



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación
y Desarrollo Tecnológico

Tesis de Doctorado

Agrupamiento difuso titubeante
para tareas de segmentación de imágenes
y reconocimiento de patrones

presentada por

MC. Virna Viridiana Vela Rincón

como requisito para la obtención del grado de
Doctora en Ciencias de la Computación

Director de tesis

Dr. Dante Mújica Vargas

Cuernavaca, Morelos, México. Junio de 2022.



Centro Nacional de Investigación y Desarrollo Tecnológico
Departamento de Ciencias Computacionales

ESC\FORDOC09

Cuernavaca, Morelos, 13/mayo/2022

ASUNTO: ACEPTACIÓN DEL TRABAJO DE TESIS DOCTORAL

DR. JUAN GABRIEL GONZÁLEZ SERNA
JEFE DEL DEPARTAMENTO DE CIENCIAS COMPUTACIONALES
PRESENTE

Los abajo firmantes, miembros del Comité Tutorial de la Tesis Doctoral de la alumna M.C. Virna Viridiana Vela Rincón manifiestan que después de haber revisado su trabajo de tesis doctoral titulado "AGRUPAMIENTO DIFUSO TITUBEANTE PARA TAREAS DE SEGMENTACIÓN DE IMÁGENES Y RECONOCIMIENTO DE PATRONES", realizado bajo la dirección del Dr. Dante Mújica Vargas, el trabajo se ACEPTA para proceder a su impresión.

ATENTAMENTE
"Excelencia en Educación Tecnológica®"
"Educación Tecnológica al Servicio de México"

DR. DANTE MÚJICA VARGAS
CENIDET

DR. RAÚL PINTO ELÍAS
CENIDET

DR. NOÉ ALEJANDRO CASTRO SÁNCHEZ
CENIDET

DRA. ANDREA MAGADÁN SALAZAR
CENIDET

DR. JEAN MARIE VIANNEY KINANI
IPN UPIIH

C.c.p.: Lic. Silvia del Carmen Ortiz Fuentes / Jefa del Depto. de Servicios Escolares
Dr. Carlos Manuel Astorga Zaragoza / Subdirector Académico
Expediente

SEP TecNM CENTRO NACIONAL DE INVESTIGACIÓN
Y DESARROLLO TECNOLÓGICO

RECIBIDO
18 MAY 2022
SUBDIRECCIÓN ACADÉMICA

EDUCACIÓN SECRETARÍA DE EDUCACIÓN PÚBLICA
16 MAY 2022
CENTRO NACIONAL DE INVESTIGACIÓN
Y DESARROLLO TECNOLÓGICO
SERVICIOS EDUCACIONALES
RECIBIDO
EBN



Interior Internado Palmira S/N, Col. Palmira, C. P. 62490, Cuernavaca, Morelos
Tel. 01 (777) 3627770, ext. 3201, e-mail: dcc@tecnm.mx tecnm.mx | cenidet.tecnm.mx





Cuernavaca, Mor.,

No. De Oficio:

Asunto:

18/mayo/2022

SAC/71/2022

**Autorización de
impresión de tesis**

**VIRNA VIRIDIANA VELA RINCÓN
CANDIDATA AL GRADO DE DOCTORA EN CIENCIAS
DE LA COMPUTACIÓN
P R E S E N T E**

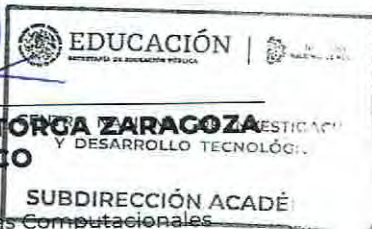
Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "AGRUPAMIENTO DIFUSO TITUBEANTE PARA TAREAS DE SEGMENTACIÓN DE IMÁGENES Y RECONOCIMIENTO DE PATRONES", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE

Excelencia en Educación Tecnológica®
"Educación Tecnológica al Servicio de México"

DR. CARLOS MANUEL ASTORZA ZARAGOZA
SUBDIRECTOR ACADÉMICO



C. c. p. Departamento de Ciencias Computacionales
Departamento de Servicios Escolares

CMAZ/CHG



Dedicatoria

*A mis padres, porque gracias ellos soy quién soy.
A José Manuel, mi compañero de vida, por su amor incondicional.
A Emma Valentina, mi hija, por inspirarme a ser mejor cada día.*

Agradecimientos

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el sustento económico para realizar mis estudios de doctorado a través de la beca otorgada.

Al Tecnológico Nacional de México/CENIDET por brindarme la oportunidad de alcanzar el grado de Doctora en Ciencias de la Computación, facilitándome sus instalaciones; así como el respaldo de su personal académico y administrativo.

Al Dr. Dante Mújica Vargas, mi director de tesis, agradezco su esfuerzo, su dedicación y la orientación que me ofreció a lo largo de estos cuatro años, en los que logré un aprendizaje muy valioso para desarrollarme profesionalmente; sin su apoyo esta investigación no sería una realidad.

A mi comité revisor, Dr. Manuel Mejia Lavalle[†], Dr. Raúl Pinto Elias, Dr. Noé A. Casto Sánchez, Dra. Andrea Magadán Salazar, Dr. Jean Marie Vianney Kinani, por el tiempo invertido, sus aportaciones y comentarios para la mejora de esta investigación.

A mis compañeros del CENIDET, gracias por su confianza, por compartir sus conocimientos conmigo, por esos gratos momentos y las risas vividas en esta travesía.

El apoyo y el amor de toda mi familia; especialmente a mis hermanos, que siempre han creído en mi; a Fabita y Paco por cobijarme como una hija; a mi tío Marco Antonio por sus buenos consejos y por siempre estar presente.

¡Gracias!

Resumen

Entre las distintas áreas de la Inteligencia Artificial se encuentra el Reconocimiento de Patrones y la Visión por Computadora, que se han beneficiado utilizando los métodos de aprendizaje automático. Normalmente, el más utilizado es el aprendizaje no supervisado, siendo un caso especial de éste el agrupamiento o *clustering*. El agrupamiento es un técnica que permite identificar grupos dentro de un conjunto de datos, de tal manera que estos grupos o *clusters* compartan ciertas características similares.

Un algoritmo clave en esta área es el algoritmo *Fuzzy C-Means*, que maneja cierto grado de pertenencia para representar a cada dato con respecto a cada uno de los grupos, permitiendo ser mas flexible que el agrupamiento clásico, de tal manera que se puede expresar la pertenencia de un dato hacia todos los grupos simultáneamente. La teoría de este algoritmo se ha ido extendiendo con la finalidad de mejorar la exactitud del grado de pertenencia asignado a un dato, tal es el caso de los conjuntos difusos tipo-2 que permiten una distribución de posibilidades, los conjuntos difusos intervalo evaluados un margen de error y los conjuntos difusos intuitivos permite un grado de pertenencia, de no pertenencia e indecisión. Sin embargo su dificultad es debido a que se tiene un conjunto de valores posibles.

Como una alternativa, los Conjunto Difusos Titubeantes (*Hesitant Fuzzy Sets*) permiten expresar el agrupamiento de los datos a través de varios posibles valores de pertenencia para un sólo elemento de un conjunto de referencia, a diferencia de las demás teorías, se puede tener varios valores posible en lugar de uno solo, permitiendo una mayor flexibilidad y un mejor desempeño en la agrupación. En esta investigación se propone un algoritmo de agrupamiento difuso mediante la incorporación de la lógica de funcionamiento de los conjuntos difusos titubeantes aplicable al reconocimiento de patrones y segmentación de imágenes.

Palabras clave: Segmentación de imágenes, Reconocimiento de patrones, Agrupamiento difuso, Conjuntos difusos titubeantes, Agrupamiento automático.

Abstract

Among the different areas of Artificial Intelligence is Pattern Recognition and Computer Vision, which benefit from the use of machine learning methodology. Typically, unsupervised learning is the most commonly used, a particular case of clustering. Clustering is a technique that allows groups of data that share similar characteristics to be identified within a data set.

One of the algorithms that stands out in this area is the Fuzzy C-Means (FCM), which handles a certain degree of membership to represent each data with respect to each of the clusters, allowing it to be more flexible than the classical clustering in such a way that the membership of a data can be expressed simultaneously to all clusters. The theory of this algorithm has been extended to improve the accuracy of the degree of membership assigned to data, such; this is the case of type-2 fuzzy sets that allow distribution of possibilities, interval-valued fuzzy sets which permit a margin of error and intuitionistic fuzzy sets to allow a degree of membership, non-membership and indecision. However, its constraint is due to the fact that it has a set of possible values.

As an alternative, Hesitant Fuzzy Sets allow data clustering with several possible membership values for a single element of a reference set; contrary to the other theories, it can have several possible values instead of one, allowing more flexibility and better performance in the clustering. This research proposes a fuzzy clustering algorithm by incorporating the operating logic of hesitant fuzzy sets applicable to pattern recognition and image segmentation.

Keywords: Image Segmentation, Pattern Recognition, Fuzzy Clustering, Hesitant Fuzzy Sets, Automatic clustering.

Índice General

Resumen	iii
Índice de Figuras	viii
Índice de Tablas	ix
1 Introducción	1
1.1. Planteamiento del problema	3
1.1.1. Delimitación del problema	4
1.1.2. Complejidad del problema	4
1.2. Objetivos	4
1.2.1. Objetivo General	4
1.2.2. Objetivos Específicos	4
1.3. Alcances y limitaciones	5
1.3.1. Alcances	5
1.3.2. Limitaciones	5
1.4. Justificación	5
1.4.1. Beneficios e importancia	6
1.5. Aportaciones científicas	6
1.6. Organización de la tesis	6
2 Marco teórico	7
2.1. Marco conceptual	7
2.1.1. Segmentación de imágenes	7
2.1.2. Reconocimiento de Patrones	8
2.1.3. Agrupamiento	9
2.1.4. <i>Fuzzy C-Means</i>	10
2.1.5. <i>Intuitionistic Fuzzy Sets</i>	12
2.1.6. <i>Hesitant Fuzzy Sets</i>	14
2.2. Estado del Arte	18
2.2.1. Antecedentes	18
2.2.2. Trabajos relacionados	19
3 Metodología	29
3.1. Formulación algoritmo <i>Hesitant Fuzzy C-Means</i>	29
3.2. Algoritmo <i>Parallel Hesitant Fuzzy C-Means</i>	32

3.3.	Algoritmo <i>Automatic Hesitant Fuzzy C-Means</i>	38
3.4.	Algoritmo <i>Fast and Automatic Hesitant Fuzzy Clustering</i>	40
4	Experimentación y resultados	44
4.1.	Introducción	44
4.2.	Conjunto de datos	44
4.3.	Métricas	45
4.4.	Métodos de comparación	48
4.5.	Experimentos	57
4.6.	Resultados	57
4.6.1.	Tiempo de ejecución	57
4.6.2.	Calidad del agrupamiento	60
4.6.3.	Discusión de los resultados	73
5	Conclusiones	74
5.1.	Objetivos y alcances logrados	74
5.2.	Resultados de la investigación	75
5.2.1.	Productos	75
5.2.2.	Aportaciones	76
5.2.3.	Conclusiones	76
5.2.4.	Trabajos futuros	77
	Referencias	78
	Apéndice	86

Índice de Figuras

1.1. Fases del procesamiento de imágenes.	1
1.2. Técnicas para segmentación de imágenes (Szeliski, 2010).	1
1.3. Grupos de diferentes formas y tamaños (Mújica Vargas, 2013).	2
1.4. Agrupamiento en $c = 5$ grupos (Mújica Vargas, 2013).	2
2.1. Ejemplo de una segmentación en tres regiones.	8
2.2. Etapas del sistema de reconocimiento de patrones (Bishop, 2006).	9
2.3. Agrupamiento de datos generados aleatoriamente (Wierzchoń and Kłopotek, 2018).	10
2.4. Representación gráfica de una función de pertenencia (Ngan et al., 2019).	12
2.5. Representación gráfica de los conjuntos difusos intuitivos (Ngan et al., 2019).	13
2.6. Representación gráfica de conjuntos difusos titubeantes (Ngan et al., 2019).	16
2.7. Optimización de rutas por el algoritmo HFS-A* (Wang et al., 2018b).	19
2.8. Ejemplo de la aplicación de los conjuntos difusos titubeantes en la toma de decisiones.	20
2.9. Resultados de la segmentación de imágenes del algoritmo FRFCM (Lei et al., 2018a).	22
2.10. Resultados de la segmentación de imágenes algoritmo DSFCM_N (Zhang et al., 2018b).	22
2.11. Resultados de la segmentación de imágenes algoritmo AFCE (Lei et al., 2019b).	23
2.12. Sistema de diagnóstico dental.	24
2.13. Arquitectura de SegNet (Badrinarayanan et al., 2017).	25
2.14. Arquitectura de SegNet (Alom et al., 2019).	25
2.15. Segmentación usando autómatas celulares (Athertya et al., 2019).	26
2.16. Ejemplo de una técnica de umbralización. (Li et al., 2020).	26
2.17. Ejemplo del grafo bipartito (Nie et al., 2019).	27
2.18. Modelo con una estructura k-means profunda de tres capas (Huang et al., 2021a).	28
3.1. Proceso <i>Fork-Join</i>	33
3.2. Implementación del algoritmo <i>Parallel Hesitant Fuzzy C-Means</i>	35
3.3. Proceso del algoritmo <i>Automatic Hesitant Fuzzy C-Means</i>	38
3.4. Proceso del algoritmo <i>Fast and Automatic Hesitant Fuzzy Clustering</i>	41
4.1. Imágenes muestra de cada base de datos.	44
4.2. Conjunto de imágenes de glóbulos blancos (WBC).	45
4.3. Resultados en bases de datos de imágenes médicas.	61
4.4. Resultado de la segmentación en base de datos médicas (ISIC2019 y WBC).	62
4.5. Resultados de la base de datos MRCORID.	63
4.6. Resultado de la segmentación en base de datos MRCORID.	64
4.7. Resultados de la base de datos Weizman.	65

4.8. Resultado de la segmentación en base de datos Weizmann.	65
4.9. Resultados de la base de datos BSDS500.	66
4.10.Resultado de la segmentación en base de datos BSDS500.	67
4.11.Gráficos de los resultados cuantitativos.	68
4.12.Resultado del índice CH.	69
4.13.Gráfica de los resultados cuantitativos.	70
4.14.Resultado de la segmentación de la imagen 66075.	71
4.15.Resultado de la segmentación.	72
1. Constancia de autor del artículo publicado en el congreso MCPR 2020.	86
2. Primera página del artículo publicado en el congreso MCPR 2020.	87
3. Constancia de autor del artículo publicado en el congreso MICAI 2020.	88
4. Primera página del artículo publicado en el congreso MICAI 2020.	89
5. Primera página del artículo publicado en la revista <i>Signal, Image and Video Processing</i>	90
6. Constancia de satisfacción de la Estancia Académica.	91
7. Sometimiento de un artículo en la revista <i>Computación y Sistemas</i>	92
8. Primera página del artículo sometido a la revista <i>Computación y Sistemas</i>	93
9. Sometimiento de artículo en la revista <i>Engineering Applications of Artificial Intelligence</i>	94
10. Primera página del artículo sometido a la revista <i>Engineering Applications of Artificial Intelligence</i>	95

Índice de Tablas

3.1. Ejemplos de código secuencial y paralelo.	33
3.2. Directivas de OpenMP utilizadas.	34
4.1. Bases de datos reales.	45
4.2. Rendimiento de ejecución promedio en bases de datos de imágenes médicas.	58
4.3. Rendimiento de ejecución promedio de la base de datos MRCORID.	58
4.4. Rendimiento de ejecución promedio de la base de datos Weizman.	58
4.5. Rendimiento de ejecución promedio de la base de datos BSDS500.	59
4.6. Tiempo de ejecución e iteraciones de cada algoritmo para cada base de datos.	59
4.7. Rendimiento de ejecución promedio.	60
4.8. Resultado promedio del agrupamiento para bases de datos médicas.	60
4.9. Resultado promedio del agrupamiento para base de datos MRCORID.	63
4.10.Resultado promedio del agrupamiento para base de datos Weizman.	64
4.11.Resultado promedio del agrupamiento para base de datos BSDS500.	66
4.12.Resultado promediado de cada algoritmo	68
4.13.Resultado promedio de la segmentación de imágenes.	70
5.1. Objetivos realizados.	74
5.2. Alcances realizados.	75

1 | Introducción

La Visión Artificial o Visión por Computadora es considerada como una cadena de procesamiento de imágenes, tal como se observa en la Figura 1.1, dentro de las etapas de esta cadena se encuentra la segmentación de imágenes, uno de los problemas más antiguos y estudiados que es definida como un proceso mediante el cual una imagen es dividida en distintas regiones, pertenecientes a diferentes objetos, de forma que cada región cumple cierto criterio de homogeneidad (Gonzalez et al., 2002; Szeliski, 2010).

