based Systems*, 22, pp.477-–481.

[102] Zhang M., Ma J., Gong M., (2017). Unsupervised Hyperspectral Band Selection by Fuzzy Clustering with Particle Swarm Optimization, *IEEE Geoscience and Remote Sensing Letters*, 14(5), pp.773–777.

[103] Zhang H., Wang Q., Shi W., and Hao M., (2017). A Novel Adaptive Fuzzy Local Information C-Means Clustering Algorithm for Remotely Sensed Imagery Classification, *IEEE Transactions on Geoscience and Remote Sensing*, 55(9), pp. 5057–5068.

[104] Zhang, D., Chen, S., (2004). A novel kernelized fuzzy c-means algorithm with application in medical image segmentation. *Artif. Intell. Med.* 32, 37—50.

[105] Zhang, D.Q., Chen, S.C., (2003). Clustering incomplete data using kernel based fuzzy cmeans algorithm. *Neural Process. Lett.* 18(3), 155—162.

[106] Zhao, B., Kwok, J., Zhang, C., (2009). Multiple kernel clustering. *9th SIAM Int. Conf. Data Mining.* pp. 638—649.

[107] Zhao, F., Jiao, L., Liu, H., (2013). Kernel generalized fuzzy C-means clustering with spatial information for image segmentation. *Digit. Signal Process.* 23, 184–199.

[108] Zhong Y., Ma A., and Zhang L., (2014). An Adaptive Memetic Fuzzy Clustering Algorithm With Spatial Information for Remote Sensing Imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(4), pp. 1235–1248.

[109] Wang, C.; Xu, A.; Li, C.; Zhao, X., (2016). Interval type-2 fuzzy based neural network for high resolution remote sensing image segmentation. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, pp. 385-–391.

[110] Wang, W., Zhang, Y., (2007). On fuzzy cluster validity indices. *Fuzzy Sets and Systems* 158, 2095-–2117.

[111] Wang, C., Xu, A. and Li, X., (2018). Supervised Classification High-Resolution Remote-Sensing Image Based on Interval Type-2 Fuzzy Membership Function, *Remote Sensing*, 10, 710.

[112] Wang Z. and Bovik A. C., (2002). A universal image quality index. *IEEE signal processing letters*, Vol. 9(3), pp. 81–84.

[113] Wang Z. and Bovik A. C., (2009). Mean squared error: love it or leave it? A new look at signal fidelity measures. *IEEE signal processing magazine*, pp. 98–117.

[114] Wu Y., Miao Q., Ma W., Gong M., Wang S., (2018). Particle Swarm Optimization Sample Consensus Algorithm for Remote Sensing Image Registration, *IEEE Geoscience and Remote Sensing Letters*, 15(2), pp.242–246.