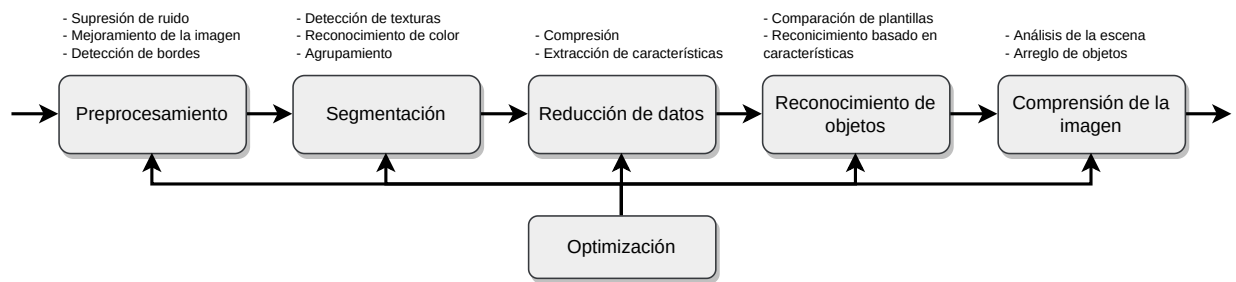


Figura 1.1: Fases del procesamiento de imágenes.

En la literatura existen diferentes técnicas para segmentar imágenes, principalmente se dividen en dos tipos: a) locales, que son basadas en bordes, regiones y píxeles y b) globales que son por clases o instancias. Algunos ejemplos de las técnicas más populares se exhiben en la Figura 1.2.

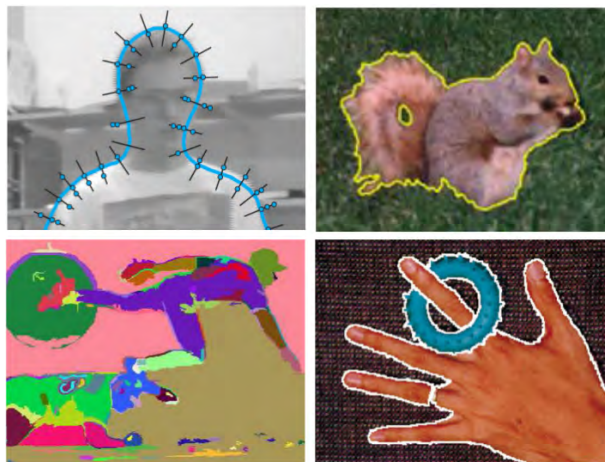


Figura 1.2: Técnicas para segmentación de imágenes (Szeliski, 2010).

Dentro de las técnicas para segmentación de imágenes por píxeles se encuentra el agrupamiento (*Clustering*) también conocido como aprendizaje no supervisado, que es una de las actividades más importantes del hombre, la cual permite explorar la naturaleza desconocida de un conjunto de datos (Bezdek et al., 1999). El agrupamiento descompone un conjunto dado de objetos en subgrupos o agrupaciones en función de la similitud. Su objetivo es dividir un conjunto de datos en grupos, de forma que los objetos pertenecientes al mismo grupo sean lo más parecido posible, mientras que aquellos que pertenecen a diferentes grupos sean lo más distinto posible (De Oliveira and Pedrycz, 2007). En la Figura 1.3 se observan algunos ejemplos de agrupamiento de diferentes formas y tamaños. El agrupamiento no sólo es aplicado en la segmentación de imágenes, también en otras áreas como lo es el reconocimiento de patrones.

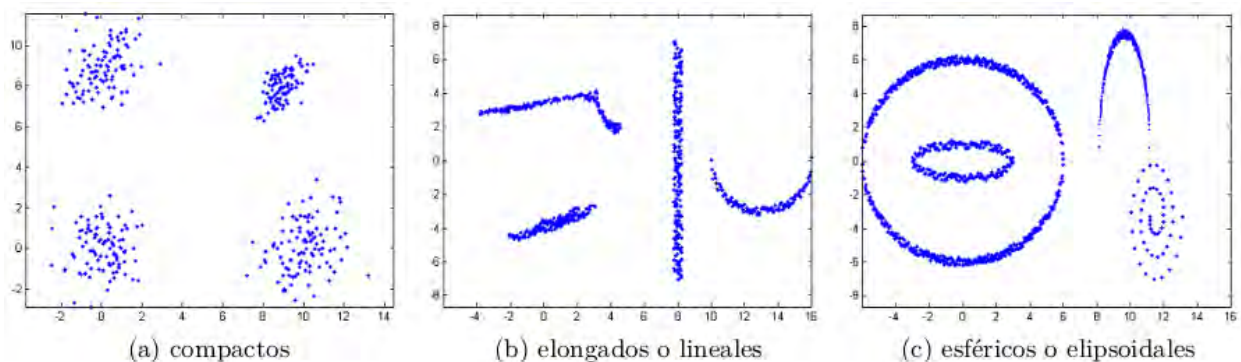


Figura 1.3: Grupos de diferentes formas y tamaños (Mújica Vargas, 2013).

El algoritmo más conocido de agrupamiento es *K-means*, en este modelo cada dato dentro de un conjunto de datos se asigna a un solo grupo (Bezdek et al., 1999). Dado que es un algoritmo rígido y no permite que un dato pertenezca en cierto grado a dos grupos esta teoría se ha extendido al algoritmo *Fuzzy C-Means* (FCM), el cual permite la pertenencia gradual en un intervalo cerrado $\mu \in [0, 1]$ de los datos con respecto a los grupos o regiones de interés. Esta flexibilidad permite expresar la pertenencia de un dato a todos los grupos o regiones en forma simultánea (Bezdek et al., 1999; De Oliveira and Pedrycz, 2007). En la Figura 1.4 se observa como se comportan los dos algoritmos mencionados dado un conjunto de datos, mostrando que el algoritmo FCM es capaz de definir los límites que separan los cinco grupos, a diferencia del otro algoritmo que presenta un traslape de dos grupos.

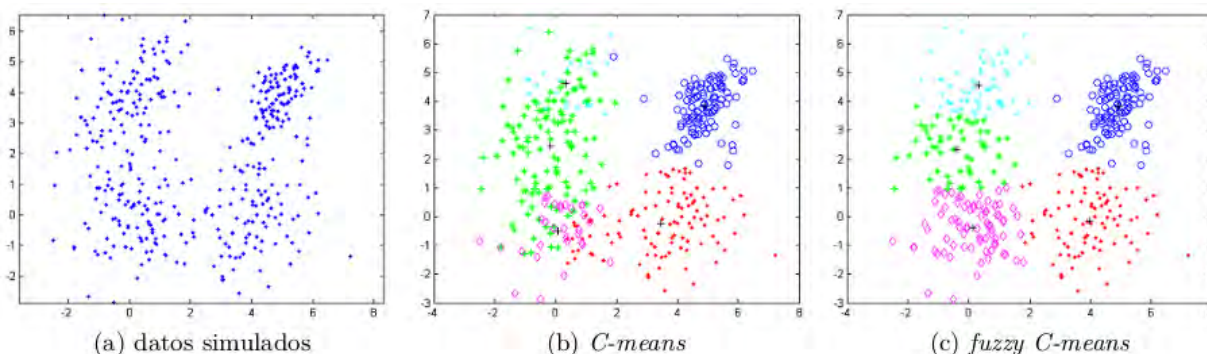


Figura 1.4: Agrupamiento en $c = 5$ grupos (Mújica Vargas, 2013).

Dado que los grados de pertenencia de un conjunto difuso son expresados en valores precisos extraídos del intervalo $[0, 1]$, no se puede capturar la capacidad humana para expresar grados de pertenencia imprecisos o vagos. Por ejemplo, en el entorno real, específicamente en la toma de decisiones, puede surgir imprecisión debido a información incompleta, información no obtenible, ignorancia parcial, entre otras. Por lo que, para un tomador de decisiones (o un experto) le es difícil determinar un número tan claro, pero, si se le pide a un grupo de tomadores de decisiones que evalúen las alternativas candidatas, a menudo encuentran algunos desacuerdos y se tienen opiniones diferentes sobre las alternativas, estas no pueden persuadirse fácilmente y es difícil obtener un resultado de consenso, lo que se obtiene es un conjunto de posibles valores. En tal caso, el conjunto difuso tradicional no se puede usar para describir las opiniones del grupo. Por lo tanto, la teoría de conjuntos difusos clásica se ha extendido en varias formas diferentes, como los conjuntos difusos intuitivos (IFS: *Intuitionistic Fuzzy Sets*), los conjuntos difusos intervalo-evaluados, los conjuntos difusos de tipo-2, entre otros. Todas estas extensiones están basadas en el mismo razonamiento: asignar el grado de pertenencia de un elemento a un conjunto fijo no es una tarea clara. Por ejemplo, los IFS asignan a cada elemento un grado de pertenencia $\mu_F(x)$, un grado de no pertenencia $\nu_F(x)$ y un grado de indecisión $\pi_F(x)$, y eso permite que el método sea más poderoso que el conjunto difuso para tratar la vaguedad y la incertidumbre (Liao and Xu, 2017).

Sin embargo, la dificultad de establecer el grado de pertenencia no se debe a tener alguna distribución de posibilidades (como en el conjunto difuso tipo-2) o un margen de error (como en los conjuntos difusos intervalo evaluados y IFS), sino porque se tiene un conjunto de valores posibles. En tales casos, una alternativa para dar solución a los problemas antes mencionados es mediante Conjuntos Difusos Titubeantes (*Hesitant Fuzzy Sets*, HSF), como una generalización del conjunto difuso. Los HFS permiten que el grado de pertenencia de un elemento a un conjunto sea representado por varios valores posibles entre 0 y 1 y pueden describir mejor las situaciones en las que las personas tienen dudas al proporcionar sus preferencias sobre los objetos en el proceso de toma de decisiones. Los HFS fueron propuestos originalmente por Torra (Liao and Xu, 2017; Xu, 2016).

Con esta tesis doctoral se introduce en el Estado del Arte un algoritmo de agrupamiento difuso mediante la incorporación de la lógica de funcionamiento de los conjuntos difusos titubeantes; es decir, se genera un conjunto de valores de pertenencia simultáneamente y se hace una evaluación de tal forma que el algoritmo seleccione el mejor valor para el agrupamiento; aplicable al reconocimiento de patrones y segmentación de imágenes.

1.1. Planteamiento del problema

El agrupamiento de píxeles o patrones basado en los conjuntos difusos clásicos, aplicados a la segmentación de imágenes y el reconocimiento de patrones, son expresados mediante un valor preciso, que representa el grado de pertenencia del dato hacia cierto grupo, pero no pueden expresar valores imprecisos o vagos como lo haría un humano, éstos pueden ser dados por información incompleta. Por lo tanto, esta teoría se ha extendido hacia los conjuntos difusos intuitivos, en los cuales además del grado de pertenencia, se asigna un grado de no pertenencia y un grado de indecisión, permitiéndole ser un método más eficiente para tratar la vaguedad. Sin embargo, la dificultad para establecer el grado de pertenencia radica en que se tiene un conjunto de valores posibles, por esta razón se han propuesto los conjuntos difusos titubeantes, que permiten describir mejor las situaciones donde se tienen dudas al proporcionar sus preferencias sobre un objeto, permitiendo establecer varios valores posibles de pertenencia para un solo elemento de un conjunto de referencia. Éste tipo

de conjuntos han sido aplicados en la toma de decisiones, por lo que se pretende obtener un mejor agrupamiento de píxeles o patrones aplicando esta teoría.

Hipótesis: Es posible lograr un mejor agrupamiento de píxeles o patrones integrando la teoría de los conjuntos difusos titubeantes, los cuales aunado a múltiples atributos consideran múltiples decisiones, en forma simultánea.

1.1.1. Delimitación del problema

En esta investigación se desarrollan las expresiones matemáticas que definen la función objetivo y las expresiones de actualización de un algoritmo de agrupamiento basado en conjuntos difusos titubeantes (*Hesitant Fuzzy Sets*), en el que mediante datos reales se evalúa la calidad del agrupamiento para ser aplicado al reconocimiento de patrones y la segmentación de imágenes.

1.1.2. Complejidad del problema

La complejidad del problema radica en que al ser un enfoque relativamente nuevo, se requiere generar el conocimiento, es decir, las bases matemáticas del algoritmo de agrupamiento, para que sea implementado en el reconocimiento de patrones y la segmentación de imágenes.

1.2. Objetivos

1.2.1. Objetivo General

Proponer, implementar y evaluar un algoritmo de agrupamiento basado en Conjuntos Difusos Titubeantes (*Hesitant Fuzzy Sets*), el cual pueda ser utilizado para la segmentación de imágenes y el reconocimiento de patrones.

1.2.2. Objetivos Específicos

- Estudiar el algoritmo de agrupamiento *Fuzzy C-Means* y sus variantes.
- Estudiar y comprender la teoría de los Conjuntos Difusos Titubeantes y su aplicación en la segmentación de imágenes y el reconocimiento de patrones.
- Proponer un algoritmo de agrupamiento basado en la teoría de los conjuntos difusos titubeantes.
- Evaluar la calidad del agrupamiento con datos sintéticos.
- Aplicar y evaluar el algoritmo propuesto en el reconocimiento de patrones y la segmentación de imágenes médicas.

1.3. Alcances y limitaciones

1.3.1. Alcances

- Analizar dos algoritmos de agrupamiento difusos existentes.
- Analizar la teoría de los conjuntos difusos titubeantes.
- Plantear y desarrollar un algoritmo de agrupamiento basado en los conjuntos difusos titubeantes.
- Evaluar la calidad de agrupamiento del algoritmo propuesto con al menos dos métricas definidas en la literatura.
- Aplicar el algoritmo propuesto en al menos dos bases de datos especializadas en imágenes médicas y en dos más para el reconocimiento de patrones y poder evaluar su desempeño.
- La comparación de los resultados se hará con al menos cinco algoritmos de agrupamiento por cada una de las áreas de aplicación mencionadas en esta propuesta.

1.3.2. Limitaciones

- No existe un referente ya que no hay investigaciones previas de los conjuntos difusos titubeantes aplicados a la segmentación de imágenes.
- No se considera la presencia de ruido en la segmentación de imágenes.
- No se considera la presencia de datos atípicos o *outliers* para el Reconocimiento de Patrones.
- Se utilizan imágenes en escala de grises y a color.

1.4. Justificación

La segmentación de imágenes y el reconocimiento de patrones son campos de investigación que han avanzado rápidamente en el ámbito de la inteligencia artificial, siempre buscando el óptimo agrupamiento de píxeles o clasificación de los patrones. La importancia de tener resultados óptimos radica en su aplicación, en el área médica por ejemplo son requeridos para no cometer errores en los diagnósticos o en las operaciones que realizan los médicos en los distintos órganos. Actualmente, existen diferentes métodos para segmentar una imagen o agrupar datos, esta investigación se basa en el *clustering*, específicamente en el agrupamiento difuso. Uno de los algoritmos más utilizado es el *Fuzzy C-Means* que asigna cierto grado de pertenencia a cada elemento con respecto a los grupos definidos, el cual se ha extendido a distintas teorías, tal es el caso de los conjuntos difusos intuitivos que consta de tres partes: grado de pertenencia, grado de no pertenencia y grado de indecisión. También están los conjuntos difusos titubeantes que permiten que la pertenencia tenga un conjunto de valores, estos se han aplicado en la toma de decisiones para obtener la alternativa óptima cuando se tienen múltiples atributos y múltiples personas. La literatura menciona que es un enfoque útil para evitar problemas dado que cada alternativa puede describirse en un conjunto difuso titubeante bajo los términos de los expertos y cada atributo mediante un elemento difuso

titubeante. Al implementar este enfoque, al agrupamiento de datos, éste podría tener mejoras y ser aplicado de manera eficiente al procesamiento de imágenes y al reconocimiento de patrones.

1.4.1. Beneficios e importancia

Los principales beneficios obtenidos con esta investigación son:

- Justificación matemática de un algoritmo difuso titubeante para agrupar datos.
- Mejora de la calidad del agrupamiento al incluir la teoría de los conjuntos difusos titubeantes.

La importancia de la investigación radica en tener un referente en el estado del arte de un algoritmo difuso titubeante para agrupar datos, en este caso aplicado a la segmentación de imágenes médicas y al reconocimiento de patrones, que presente mejores resultados que los algoritmos que existen actualmente.

1.5. Aportaciones científicas

Con la elaboración de esta investigación las principales aportaciones científicas obtenidas son las siguientes:

- Algoritmo de agrupamiento basado en Conjuntos Difusos Titubeantes aplicado al Reconocimiento de Patrones.
- Algoritmo para segmentar imágenes reales y médicas basado en Conjuntos Difusos Titubeantes.
- Bases matemáticas para que esta investigación se extienda hacia otras metodologías basadas en el agrupamiento difuso, por ejemplo, además de incluir posibles valores de pertenencia, incluya varios valores de no pertenencia e indecisión como los *Intuitionistic Fuzzy Sets*.

1.6. Organización de la tesis

Este documento está compuesto por cinco capítulos, además de anexos y referencias. El capítulo 2 contiene el marco conceptual donde se definen conceptos como segmentación de imágenes, reconocimiento de patrones, y tres teorías de los conjuntos difusos que facilitarán la comprensión de esta proyecto de investigación; en este capítulo también se presenta los antecedentes y el estado del arte mencionando aquellos trabajos que fueron tomados como base para esta investigación. El capítulo 3 se refiere a la metodología que se siguió en esta investigación, detallando las principales aportaciones. En el capítulo 4 ilustra la experimentación realizada y los resultados obtenidos. Finalmente, en el capítulo 5 se describen las conclusiones generales de la investigación y se exponen las propuestas para trabajos futuros.

2 | Marco teórico

En este capítulo se proporcionan algunos conceptos importantes para el entendimiento de esta investigación, también se presentan algunos proyectos de tesis que anteceden a este documento. Finalmente, se resumen distintas investigaciones que fueron tomadas como base para este documento de tesis, estas se dividen en cuatro secciones: (1) uso de los conjuntos difusos titubeantes, (2) conjuntos difusos y variantes aplicados a la segmentación de imágenes, (3) otros métodos actuales para la segmentación de imágenes y (4) aplicaciones en el reconocimiento de patrones.

2.1. Marco conceptual

A continuación se describen algunos conceptos que serán de utilidad para el entendimiento de la investigación.

2.1.1. Segmentación de imágenes

La segmentación de imágenes es una tarea fundamental en la visión por computadora y el análisis de imágenes. Su objetivo es delinear con precisión los contornos y las superficies para proporcionar una partición efectiva, rápida y confiable de un objeto. Es una herramienta aplicada en el análisis de imágenes médicas, la detección de objetos, reconocer rostros o voces, los sistemas de control de tráfico, vídeo vigilancia, entre otros (Dougherty, 2011). Un ejemplo de segmentación de imágenes se muestra en la Figura 2.1, siendo una imagen complicada al tener diferentes texturas y cambios de iluminación. Existen diversos métodos de segmentación, que son agrupados en (Jähne, 2005; Snyder and Qi, 2010; Szeliski, 2010; Russ, 2016):

- Basados en píxeles: utiliza únicamente los valores de la intensidad de los píxeles individuales, agrupándolos de acuerdo a sus características sin considerar su relación espacial, los métodos más utilizados están basados en umbrales, histogramas, agrupamiento o *clustering* y redes neuronales.
- Basados en bordes: localizan los bordes de un objeto o región e intenta seguirlos, estos métodos van a proporcionar contornos bien definidos cuando ambos lados del contorno tengan características diferentes. De los métodos más usados es contornos activos o *snake*.
- Basados en regiones: analiza la intensidad en áreas con diferentes tamaños, partiendo de un elemento, píxel o región y va analizando aquellos que estén en su entorno y los agrupa siempre que cumplan alguna condición de homogeneidad. Dentro de los principales métodos están el crecimiento de regiones y los morfológicos.

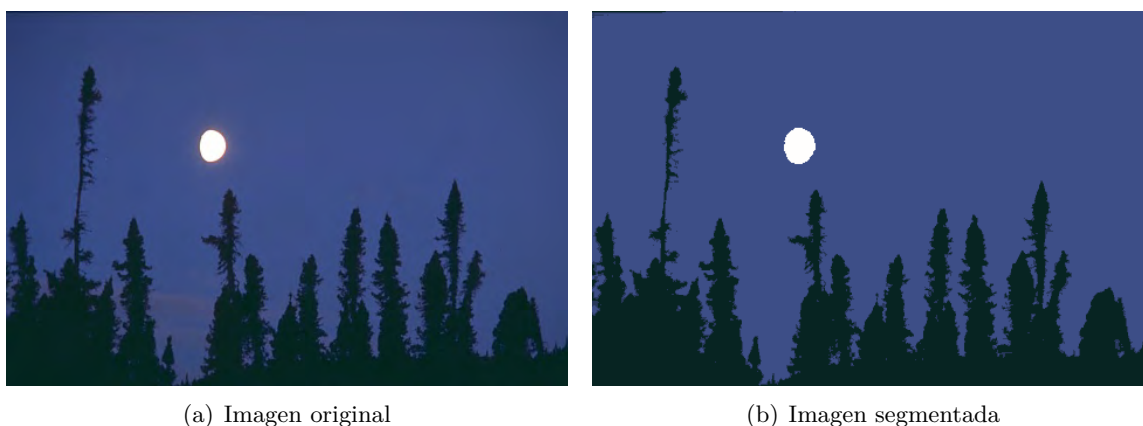


Figura 2.1: Ejemplo de una segmentación en tres regiones.

2.1.2. Reconocimiento de Patrones

El Reconocimiento de Patrones es la ciencia y arte asignar nombres a los objetos naturales del mundo real, considerado parte de la Inteligencia Artificial ya que identifica patrones repetitivos en los datos de entrada (Leondes, 1998). Su objetivo es clasificar los objetos (o patrones) en varias categorías o clases dependiendo de su aplicación, estas pueden ser identificación facial, reconocimiento automático de la voz, reconocimiento de la escritura a mano o cualquier otra medida que se necesite clasificar. A estos objetos se les conoce como patrones (Theodoridis and Koutroumbas, 2009). En el área de Visión Artificial, por ejemplo, el agrupamiento de píxeles en distintos grupos, puede considerarse una aplicación del reconocimiento de patrones.

A veces, cuando se habla de patrones, se refiere a objetos o formas que se pueden percibir, sin embargo no es necesario que sea visible, puede ser un sistema de datos. Un ejemplo claro puede ser el estudio del pronóstico del tiempo, este se basa en patrones especificados en los mapas de contorno de presión y los datos del radar en un área. Por lo cual, un patrón puede definirse como una descripción cuantitativa de un objeto o alguna otra entidad de interés (Bow, 2002).

Un sistema de reconocimiento de patrones clasifica el patrón de entrada en una clase específica y se realiza en dos fase: primero el análisis o la descripción donde se extraen las características del objeto o patrón que se está analizando y segundo, la clasificación que permite reconocer el objeto utilizando características obtenidas de la primer fase (Kpalma and Ronsin, 2007). Existen cuatro metodologías principales que son (Bezdek et al., 1999; Kpalma and Ronsin, 2007):

- Enfoque estadístico: basado en estadísticas y probabilidades. Los patrones a clasificar se representan en un conjunto de características definidas en un vector multidimensional específico, para compararlos se utilizan medidas para observar distancias entre los puntos en el espacio estadístico.
- Enfoque sintáctico: se basa en la relación entre las características. Los patrones se representan por estructuras jerárquicas.
- Comparación de plantillas: Es utilizado en el procesamiento de imágenes, para localizar e identificar partes de la imagen que coinciden con una plantilla o modelo.
- Redes neuronales: es un proceso que aprende a resolver problemas de acuerdo al conocimiento disponible. Simula el funcionamiento del cerebro, además de tener elementos de procesamiento

interconectados simulando las neuronas, a través de la interconexión, cada neurona pasa información a otra.

En la Figura 2.2 se presenta un esquema general de las etapas de un sistema de reconocimiento de patrones basado en el enfoque estadístico; en la visión por computadora, la entrada puede ser una o más imágenes. El primer paso es extraer las características, las cuales se pueden reducir eliminando aquellas que no aporten información relevante para evitar errores en el proceso de clasificación. Finalmente, la clasificación asigna cada uno de los datos a los grupos o clases, de acuerdo a las características extraídas. Comúnmente se utiliza el *clustering* o agrupamiento, el método más representativo es *Fuzzy C-Means*. En esta etapa se utiliza el aprendizaje automático, su objetivo es el desarrollo de los métodos que permiten a las computadoras aprender.

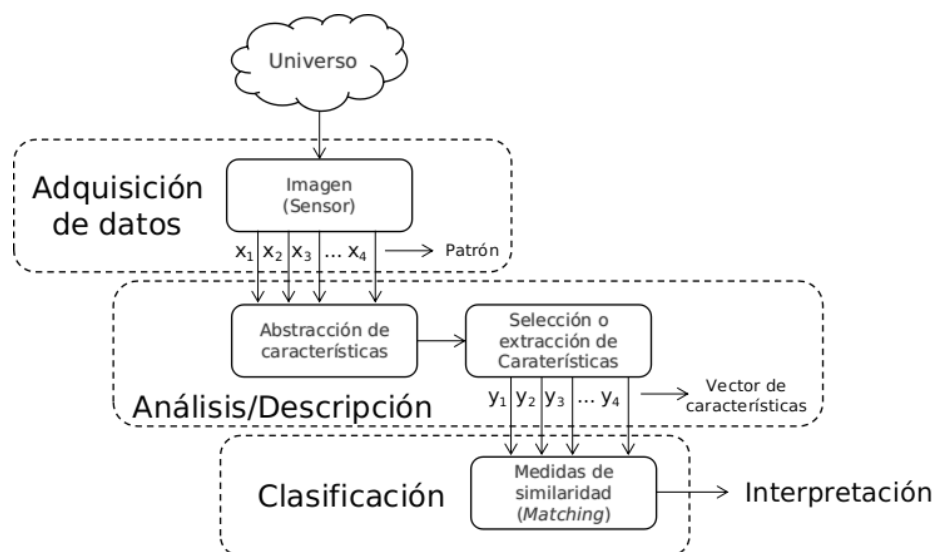


Figura 2.2: Etapas del sistema de reconocimiento de patrones (Bishop, 2006).

2.1.3. Agrupamiento

El agrupamiento de datos, también conocido como *clustering*, análisis de clústers o clasificación no supervisada, es una técnica para generar grupos de objetos o también llamados clústers, de tal forma que los objetos de un mismo grupo compartan características muy similares y sean diferentes a los objetos de otros grupos. Su función puede considerarse como un problema para agrupar objetos de acuerdo a su similitud. Algunas de sus aplicaciones se pueden observar en la minería de textos, procesamiento de imágenes, expresiones genéticas, segmentación de imágenes, entre otras (Gan et al., 2007; Wierzchoń and Kłopotek, 2018).

Matemáticamente, en el agrupamiento se supone que se tiene un conjunto de n datos $X = \{x_1, x_2, \dots, x_n\}$ donde cada dato está representado como un vector $x_i = (x_{i1}, x_{i2}, \dots, x_{im})^T$, de dimensión m , x_{ij} denota la j -ésima característica del dato $x_i \in \mathbb{R}^m$ es llamado en algunos casos vector de características o imagen (Pătraşcu, 2007). La teoría de agrupamiento clásico divide el conjunto de datos en $k < m$ grupos $c = \{c_1, c_2, \dots, c_k\}$ donde cada i -ésimo grupo C_i es llamado clúster; esta división está restringida por:

1. Cada clúster debe contener al menos un objeto, $C_j \neq \emptyset, j = 1, \dots, k$.
2. Cada dato debe pertenecer a un grupo determinado, $\bigcup_{j=1}^k C_j = X$.
3. Cada dato debe pertenecer exactamente a un clúster, $C_{j_1} \cap C_{j_2} = \emptyset, j_1 \neq j_2$.

Un ejemplo puede visualizarse en la Figura 2.3; en la que se presenta un conjunto de datos generado aleatoriamente que es agrupado en 3 clústers. El agrupamiento clásico, esta restringido a que un dato no puede pertenecer a más de dos grupos de forma simultanea, es por eso que se extendió al agrupamiento difuso.

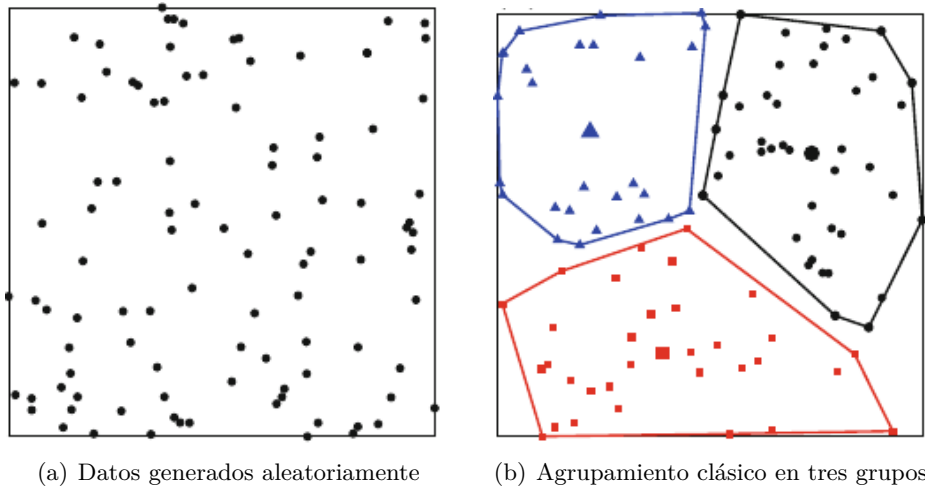


Figura 2.3: Agrupamiento de datos generados aleatoriamente (Wierzchoń and Kłopotek, 2018).

2.1.4. *Fuzzy C-Means*

El agrupamiento difuso, a diferencia del clásico, permite que cada dato tenga una pertenencia gradual en $\mu \in [0, 1]$ con respecto a los grupos o regiones, de tal forma que un dato puede pertenecer a más de un grupo o región simultáneamente. Este concepto, grado de pertenencia, se basa en la definición e interpretación de los conjuntos difusos. De acuerdo con la teoría de los conjuntos difusos, la asignación de los grupos u_{ij} es el grado de pertenencia de un dato x_i al grupo Γ_j , de forma que: $u_{ij} = \mu_{\Gamma_j}(x_i) \in [0, 1]$. En el agrupamiento difuso, no existe una única etiqueta que indique el grupo al que pertenece un dato, sino que se asocian a un vector de etiquetas difusas donde a dato x_i se le establece su grado de pertenencia hacia cada grupo c (Bezdek et al., 1999; De Oliveira and Pedrycz, 2007):

$$\mu_i = (\mu_{i1}, \dots, \mu_{ic})^T \quad (2.1)$$

La matriz $U = (\mu_{ij}) = (\mu_i, \dots, \mu_n)$ de tamaño $c \times n$ es llamada matriz de partición difusa. Entonces, dado un conjunto de datos $X = \{x_1, \dots, x_n\}$ y c el número de grupos $1 < c < n$ representando los conjuntos difusos $\mu_{\Gamma_j}, (j = 1, \dots, c)$ llamaremos a $U_f = (\mu_{ij}) = (\mu_{\Gamma_j}(x_i))$ una partición difusa U_f de X sí (Bezdek et al., 1999; De Oliveira and Pedrycz, 2007):

$$\sum_{i=1}^n \mu_{ij} > 0, \forall j \in \{1, \dots, c\}, \quad (2.2)$$

$$\sum_{j=1}^c \mu_{ij} = 1, \forall i \in \{1, \dots, n\}, \quad (2.3)$$

donde la restricción (2.2) garantiza que ningún grupo estará vacío y la restricción (2.3) asegura que para cada dato, la suma de todos sus grados de pertenencia será igual a 1 (De Oliveira and Pedrycz, 2007). Con esto, se asegura que todos los datos reciben el mismo peso además de ser igualmente incluidos en el particionamiento; por último con estas restricciones ningún grupo puede contener el total de pertenencia de todos los datos.

Para realizar una agrupación difusa, el primer paso es definir una función objetivo, para esto, se dice que entre más cerca este un dato al centro de un grupo su grado de pertenencia debe ser más alto con respecto a ese grupo, entonces, las distancias entre los centros de los grupos y los datos asignados a estos deben ser mínimas. Por lo tanto el problema de dividir un conjunto de datos en c grupos es una tarea para minimizar las distancias cuadradas entre los datos y los centros de los grupos; también se pueden maximizar estos grados de pertenencia. La función objetivo J se basa en la suma de las distancias mínimas cuadradas. Formalmente, para que un modelo de agrupamiento difuso sea óptimo, dado un conjunto de datos X en c grupos se debe minimizar la siguiente función objetivo (Bezdek et al., 1999):

$$J(X, U, V) = \sum_{i=1}^n \sum_{j=1}^c \mu_{ij}^m \|x_i - v_j\|^2 \quad (2.4)$$

considerando que siempre se deben de satisfacer las restricciones (2.2) y (2.3). En la expresión 2.4, de la función objetivo, el parámetro m es llamado fuzzificador o exponente de ponderación y determina cuan difuso es el agrupamiento, además, su valor debe ser $m > 1$. Usualmente, según la literatura, un valor de $m = 2$ es seleccionado debido a que permite obtener mejores resultados. Para minimizar la función objetivo J se deben seguir dos pasos: 1) los grados de pertenencia se optimizan fijando los parámetros de los grupos (ecuación (2.5)) y 2) los centros de los grupos son optimizados fijando los grados de pertenencia (ecuación (2.6)) (Bezdek et al., 1999).

$$U_\tau = j_U (V_{\tau-1}), \tau > 0, \quad (2.5)$$

$$V_\tau = j_V (U_\tau), \quad (2.6)$$

En estos pasos, los valores óptimos son calculados usando las ecuaciones de actualización de los parámetros j_U y j_V para los grados de pertenencia y los centros de los grupos respectivamente. Estas fórmulas de actualización son obtenidas simplemente poniendo a cero la derivada parcial de la función objetivo J con respecto al parámetro a optimizar. Las ecuaciones resultantes de los dos pasos iterativos forman el algoritmo de agrupamiento Fuzzy C-Means. Los grados de pertenencia se obtienen con la siguiente fórmula de actualización:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - v_j\|^2}{\|x_i - v_k\|^2} \right)^{\frac{2}{m-1}}} \quad (2.7)$$

Esta expresión muestra que el grado de pertenencia difuso no solo depende de la distancia del dato x_i al grupo j , sino también de las distancias entre el dato y los demás grupos. La fórmula

de actualización de j_V para los parámetros de los grupos depende de los parámetros usados para describir a un grupo y sobre la medida de distancia seleccionada. En el modelo FCM básico los vectores de los centros de los grupos sirven como prototipos. Consecuentemente la derivación de J con respecto a los centros produce

$$v_j = \frac{\sum_{i=1}^n \mu_{ij}^m x_i}{\sum_{i=1}^n \mu_{ij}^m}, \quad (2.8)$$

La selección de los centros óptimos de los grupos para las pertenencia fijas de los datos a los grupos tiene la forma del cálculo del valor medio generalizado, por lo cual el algoritmo de agrupamiento *Fuzzy C-Means* toma ese nombre (De Oliveira and Pedrycz, 2007). Un ejemplo gráfico de una función de pertenencia μ se muestra en la Figura 2.4.

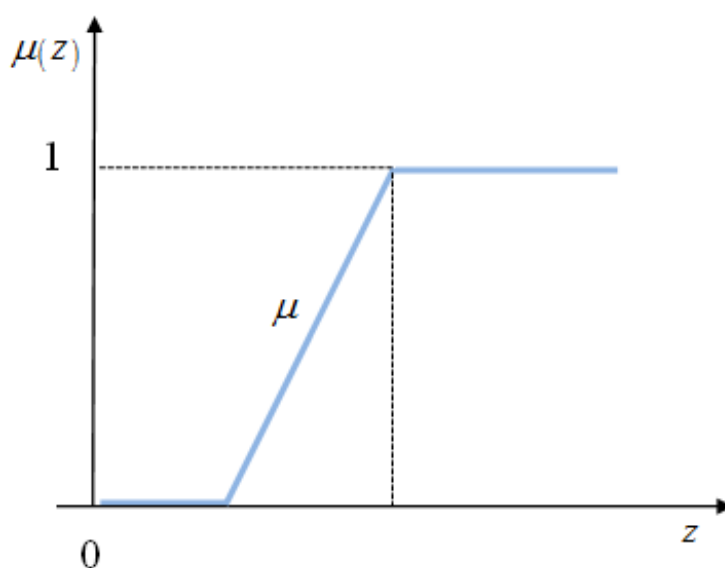


Figura 2.4: Representación gráfica de una función de pertenencia (Ngan et al., 2019).

2.1.5. *Intuitionistic Fuzzy Sets*

Los grados de pertenencia en un conjunto difuso se expresan con valores precisos extraídos del intervalo $[0, 1]$, y es difícil capturar la capacidad humana para expresar grados de pertenencia imprecisos y/o vagos. Por ejemplo, en la toma de decisiones puede surgir imprecisión debido a la información incuantificable, incompleta o no obtenible; a menudo, para un tomador de decisiones (o un experto) le es difícil determinar un número tan claro. Por otro lado, si un grupo de tomadores de decisiones (o expertos) evalúan las alternativas candidatas pueden haber algunos desacuerdos entre ellos, debido a las diferentes opiniones que tiene cada uno, haciendo difícil obtener un resultado de consenso, lo que puede obtenerse es un conjunto de valores posibles. En tal caso, el conjunto difuso clásico no se puede usar para describir opiniones de un grupo. Es por eso que la teoría de conjuntos difusos clásica se ha extendido en varias formas diferentes, siempre basándose en el mismo razonamiento: la asignación del grado de pertenencia de un elemento a un conjunto fijo no es clara (Liao and Xu, 2017).

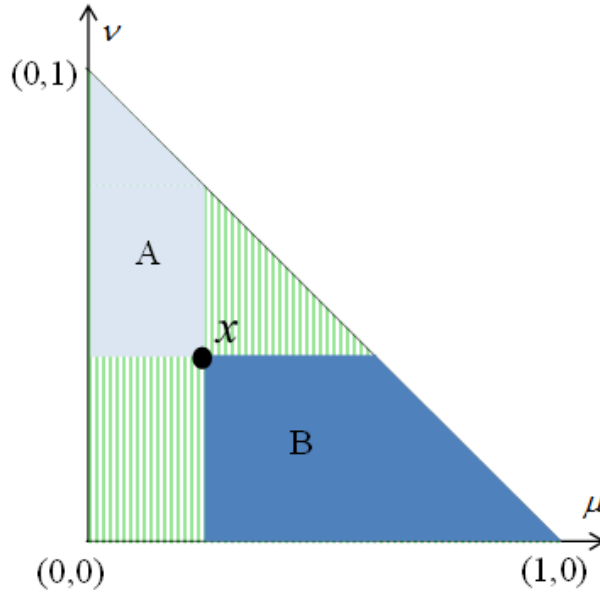


Figura 2.5: Representación gráfica de los conjuntos difusos intuitivos (Ngan et al., 2019).

Entonces, los conjuntos difusos intuitivos (IFS, *Intuitionistic Fuzzy Sets*) asignan a cada elemento un grado de pertenencia $\mu(x)$, además, un grado de no pertenencia $\nu(x)$ y un grado de indecisión $\pi(x)$, permitiéndole ser un método más poderoso que el conjunto difuso para tratar la incertidumbre. Un ejemplo del dominio de los valores difusos intuitivos se muestra en la Figura 2.5. Dado X ser el universo de datos, dado $I \subset X$ ser un conjunto fijo. Entonces un IFS I en X es expresado de la siguiente forma (Liao and Xu, 2017; Atannasov, 1999):

$$I = \{\langle x, \mu_I(x), \nu_I(x) \rangle \mid x \in X\} \quad (2.9)$$

donde las funciones $\mu_I : I \mapsto [0, 1]$ y $\nu_I : I \mapsto [0, 1]$ definen los grados de pertenencia y no pertenencia del elemento $x \in X$ al conjunto I , respectivamente, y para cada $x \in X$

$$0 \leq \mu_I + \nu_I \leq 1 \quad (2.10)$$

obviamente, cada conjunto difuso intuitivo es de la siguiente forma:

$$I = \{\langle x, \mu_I(x), \nu_I(x) \rangle \mid x \in X\} \quad (2.11)$$

En forma generalizada, el tercer parámetro corresponde al grado de indecisión $\pi_I(x)$, que se incluye en la expresión (1.11), y es definido por:

$$I = \{\langle x, \mu_I(x), \nu_I(x), \pi_I(x) \rangle \mid x \in X\} \quad (2.12)$$

donde el grado de indecisión puede ser expresado como $\pi_I = 1 - \mu_I(x) - \nu_I(x)$. Un caso específico es aquel en el que $\pi_I = 0$, implica que el conjunto difuso intuitivo se convierte en un conjunto difuso ordinario (Atannasov, 1999). Con esta teoría se formuló el algoritmo de agrupamiento *Intuitionistic*

Fuzzy C-Means, (Xu and Wu, 2010) cuya función objetivo es:

$$J(U; V; X) = \sum_{i=1}^N \sum_{j=1}^c u_{ij}^m d^2(x_i, v_j) \quad (2.13)$$

donde $X = \{x_1, x_2, \dots, x_n\}$ son N elementos difusos intuitivos, cada x_i esta representado como conjuntos difusos intuitivos de la forma $x_i = \{(\mu(x_i)), \nu(x_i), \pi(x_i)\}$, m es el factor difuso ($m > 1$), c el número de grupos ($1 < c < N$), $V = \{v_1, v_2, \dots, v_c\}$ son los centros de los grupos, cada v_j se representa en términos de los conjuntos difusos intuitivos como $v_j = \{(\mu(v_j)), \nu(v_j), \pi(v_j)\}$. u_{ij} es el grado de pertenencia del i -ésimo elemento con respecto al j -ésimo grupo y $U = (u_{ij})_{c \times N}$ es la matriz de pertenencia. $d^2(x_i, v_j)$ es la medida de distancia entre datos y centroides, y esta definida por:

$$d^2(x_i, v_j) = ((\mu(x_i) - \mu(v_j))^2 + (\nu(x_i) - \nu(v_j))^2 + (\pi(x_i) - \pi(v_j))^2) \quad (2.14)$$

La ecuación para actualizar los grados de pertenencia u_{ij} se calcula con:

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{d^2(x_i, v_j)}{d^2(x_i, v_k)} \right)^{\frac{2}{m-1}}}, 1 \leq j \leq c; 1 \leq i \leq N \quad (2.15)$$

Y los centroides $v_j = \{(\mu(v_j)), \nu(v_j), \pi(v_j)\}$ se calculan a partir de:

$$\mu(v_j) = \frac{\sum_{i=1}^N u_{ij}^m \mu(x_i)}{\sum_{i=1}^N u_{ij}^m}, 1 \leq j \leq c \quad (2.16)$$

$$\nu(v_j) = \frac{\sum_{i=1}^N u_{ij}^m \nu(x_i)}{\sum_{i=1}^N u_{ij}^m}, 1 \leq j \leq c \quad (2.17)$$

$$\pi(v_j) = \frac{\sum_{i=1}^N u_{ij}^m \pi(x_i)}{\sum_{i=1}^N u_{ij}^m}, 1 \leq j \leq c \quad (2.18)$$

2.1.6. *Hesitant Fuzzy Sets*

Los conjuntos difusos titubeantes (HFS, *Hesitant Fuzzy Sets*) se originan a partir de la dificultad que existe para determinar el grado de pertenencia de un elemento con respecto a un grupo, debido a que en ocasiones es difícil obtener un valor preciso o un margen de error, y lo que se obtiene es un conjunto de valores posibles; este tipo de conjuntos son una generalización del conjunto difuso, que describe mejor aquellas situaciones donde las personas tienen dudas al proporcionar sus preferencias sobre los elementos en el proceso de la toma de decisiones. Los HFS fueron propuestos originalmente por Torra (Liao and Xu, 2017; Xu, 2016).

Los conjuntos difusos titubeantes permiten tener varios posibles valores de pertenencia para un solo elemento x de un conjunto de referencia X (Torra, 2010). Dado X ser el universo de datos, un HFS sobre X está en términos de una función h que cuando se aplica a X regresa un subconjunto de $[0, 1]$. Para entenderse con mayor facilidad, los conjuntos difusos titubeantes matemáticamente se representan como (Xu, 2016):

$$Z = \{\langle x, h_Z(x) \rangle | x \in X\} \quad (2.19)$$

donde $h_Z(x)$ es el conjunto de valores posibles en $[0, 1]$ y denota los posibles grados de pertenencia del elemento $x \in X$ al conjunto Z , el parámetro $h = h_Z(x)$ es llamado elemento difuso titubeante (*Hesitant Fuzzy Element*: HFE) y Θ el conjunto de todos los HFEs (De Oliveira and Pedrycz, 2007; Xu, 2016). HFE es considerado la unidad básica de un conjunto difuso titubeante. Describe la situación de manera más completa que un número 0.4 (o 0.6, o 0.7), o un número difuso con valores del intervalo $[0.4, 0.7]$ o el número difuso intuitivo $(0.4, 0.6)$. Ya que el grado en que la alternativa debe satisfacer al atributo no es 0.4 o 0.6, ni el intervalo entre 0.4 y 0.7, sino los tres posibles valores, 0.4, 0.6, 0.7 (Xu, 2016).

Por ejemplo, sea $X = \{x_1, x_2, x_3\}$ un conjunto fijo. $h_Z(x_1) = \{0.2, 0.4, 0.5\}$, $h_Z(x_2) = \{0.3, 0.4\}$ y $h_Z(x_3) = \{0.3, 0.2, 0.5, 0.6\}$ son los elementos difusos titubeantes (HFEs) de $x_i (i = 1, 2, 3)$ del conjunto Z , respectivamente. Entonces Z puede considerarse como un conjunto difuso titubeante:

$$Z = \{\langle x_1, 0.2, 0.4, 0.5 \rangle, \langle x_2, 0.3, 0.4 \rangle, \langle x_3, 0.3, 0.2, 0.5, 0.6 \rangle\}$$

Algunos elementos difusos titubeantes especiales para x en X son (Xu, 2016):

- (1) Conjunto vacío: $h(x) = \{0\}$, para simplificar se denota 0^* .
- (2) Conjunto completo: $h(x) = \{1\}$ se denota I^* .
- (3) Ignorancia completa (todos son posibles): $h(x) = [0, 1]$, se denota U^* .
- (4) Conjunto sin sentido: $h(x) = \{\emptyset^*\}$.

En otras palabras se dice que sí h no devuelve ningún valor, se afirma que es un conjunto sin sentido. Si regresa el conjunto $[0, 1]$, significa que todos los valores entre 0 y 1 son posibles, se le llama ignorancia completa. Si devuelve un sólo valor, tiene sentido ya que el único valor también se puede ver como un subconjunto de $[0, 1]$. Si el valor devuelto es 0, quiere decir que el grado de pertenencia es cero, por lo tanto es un conjunto vacío. Si el valor es 1, es un conjunto completo. En la toma de decisiones, un conjunto vacío muestra que todos se oponen a la alternativa. Un conjunto completo que todos están de acuerdo con ella. Mientras que la ignorancia completa se refiere a que no tienen idea de la alternativa (Xu, 2016).

Un ejemplo gráfico para representar la función difusa h de este tipo de conjuntos se observa en la Figura 2.6 donde los diferentes valores de pertenencia expresan la evaluación de cada uno de los expertos, los cuales se pueden englobar en una nueva función h , en el ejemplo es de forma trapezoidal, pero podría tener cualquier otra forma, dependiendo de la evaluación de los expertos.

Para los conjuntos difusos titubeantes existen distintas operaciones que pueden definirse a partir de dos o mas HFEs, tal es el caso de unión e intersección (Xu, 2016):

$$(1) g \cup h = \cup_{\lambda_1 \in g, \lambda_2 \in h} \max\{\lambda_1, \lambda_2\}$$

$$(2) g \cap h = \cup_{\lambda_1 \in g, \lambda_2 \in h} \min\{\lambda_1, \lambda_2\}$$

donde g y h son dos elementos difusos titubeantes. Otras operaciones básicas se describen a continuación, en donde $\lambda > 0$:

$$(1) g^\lambda = \cup_{\lambda \in g} \{\gamma^\lambda\}$$

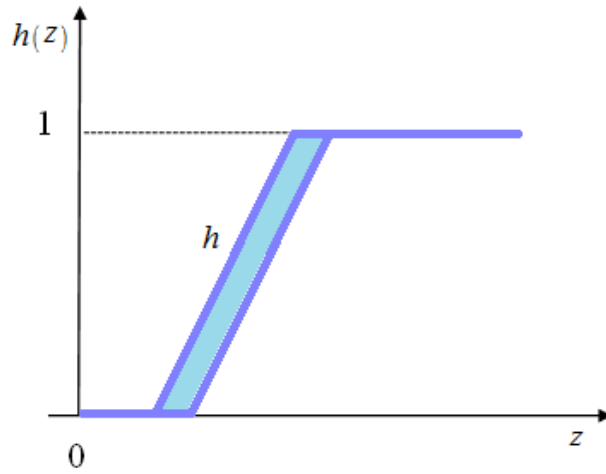


Figura 2.6: Representación gráfica de conjuntos difusos titubeantes (Ngan et al., 2019).

$$(2) \lambda g = \bigcup_{\lambda \in g} \{1 - (1 - \gamma^\lambda)\}$$

$$(3) g \oplus h = \bigcup_{\lambda_1 \in g, \lambda_2 \in h} \{\gamma_1 + \gamma_2 - \gamma_1 \gamma_2\}$$

$$(4) g \otimes h = \bigcup_{\lambda_1 \in g, \lambda_2 \in h} \{\gamma_1 \gamma_2\}$$

$$(5) g \ominus h = \bigcup_{\lambda_1 \in g, \lambda_2 \in h} \{\bar{\gamma}\}, \text{ donde } \bar{\gamma} = \begin{cases} \frac{\gamma_1 - \gamma_2}{1 - \gamma_2}, & \text{si } \gamma_1 \geq \gamma_2 \text{ y } \gamma_2 \neq 1 \\ 0, & \text{cualquier otro caso} \end{cases}$$

$$(6) g \oslash h = \bigcup_{\lambda_1 \in g, \lambda_2 \in h} \{\bar{\gamma}\}, \text{ donde } \bar{\gamma} = \begin{cases} \frac{\gamma_1}{\gamma_2}, & \text{si } \gamma_1 \leq \gamma_2 \text{ y } \gamma_2 \neq 0 \\ 1, & \text{cualquier otro caso} \end{cases}$$

Otro de los conceptos importantes dentro de los conjuntos difusos titubeantes son las medidas de similitud y distancia que miden la diferencia existente entre dos conjuntos difusos titubeantes, estas son definidas como (Xu, 2016):

1. Sea Z_1 y Z_2 dos conjuntos difusos titubeantes en X , la medida de distancia entre Z_1 y Z_2 esta definida como $d(Z_1, Z_2)$ y satisface las siguientes propiedades:

- (1) $0 \leq d(Z_1, Z_2) \leq 1$
- (2) $d(Z_1, Z_2) = 0$ si y solo si $Z_1 = Z_2$
- (3) $d(Z_1, Z_2) = d(Z_2, Z_1)$

2. Sea Z_1 y Z_2 dos conjuntos difusos titubeantes en X , la medida de similitud entre Z_1 y Z_2 esta definida como $\bar{s}(Z_1, Z_2)$ y satisface las siguientes propiedades:

- (1) $0 \leq \bar{s}(Z_1, Z_2) \leq 1$
- (2) $\bar{s}(Z_1, Z_2) = 1$ si y solo si $Z_1 = Z_2$

$$(3) \bar{s}(Z_1, Z_2) = \bar{s}(Z_2, Z_1)$$

En muchos casos el número de valores de dos elementos difusos titubeantes puede ser diferente $l_{Z_1(x_i)} \neq l_{Z_2(x_i)}$, siendo $l_{Z_1(x_i)}$ y $l_{Z_2(x_i)}$ el número de valores en $Z_1(x_i)$ y $Z_2(x_i)$, respectivamente. Comúnmente se toma $l_{x_i} = \max\{l_{Z_1(x_i)}, l_{Z_2(x_i)}\}$ para cada x_i en X (Xu and Xia, 2011). Por ejemplo:

Sea $h_{Z_1}(x_i) = \{0.1, 0.2, 0.3\}$, $h_{Z_2}(x_i) = \{0.4, 0.5\}$, y $l_{Z_1(x_i)} > l_{Z_2(x_i)}$. Para operar correctamente se debe extender $h_{Z_2}(x_i)$ hasta que tenga la misma longitud que $h_{Z_1}(x_i)$, pudiéndose realizar desde dos perspectivas diferentes, los optimistas, que añaden el máximo valor están anticipando resultados deseables, mientras que los pesimistas esperan resultados desfavorables al añadir el valor menor.

$$(1) \text{ Optimista: } h_{Z_2}(x_i) = \{0.4, 0.5, 0.5\}$$

$$(2) \text{ Pesimista: } h_{Z_2}(x_i) = \{0.4, 0.4, 0.5\}$$

2.2. Estado del Arte

En esta sección se presentan investigaciones relevantes relacionadas con los conjuntos difusos titubeantes, entre las que se incluye la base teórica de este tipo de conjuntos, sus medidas de distancia y similitud, así como algunas propuestas actuales de segmentación de imágenes y reconocimiento de patrones.

2.2.1. Antecedentes

Algunas de las investigaciones que anteceden a la actual tesis doctoral se describen a continuación:

- **Extracción de Superpíxeles y Fusión Local Mediante un Algoritmo de Agrupamiento Difuso Intuicionista (Domínguez, 2019):** El objetivo de este trabajo de investigación fue adaptar el algoritmo de Agrupamiento Difuso Intuitivo (*Intuitionistic Fuzzy C-Means*, IFCM) para extraer superpíxeles eficientemente en imágenes de resonancia magnética del cerebro. El algoritmo que se propuso en la investigación se desempeñó correctamente en las pruebas realizadas con los conjuntos de datos de *BrainWeb* y *BSDS500*, obteniendo resultados favorables en la mayoría de las métricas de recuperación de límites. Además, el algoritmo fue capaz de generar superpíxeles con alta adherencia a los bordes, permitiéndole preservar los detalles pequeños en las imágenes.
- **Sobresgmentación de Imágenes a Color Mediante un Grafo Difuso Basado en una Medida de Distancia Espacial y Color (Luna, 2021):** En esta investigación describen un algoritmo de sobresgmentación de imágenes a color a partir de un grafo difuso que se basa en una medida de distancia espacial y color, su principal ventaja es aprovechamiento de la información de los vecinos para generar el grafo difuso que relaciona píxeles. El algoritmo fue comparado con otros algoritmos del estado del arte, en 200 imágenes seleccionadas de la base de datos *BSDS500*, evaluando la sobresgmentación con tres métricas. Los resultados experimentales demostraron la efectividad del algoritmo superando a los algoritmos con los que se comparó, siendo eficiente con pocos superpíxeles, además de tener una rápida adherencia a los bordes.

2.2.2. Trabajos relacionados

Los trabajos relacionados con el tema de esta investigación se han dividido en cuatro secciones, que describen en donde se han utilizado los conjuntos difusos titubeantes, algunos algoritmos basados en los conjuntos difusos y variantes aplicados a la segmentación de imágenes, así como otros métodos actuales.

Uso de los conjuntos difusos titubeantes

Los conjuntos difusos titubeantes fueron introducidos a la literatura por Torra (2010), con el objetivo de definir la pertenencia de un elemento, debido a que se tiene un conjunto de valores posibles como es el caso de la toma de decisiones, donde cada experto asigna un valor distinto a un elemento, de acuerdo a su opinión o experiencias, entonces es útil tratar con todos los valores posibles. Existen muchas investigaciones que abordan este tema, todas enfocadas principalmente a la toma de decisiones, algunas proponen operaciones y operadores de agregación, (Xia and Xu, 2011) coeficientes de correlación (Chen et al., 2013) y medidas de distancia (Zeng et al., 2016; Junhua Hu and Chen, 2017) entre los elementos difusos titubeantes, y algunas ofrecen aplicaciones o algoritmos para resolver problemas de toma de decisiones; en (Aliahmadipour et al., 2017) se abordan algunos algoritmos de agrupamiento jerárquico con estos conjuntos difusos titubeantes, para demostrar que pueden aplicarse en diferentes contextos del agrupamiento. Otras aplicaciones de los conjuntos difusos titubeantes de desarrolla en (Wang et al., 2018b) que presenta un sistema de planificación de rutas mediante el algoritmo A* en el que se incorpora la teoría de los conjuntos difusos vacilantes (Figura 2.7).

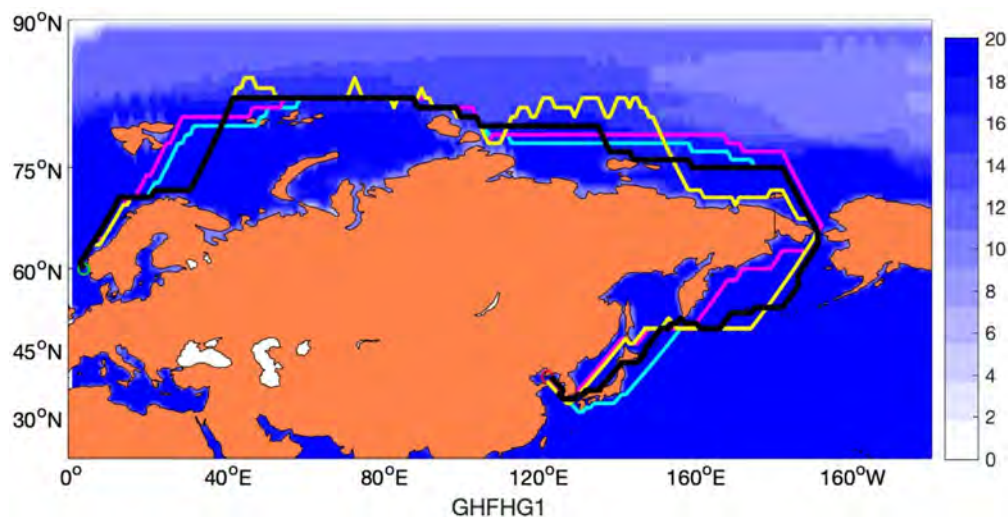


Figura 2.7: Optimización de rutas por el algoritmo HFS-A* (Wang et al., 2018b).

A este concepto también se le han agregado algunas características para hacer su uso más adecuado, como permitir distintos grados de pertenencias ponderados o sus proporciones asociadas (Farhadinia, 2017; Xiong et al., 2018), agregar un vector de características de información difusa titubeante (Lv et al., 2019), utilizar la teoría de grafos en circunstancias difusas titubeantes (Naz and Akram, 2019), entre otras. La mayoría de estas investigaciones dan explicaciones detalladas de estos conjuntos además de exponer los fundamentos necesarios para entenderlos.

Un ejemplo de como se ven este tipo de conjuntos se observa en la Figura 2.8, presentado en Farhadinia (2017) que ilustró la efectividad de sus conceptos a través de un problema de diagnóstico médico, en donde tres expertos médicos proporcionaron la posible evaluación de los diagnósticos con respecto a los síntomas (Figura 2.8a), además evaluaron y dieron los síntomas característicos de los pacientes considerados (Figura 2.8b) y finalmente encontrar el grado de dependencia entre los diagnósticos y los pacientes (Figura 2.8c).

	Temperature	Headache	Cough	Stomach pain	Chest pain
Viral fever	$\left\{0.5, \frac{2}{3}\right\}, \left\{0.7, \frac{1}{3}\right\}$	$\left\{0.3, \frac{3}{3}\right\}$	$\left\{0.4, \frac{1}{3}\right\}, \left\{0.6, \frac{2}{3}\right\}$	$\left\{0.3, \frac{2}{3}\right\}, \left\{0.4, \frac{1}{3}\right\}$	$\left\{0.6, \frac{2}{3}\right\}, \left\{0.7, \frac{1}{3}\right\}$
Malaria	$\left\{0.4, \frac{1}{3}\right\}, \left\{0.7, \frac{2}{3}\right\}$	$\left\{0.5, \frac{2}{3}\right\}, \left\{0.8, \frac{1}{3}\right\}$	$\left\{0.5, \frac{1}{3}\right\}$	$\left\{0.2, \frac{2}{3}\right\}, \left\{0.3, \frac{1}{3}\right\}$	$\left\{0.7, \frac{1}{3}\right\}, \left\{0.8, \frac{2}{3}\right\}$
Typhoid	$\left\{0.3, \frac{2}{3}\right\}, \left\{0.4, \frac{1}{3}\right\}$	$\left\{0.5, \frac{1}{3}\right\}, \left\{0.6, \frac{2}{3}\right\}$	$\left\{0.1, \frac{2}{3}\right\}, \left\{0.4, \frac{1}{3}\right\}$	$\left\{0.6, \frac{3}{3}\right\}$	$\left\{0.5, \frac{2}{3}\right\}, \left\{0.7, \frac{1}{3}\right\}$
Stomach problem	$\left\{0.5, \frac{2}{3}\right\}, \left\{0.7, \frac{1}{3}\right\}$	$\left\{0.5, \frac{2}{3}\right\}, \left\{0.7, \frac{1}{3}\right\}$	$\left\{0.7, \frac{1}{3}\right\}$	$\left\{0.4, \frac{1}{3}\right\}, \left\{0.5, \frac{1}{3}\right\}$	$\left\{0.5, \frac{1}{3}\right\}, \left\{0.8, \frac{2}{3}\right\}$
Chest problem	$\left\{0.2, \frac{2}{3}\right\}, \left\{0.3, \frac{1}{3}\right\}$	$\left\{0.4, \frac{2}{3}\right\}, \left\{0.7, \frac{1}{3}\right\}$	$\left\{0.4, \frac{1}{3}\right\}, \left\{0.6, \frac{2}{3}\right\}$	$\left\{0.3, \frac{2}{3}\right\}, \left\{0.4, \frac{1}{3}\right\}$	$\left\{0.4, \frac{3}{3}\right\}$

(a) Síntomas característicos para cada diagnóstico

	Temperature	Headache	Cough	Stomach pain	Chest pain
Al	$\left\{0.4, \frac{3}{3}\right\}$	$\left\{0.5, \frac{2}{3}\right\}, \left\{0.7, \frac{1}{3}\right\}$	$\left\{0.6, \frac{2}{3}\right\}, \left\{0.7, \frac{2}{3}\right\}$	$\left\{0.2, \frac{2}{3}\right\}, \left\{0.4, \frac{1}{3}\right\}$	$\left\{0.1, \frac{1}{3}\right\}, \left\{0.2, \frac{2}{3}\right\}$
Bob	$\left\{0.6, \frac{2}{3}\right\}, \left\{0.7, \frac{1}{3}\right\}$	$\left\{0.5, \frac{1}{3}\right\}, \left\{0.8, \frac{2}{3}\right\}$	$\left\{0.5, \frac{2}{3}\right\}, \left\{0.6, \frac{1}{3}\right\}$	$\left\{0.3, \frac{3}{3}\right\}$	$\left\{0.4, \frac{2}{3}\right\}, \left\{0.5, \frac{1}{3}\right\}$
Joe	$\left\{0.2, \frac{2}{3}\right\}, \left\{0.3, \frac{1}{3}\right\}$	$\left\{0.5, \frac{3}{3}\right\}$	$\left\{0.2, \frac{2}{3}\right\}, \left\{0.4, \frac{1}{3}\right\}$	$\left\{0.6, \frac{1}{3}\right\}, \left\{0.7, \frac{2}{3}\right\}$	$\left\{0.5, \frac{2}{3}\right\}, \left\{0.7, \frac{1}{3}\right\}$
Ted	$\left\{0.4, \frac{3}{3}\right\}$	$\left\{0.4, \frac{2}{3}\right\}, \left\{0.7, \frac{1}{3}\right\}$	$\left\{0.3, \frac{1}{3}\right\}, \left\{0.4, \frac{2}{3}\right\}$	$\left\{0.7, \frac{2}{3}\right\}, \left\{0.8, \frac{1}{3}\right\}$	$\left\{0.5, \frac{1}{3}\right\}, \left\{0.6, \frac{2}{3}\right\}$

(b) Síntomas característicos de cada paciente

	Viral fever	Malaria	Typhoid	Stomach problem	Chest Problem
Al	0.7984	0.5376	0.5998	0.6467	0.7906
Bob	0.7261	0.7581	0.8549	0.7881	0.7781
Joe	0.8030	0.6568	0.5697	0.6706	0.7356
Ted	0.8513	0.5778	0.8723	0.6798	0.8433

(c) Grado de dependencia entre diagnósticos y pacientes.

Figura 2.8: Ejemplo de la aplicación de los conjuntos difusos titubeantes en la toma de decisiones.

Parte importante de la teoría de los conjuntos difusos titubeantes son las distintas medidas de similitud y distancia (Li et al., 2015; Zeng et al., 2016). Dentro de las distancias más usadas para estos conjuntos se encuentran las definidas en Xu and Xia (2011), distancia Hamming normalizada titubeante:

$$d_h(Z_1, Z_2) = \frac{1}{n} \sum_{i=1}^n \left[\frac{1}{l_{x_i}} \sum_{j=1}^{l_{x_i}} \left| h_{Z_1}^{\sigma^j}(x_i) - h_{Z_2}^{\sigma^j}(x_i) \right| \right] \quad (2.20)$$

la distancia Euclidiana normalizada titubeante:

$$d_e(Z_1, Z_2) = \left[\frac{1}{n} \sum_{i=1}^n \left(\frac{1}{l_{x_i}} \sum_{j=1}^{l_{x_i}} \left| h_{Z_1}^{\sigma^j}(x_i) - h_{Z_2}^{\sigma^j}(x_i) \right|^2 \right) \right]^{\frac{1}{2}} \quad (2.21)$$

donde $h_{Z_1}^{\sigma^j}(x_i)$ y $h_{Z_2}^{\sigma^j}(x_i)$ son los j th valores mas grandes en $h_{Z_1}(x_i)$ y $h_{Z_2}(x_i)$, respectivamente. Las ecuaciones 2.20 y 2.21 se extendieron a una distancia normalizada titubeante generalizada:

$$d_g(Z_1, Z_2) = \left[\frac{1}{n} \sum_{i=1}^n \left(\frac{1}{l_{x_i}} \sum_{j=1}^{l_{x_i}} \left| h_{Z_1}^{\sigma^j}(x_i) - h_{Z_2}^{\sigma^j}(x_i) \right|^\lambda \right) \right]^{\frac{1}{\lambda}} \quad (2.22)$$

donde $\lambda > 0$.

En Junhua Hu and Chen (2017) plantearon una huella de límite de intervalo para analizar la variación de los valores en función de un orden específico entre diferentes elementos difusos titubeantes y determinar el grado de similitud, si la variación es pequeña, la similitud es mayor. Por otra parte, en Yang and Hussain (2018) propusieron nuevas medidas basadas en la métrica de *Hausdorff*, su ventaja radica en que no es necesario el ordenamiento del elemento difuso titubeante, ni extender su longitud hacia el más largo.

$$d_h(M, N) = \frac{1}{n} \sum_{i=1}^n \max \left(\max_j \left\{ \min_k \left\| h_M^{\sigma^j}(x_i) - h_N^{\sigma^k}(x_i) \right\| \right\}, \max_k \left\{ \min_j \left\| h_M^{\sigma^j}(x_i) - h_N^{\sigma^k}(x_i) \right\| \right\} \right) \quad (2.23)$$

Otras aplicaciones de este tipo de conjuntos se presenta en (Zhou and Xu, 2018), que utilizan las características de los HFS al combinarlos con los IFS, considerando simultáneamente los grados de pertenencia y no pertenencia y la presentación general del grado de pertenencia como un conjunto de valores posibles, su aplicación es en la toma de decisiones, específicamente en la la selección de inversiones en campos financieros.

(Li et al., 2018) proponen el algoritmo *Hesitant Fuzzy Kernel C-Means* (HFKCM) que utiliza funciones de kernel, mapeando los datos del espacio muestral a un espacio con una alta dimensión, dando como resultado una agrupación más precisa aplicados en una simulación experimental sobre distribuciones de instalaciones y la Ruta Marítima de la Seda del siglo XXI. Sin embargo, mencionaron que el algoritmo tiene una alta complejidad computacional y como se manejan grandes cantidades de datos, el tiempo de cálculo puede no ser satisfactorio. Por otra parte Bharati (2018) en su investigación describieron un algoritmo para dar solución a los problemas de programación lineal multiobjetivo, basado en los conjuntos difusos titubeantes, este fue aplicado a la planificación de la producción. Y fue beneficiado con el hecho de que se proporcionaron un conjunto de soluciones basadas en varios niveles de expertos y al tomar las decisiones se puede elegir una solución óptima de acuerdo a la situación real. Finalmente, estos conjuntos difusos titubeantes han sido principalmente utilizados en la toma de decisiones, tal como se mencionó anteriormente, pocas investigaciones mencionan su aplicación al agrupamiento de datos (Sun et al., 2018; Lv et al., 2019).

Conjuntos difusos y variantes aplicados a la segmentación de imágenes

Uno de los algoritmos más utilizados para agrupamiento es *Fuzzy C-Means*, un algoritmo clásico basado en particiones difusas, al cual se le han realizado distintas modificaciones para obtener resultados mejores en la agrupación de sus datos, como agregar la capacidad de capturar información de densidad (Pei et al., 2017; Chen et al., 2018), utilizar el crecimiento de regiones o búsqueda *Cuckoo* (Sadad et al., 2018; Wang et al., 2018a), también ha sido complementado con información espacial (Aljebory and Mohammed, 2018; Zhang et al., 2018b; Zhang and Song, 2019; Hu and

Du, 2019; Wu and Yang, 2020). Este algoritmo se ha aplicado en diferentes áreas como medicina, psicología, economía, biología, entre otras disciplinas; también se convirtió en una herramienta clave para el análisis de imágenes a través del agrupamiento de los objetos de una imagen.

De la misma forma, el algoritmo FRFCM (Lei et al., 2018a), es una mejora al algoritmo FCM que emplea la reconstrucción morfológica para suavizar las imágenes y mejorar la inmunidad al ruido preservando los detalles de la imagen tal como se puede observar en la Figura 2.9, además utiliza un filtrado que modifica la pertenencia de los píxeles sin depender del cálculo de la distancia entre píxeles vecinos y los centros de agrupamiento.



Figura 2.9: Resultados de la segmentación de imágenes del algoritmo FRFCM (Lei et al., 2018a).

El algoritmo DSFCM_N (*Deviation-sparse Fuzzy C-Means with neighbor information constraint*) (Zhang et al., 2018b) depende de las desviaciones entre los valores medidos y los valores teóricos, además también utilizan la información espacial para tener mejores resultados de agrupación al considerar que el uso de valores teóricos de datos es más confiable que simplemente considerar los datos en sí, los resultados de este algoritmo se presentan en la Figura 2.10. Por su parte el algoritmo AKFCMS (*Adaptive kernel-based FCM with spatial constraints*) Hu and Du (2019) fue basado en el agrupamiento clásico *Kernelized Fuzzy C-Means* (KFCM) además de utilizar la información espacial para introducir un coeficiente de ponderación adaptativo y controlar la influencia de los píxeles vecinos a los píxeles centrales y mejorar la calidad de la agrupación.



Figura 2.10: Resultados de la segmentación de imágenes algoritmo DSFCM_N (Zhang et al., 2018b).

Otro método efectivo para la segmentación de imágenes es el agrupamiento basado en superpíxeles (Lei et al., 2018b; Wu et al., 2019). Basándose en esta idea, el algoritmo AFCE utilizó los superpíxeles para mejorar el algoritmo *Density Peak* (DP), considerando también información espacial para obtener una segmentación de manera efectiva (Lei et al., 2019b); en esta investigación se obtuvieron tres diferentes algoritmos de segmentación (SLIC- AFCE, LSC-AFCE y MMGR-AFCE)

que están basados en tres algoritmos de superpíxeles: SLIC, LSC, y MMGR-WT; un ejemplo del resultado de la segmentación se presenta en la Figura 2.11, la primera fila es la imagen original y las otras tres el resultado de la segmentación de cada algoritmo; en estas imágenes se observa la diferencia de los tres algoritmos, en donde MMGR-AFCF tiene un resultado superior que el de los otros dos, esto es debido a que el resultado de superpíxeles generado es mejor que el de SLIC y LSC.

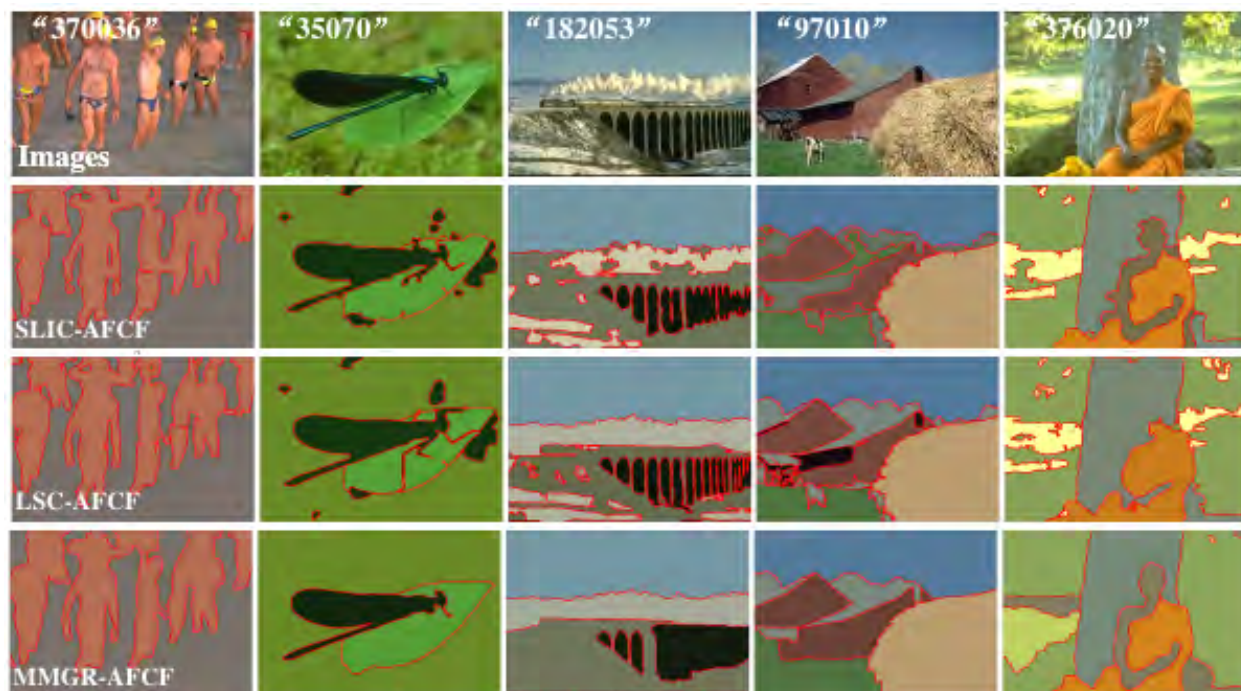
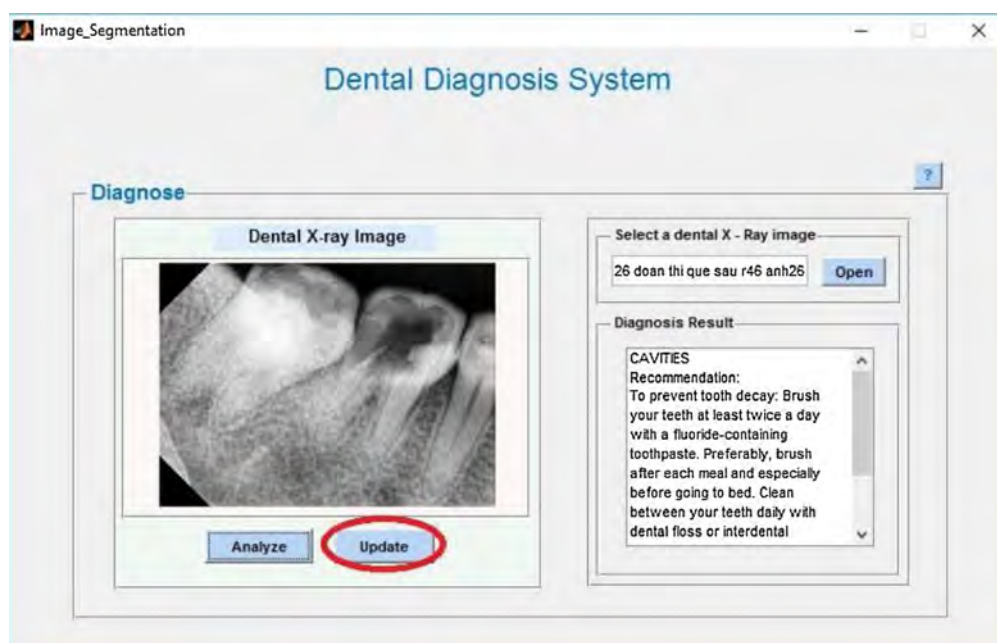
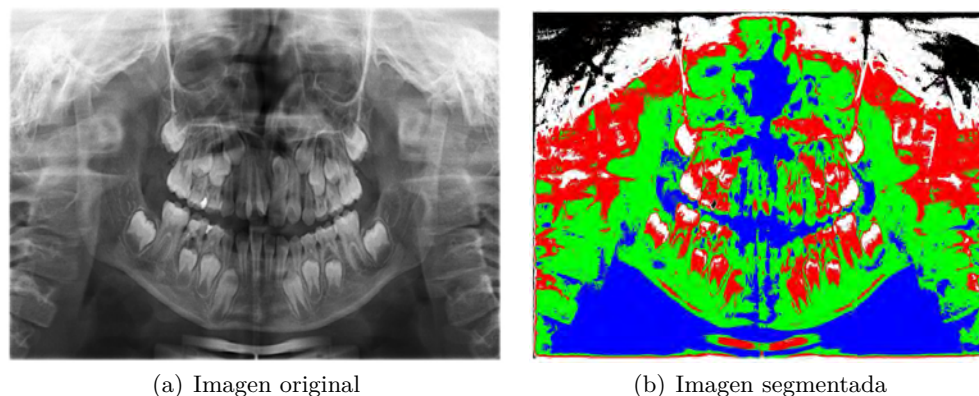


Figura 2.11: Resultados de la segmentación de imágenes algoritmo AFCF (Lei et al., 2019b).

Asimismo, el algoritmo RSSFCA (*Robust self-sparse fuzzy clustering algorithm*) (Jia et al., 2020) también conduce a excelentes resultados de segmentación de imágenes, este algoritmo incorpora el término de regularización en su función objetivo para equilibrar la escasez de pertenencias y la falta de claridad logrando un agrupamiento difuso auto-espaciado. Además emplea CCF-ADB (*Connected-Component Filtering based on Area Density Balance*) para fusionar regiones pequeñas.

En el área de segmentación de imágenes médicas, se han propuesto distintos métodos, por ejemplo, un sistema de diagnóstico dental (Tuan et al., 2018) (Figura 2.12) que utiliza agrupamiento difuso semi-supervisado para segmentar la imagen de entrada, la clasificación la realizaron a partir de un agrupamiento basado en gráficos y con un proceso de toma de decisiones determinaron la enfermedad final. En (Albashah et al., 2018) utilizaron los conjuntos difusos intuitivos para segmentar coágulos de sangre, duros y blandos mediante el método *Fuzzy C-means* intuitivo espacial. Para hacer la detección de tumores cerebrales utilizaron el concepto de los conjuntos difusos tipo 2 para procesar el ruido de las imágenes, y mediante el algoritmo *Gaussian Kernel-based Fuzzy C-Means* detectan las regiones del tumor (Kaur and Chaira, 2021). Los conjuntos difuso intuitivos también se han combinado con otras características para mejorar la segmentación, como información espacial (Kumar et al., 2019; Kala and Deepa, 2021), otras teorías difusas (Chaira, 2021) e incluso diferentes medidas de distancia (Arora and Tushir, 2018; Kumar et al., 2020). Otra de las áreas

donde se han aplicado los conjuntos difusos es en la detección de cultivos, para diferenciar las regiones maduras de las no maduras, combinando el conjunto difuso tipo 2 (T2FS), K-Means y el algoritmo de optimización cuántica modificado (MQOA) (Huang et al., 2021b).



(c) Sistema de diagnóstico

Figura 2.12: Sistema de diagnóstico dental.

Otros métodos actuales para la segmentación de imágenes

Dentro de los métodos actuales para segmentar imágenes, las Redes Neuronales han sido investigadas recientemente, demostrando tener resultados eficientes (Tomita et al., 2018; Fang et al., 2018); un ejemplo es la red neuronal convolucional (CNN), de las más utilizadas, aplicada en el área médica para segmentar imágenes de resonancia magnéticas o ultrasonido (Minnema et al., 2018; Bullock et al., 2019; Karimi and Salcudean, 2019), en imágenes de carreteras (Reis et al., 2019; Fan et al., 2019), en imágenes de escenas interiores (Zhu et al., 2021), incluso son utilizadas para la compresión de escenas, tal es el caso de SegNet (Badrinarayanan et al., 2017), que utilizan

una red neuronal convolucional totalmente profunda para la segmentación semántica por píxeles, su arquitectura se presentan en la Figura 2.13, la cual cuenta con una red codificadora, una red decodificadora y la capa final para clasificar por píxeles.

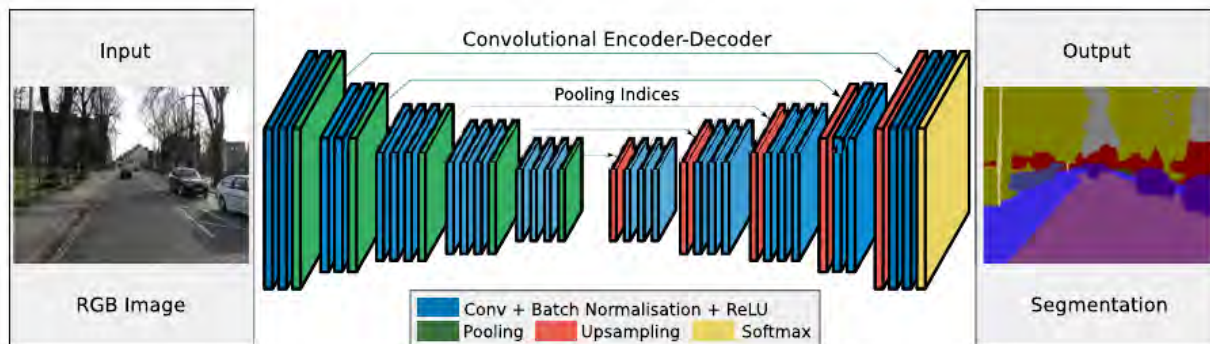


Figura 2.13: Arquitectura de SegNet (Badrinarayanan et al., 2017).

Las Redes Neuronales Convolucionales Recurrentes (RCNN) también han sido utilizadas tanto para segmentación, como clasificación y detección de imágenes médicas, mostrando buenos resultados, tal como se observa en la Figura 2.14 (Alom et al., 2019; Wollmann et al., 2019; Li et al., 2019). En la mayoría de las investigaciones con redes neuronales, mencionan que éstas pueden ser más robustas al ruido presente en las imágenes.

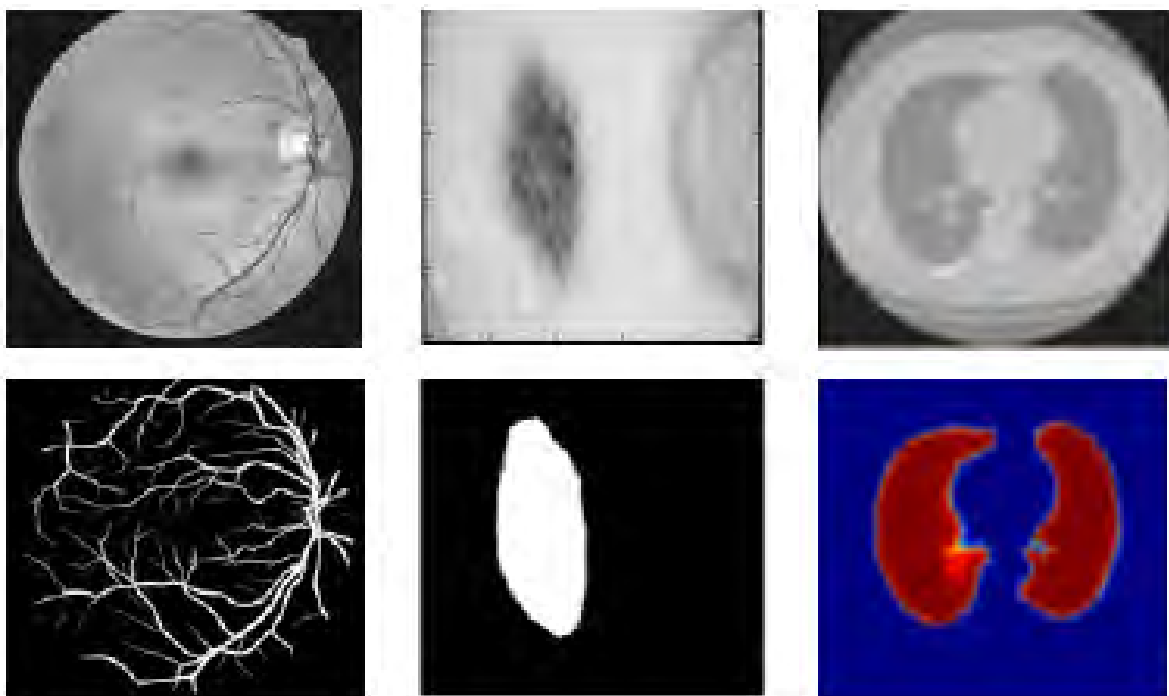


Figura 2.14: Arquitectura de SegNet (Alom et al., 2019).

Otros métodos utilizan reconstrucción morfológica para la segmentación en algoritmos de segmentación con semillas (Lei et al., 2019a); segmentación basada en autómatas celulares (en Figura 2.15 se presenta un ejemplo del método) en donde extrajeron y etiquetaron cuerpos vertebrales utilizando patrones binarios locales (LBP) y sus variantes.

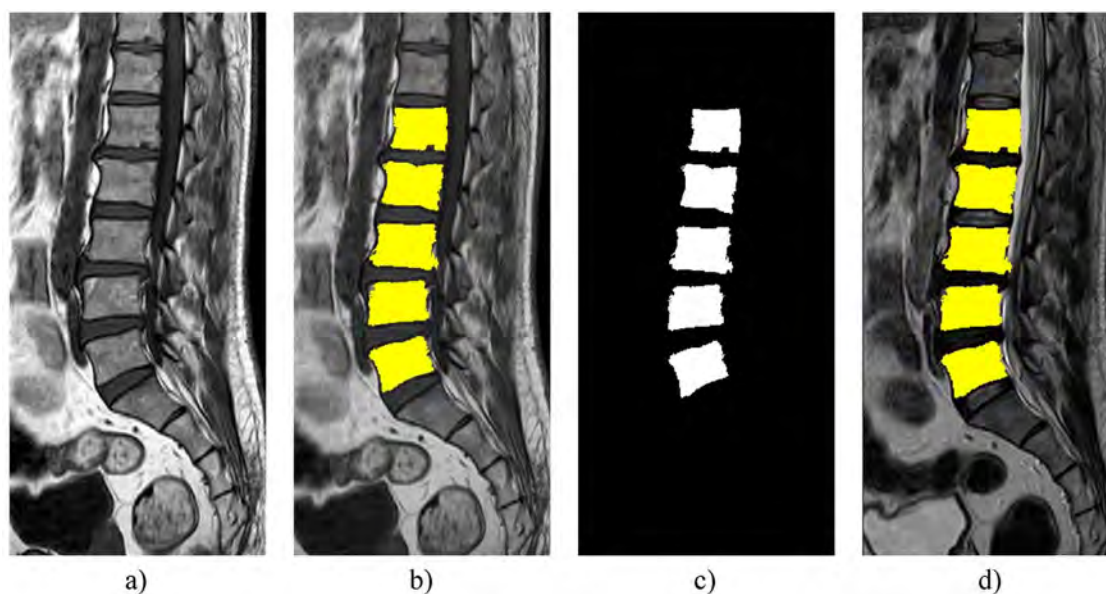


Figura 2.15: Segmentación usando autómatas celulares (Athertya et al., 2019).

Las técnicas de umbralización también son muy utilizadas para la segmentación a color, van desde aquellos algoritmos capaces de identificar adaptativamente los umbrales (Li et al., 2020; Zhou et al., 2020), hasta combinarlos con otras técnicas como Colonia de Abejas Artificiales (Zhang et al., 2018a; Khorram and Yazdi, 2019) o algoritmos genéticos (de Oliveira and Yamanaka, 2018) entre otros (Lei and Fan, 2019; Tan and Zhang, 2020). Un ejemplo de este tipo de técnicas se presenta en la Figura 2.16 que segmenta una imagen en cuatro colores, donde (a) es la imagen original, (b) el resultado inicial segmentado en 225 colores y (c) el resultado final segmentado en 4 colores.



Figura 2.16: Ejemplo de una técnica de umbralización. (Li et al., 2020).

Aplicaciones en el reconocimiento de patrones

En cuanto al reconocimiento de patrones el algoritmo KMM (*K-Multiple-Means*), (Nie et al., 2019) se basó en *k-means* para agrupa datos con múltiples medias de subgrupos en k grupos específicos. Este método formalizó el problema de agrupamiento de múltiples medias como un problema de optimización y actualiza las particiones de m medias de subgrupos y k grupos mediante una estrategia de optimización alterna. En cada iteración, las similitudes entre los datos y las medias de los subgrupos se actualizan siguiendo la partición del gráfico bipartito (Figura 2.17, izquierda: nodos como una sola componente conectada, derecha: Grafo bipartito estructurado con un número determinado de componentes conectados), y luego se actualizan las medias de los subgrupos.

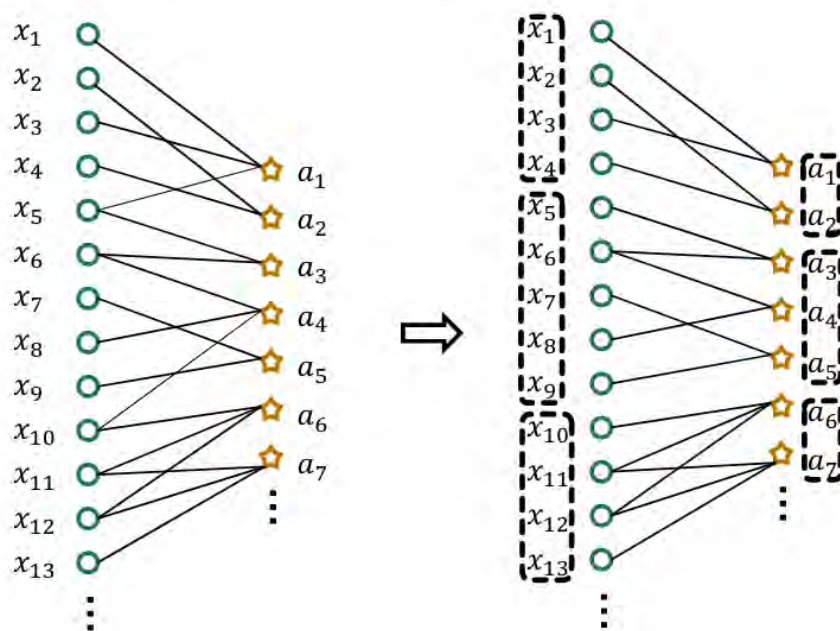


Figura 2.17: Ejemplo del grafo bipartito (Nie et al., 2019).

Dado que el algoritmo *Fuzzy C-Means* tiende a caer en mínimos locales, algunos autores han propuesto métodos que lo combinan con PSO (Particle Swarm Pptimization), tal es el caso del algoritmo HPSOFCM (Chen et al., 2014) y FCM-ELPSO de (Zhang and Ma, 2020), con esto se logra una mejora en el agrupamiento y la convergencia. Por otra parte, (Huang et al., 2021a) propusieron un modelo profundo robusto basado en *k-means* para aprender las representaciones ocultas asociadas con diferentes atributos implícitos de nivel inferior. Los datos se pueden explotar en capas, de tal manera que las muestras de datos de la misma clase se ven obligadas a estar más cerca capa por capa, beneficiando la tarea de agrupamiento (Figura 2.18). Los resultados obtenidos corroboraron que este modelo logró un avance en el rendimiento de la agrupación, comparado con métodos clásicos y de última generación.

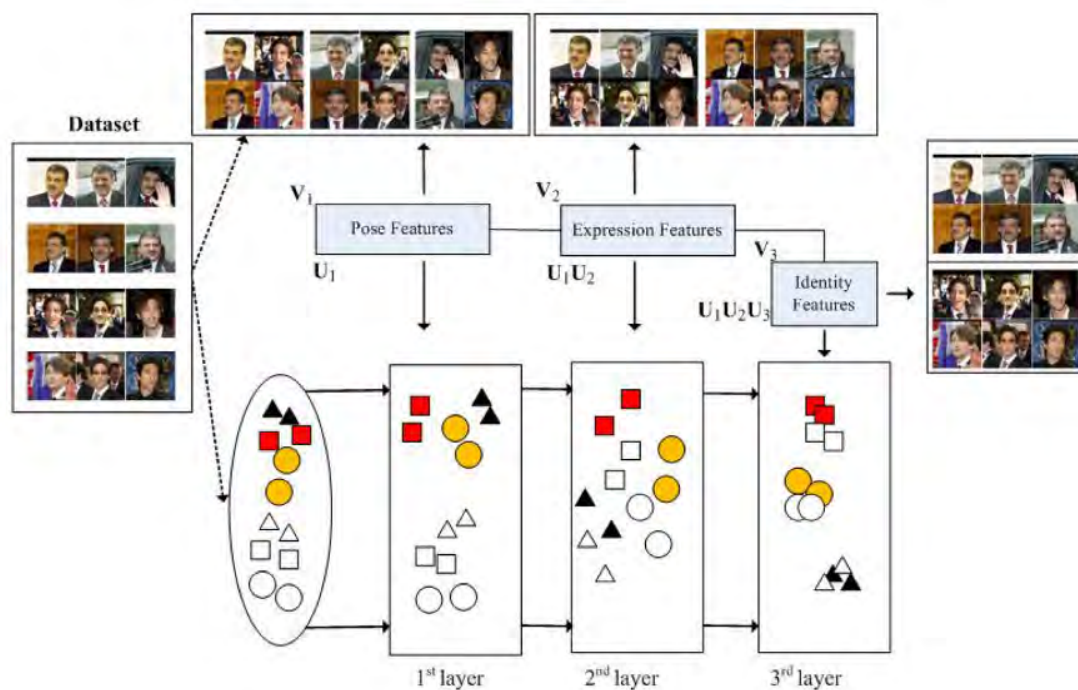


Figura 2.18: Modelo con una estructura k-means profunda de tres capas (Huang et al., 2021a).

En este capítulo se presentaron los conceptos de segmentación de imágenes y reconocimiento de patrones para comprender hacia donde se pueden aplicar las aportaciones de esta investigación, también se definió el algoritmo Fuzzy C-Means y una extensión basada en los conjuntos difusos intuitivos, así como los conjuntos difusos titubeantes. Seguido de eso se presentaron algunas investigaciones previas como antecedentes de este documento; además de una síntesis del estado del arte, en la que se destaca que los conjuntos difusos titubeantes tienen aplicaciones en la toma de decisiones, pero en el agrupamiento de datos son pocas, las cuales, principalmente están basadas en métodos que utilizan medidas de distancia o similitud, coeficientes de correlación o entropía para agrupar los datos. Teniendo en cuenta esta información, en el siguiente capítulo se describe la aportación de la investigación, presentando la formulación del algoritmo *Hesitant Fuzzy C-Means* y tres variantes que mejoran su rendimiento, el cual es aplicado tanto para el reconocimiento de objetos como para segmentar imágenes.

3 | Metodología

En esta sección se describen las principales aportaciones de esta investigación, primero se presenta la justificación matemática del algoritmo de agrupamiento basado en los conjuntos difusos titubeantes, el algoritmo *Hesitant Fuzzy C-Means*. Posteriormente, se detallan tres propuestas de mejora al algoritmo, la primera es para acelerar su tiempo de ejecución mediante una técnica de paralelización utilizando la biblioteca de OpenMP. La segunda surge de la necesidad de dar solución al problema de la inicialización de parámetros en la agrupación, se sugiere el uso del índice de Calinski-Harabasz para seleccionar el número óptimo de clusters en los que se agruparán los datos. Finalmente, se propone una alternativa para generar la información difusa titubeante, que según la literatura sugiere inicializar la matriz difusa titubeante y no hacer uso de otros algoritmos de agrupamiento.

3.1. Formulación algoritmo *Hesitant Fuzzy C-Means*

El algoritmo propuesto esta basado en la teoría de los conjuntos difusos titubeantes; dado un conjunto de datos $X = \{x_1, \dots, x_n\}$ que debe ser agrupado en c grupos, es necesario establecer para cada dato varios posibles grados de pertenencia $h(x)$ en $[0, 1]$ y reunirlos en una matriz H , denominada matriz difusa titubeante (Torra, 2010). Para ser una partición difusa debe tener dos condiciones que garanticen que ningún grupo se quede vacío (Ecuación 3.1) y que la suma de los grados de pertenencia de cada dato sea igual a 1 (Ecuación 3.2).

$$\sum_{i=1}^N u_{ij} > 0, \quad 1 \leq i \leq c \quad (3.1)$$

$$\sum_{j=1}^c u_{ij} = 1, \quad 1 \leq i \leq N \quad (3.2)$$

donde u_{ij} es interpretado como el grado de pertenencia del i -ésimo dato del j -ésimo grupo, N es el total de datos dentro del conjunto.

Para plantear un algoritmo de agrupamiento, se requiere formular una función objetivo que minimice las distancias entre los centros de los grupos y los datos, ya que mientras más cerca esté un dato a un centro su grado de pertenencia será mayor. Entonces se propone la siguiente función objetivo:

$$J_m(X_{HFS}; U, V_{HFS}) = \sum_{i=1}^N \sum_{j=1}^c u_{ij}^m d^2(h_i, v_j) \quad (3.3)$$

donde $X_{HFS} = \{h_1, h_2, \dots, h_n\}$ representa N elementos difusos titubeantes correspondiente a cada dato, u_{ij} es el grado de pertenencia del i -ésimo dato del j -ésimo grupo. $U = (u_{ij})_{c \times N}$ es la matriz de pertenencia difusa. c es el número de grupos ($1 \leq c \leq N$), m es el parámetro fuzzificador o de ponderación ($m > 1$). $V_{HFS} = \{v_1, v_2, \dots, v_j\}$ son los centros de los grupos, cada v_j está representado en términos de conjuntos difusos titubeantes $v_j = \{u_1(v_j), u_2(v_j), \dots, u_l(v_j)\}$ donde l es el número de los posibles grados de pertenencia de cada dato. Finalmente, $d^2(h_i, v_j)$ es la medida de distancia euclidiana normalizada titubeante entre los datos y los centros de los grupos, definida por Xu and Xia (2011) y descrita en la Ecuación 2.21.

La función objetivo de la Ecuación 3.3 debe ser minimizada mediante un proceso de optimización, la técnica de los multiplicadores de Lagrange (Rockafellar, 1993) permite encontrar el mínimo de una función con múltiples variables, siempre que exista alguna restricción en los valores de entrada. El primer paso es introducir una nueva variable λ y definir una nueva función:

$$\mathcal{L} = \sum_{i=1}^N \sum_{j=1}^c u_{ij}^m d^2(h_i, v_j) - \sum_{i=1}^N \lambda \left(\sum_{j=1}^c u_{ij} - 1 \right) \quad (3.4)$$

Después se deben establecer las derivadas parciales a cero:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial u_{ij}} = 0, & \Leftrightarrow & m u_{ij}^{m-1} d^2(h_i, v_j) = \lambda \\ \frac{\partial \mathcal{L}}{\partial \lambda} = 0, & \Leftrightarrow & \sum_{j=1}^c u_{ij} = 1 \end{cases} \quad (3.5)$$

Despejando u_{ij} de la Ecuación 3.5.a, se tiene

$$u_{ij} = \left[\frac{\lambda}{m d^2(h_i, v_j)} \right]^{\frac{1}{m-1}} \quad (3.6)$$

Se realiza una suma de estas ecuaciones sobre los grupos y se tiene:

$$\begin{aligned} \sum_{s=1}^c u_{is} &= \sum_{s=1}^c \left[\frac{\lambda}{m d^2(h_i, v_s)} \right]^{\frac{1}{m-1}} \\ &= \sum_{s=1}^c \left(\frac{\lambda}{m} \right)^{\frac{1}{m-1}} \left[\frac{1}{d^2(h_i, v_s)} \right]^{\frac{1}{m-1}} \\ &= \left(\frac{\lambda}{m} \right)^{\frac{1}{m-1}} \left\{ \sum_{s=1}^c \left[\frac{1}{d^2(h_i, v_s)} \right]^{\frac{1}{m-1}} \right\} = 1 \end{aligned} \quad (3.7)$$

Entonces se despeja $\left(\frac{\lambda}{m} \right)^{\frac{1}{m-1}}$

$$\left(\frac{\lambda}{m} \right)^{\frac{1}{m-1}} = \frac{1}{\sum_{s=1}^c \left(\frac{1}{d^2(h_i, v_s)} \right)^{\frac{1}{m-1}}} \quad (3.8)$$

Y sustituyendo en la Ecuación 3.6 se obtiene la ecuación de actualización de los grados de pertenencia:

$$\begin{aligned}
u_{ij} &= \left\{ \frac{1}{\sum_{s=1}^c \left[\frac{1}{d^2(h_i, v_s)} \right]^{\frac{1}{(m-1)}}} \right\} \left[\frac{1}{d^2(h_i, v_j)} \right]^{\frac{1}{(m-1)}} \\
&= \frac{1}{\sum_{s=1}^c \left(\frac{d^2(h_i, v_j)}{d^2(h_i, v_s)} \right)^{\frac{2}{(m-1)}}} \\
&= \frac{d^{-\frac{2}{m-1}}(h_i, v_j)}{\sum_{s=1}^c d^{-\frac{2}{m-1}}(h_i, v_s)}
\end{aligned} \tag{3.9}$$

Finalmente, los centros de los grupos son calculados en un proceso similar, a partir de las siguientes expresiones:

$$\begin{aligned}
\forall 1 \leq j \leq c, 1 \leq i \leq N, \\
\frac{\partial L}{\partial u_1(v_j)} = \frac{\partial L}{\partial u_2(v_j)} = \dots = \frac{\partial L}{\partial u_l(v_j)} = 0
\end{aligned} \tag{3.10}$$

donde l es el número de posibles grados de pertenencia de cada dato, entonces, estableciendo la derivada parcial a cero:

$$\frac{\partial L}{\partial u_t(v_j)} = 0, \quad v_j = \frac{\sum_{i=1}^N u_{ij}^m h_i}{\sum_{i=1}^N u_{ij}^m} \tag{3.11}$$

La implementación del algoritmo propuesto *Hesitant Fuzzy C-Means* (HFCM), para la tarea de segmentación de imágenes se resume en el Algoritmo 1. Como se indica en el pseudocódigo, los datos de entrada son la imagen X de tamaño $M \times N$ y el número de regiones c en las que se va a segmentar la imagen donde $1 < c < n$. Primero que nada, se sugiere el parámetro fuzzificador o de ponderación y el criterio de parada para obtener mejores resultados según la literatura como $m = 2$ y $\varepsilon = 0.00001$, respectivamente. Los datos de la imagen deben ser expresados en información difusa titubeante, por lo que cada pixel debe ser transformado al dominio difuso con:

$$\mu(x_i) = \frac{x_i - \text{mín}(x)}{\text{máx}(x) - \text{mín}(x)} \tag{3.12}$$

donde $\mu(x_i)$ es el grado de pertenencia del pixel x_i . Luego, es necesario llenar la matriz difusa titubeante (HFM) $H = h(x) \in [0, 1]$. En la literatura aún no existe una forma exacta para obtener este tipo de información, ya que depende de la evaluación de expertos, por eso para el algoritmo HFCM se propone como una alternativa realizar l ejecuciones con uno o varios algoritmos de agrupamiento difuso, particularmente se utilizan tres ejecuciones con el algoritmo FCM (*Fuzzy C-Means*), de esta manera cada pixel tiene su correspondiente elemento difuso titubeante (HFE), $x_i = h(x_i)$ donde:

$$h(x_i) = \{\mu_{ij}^k | 0 \leq \mu_{ij}^k \leq 1, k = 1, \dots, l\} \quad \forall x \in X \tag{3.13}$$

El siguiente paso es asignar los valores iniciales a los centros de los grupos V_{HFS} de manera aleatoria y con ellos obtener la matriz de pertenencia difusa con la Ecuación 3.9. Después de eso, se lleva a cabo un procedimiento iterativo hasta que se cumpla el criterio de parada $\|u_{ij}^a - u_{ij}^{a-1}\| < \varepsilon$. Dentro de este, en cada iteración se aumenta un índice a y los centros de los grupos se actualizan mediante la Ecuación 3.11, tal como se sugiere en el paso 7. Posteriormente, la matriz de pertenencia difusa se actualiza en el paso 8 con la Ecuación 3.9. Finalmente, el resultado del algoritmo es la imagen Y segmentada en c regiones obtenida a partir valores máximos de U_{ij} .

Algorithm 1 Hesitant Fuzzy C-Means

Input: Image X , number of regions c

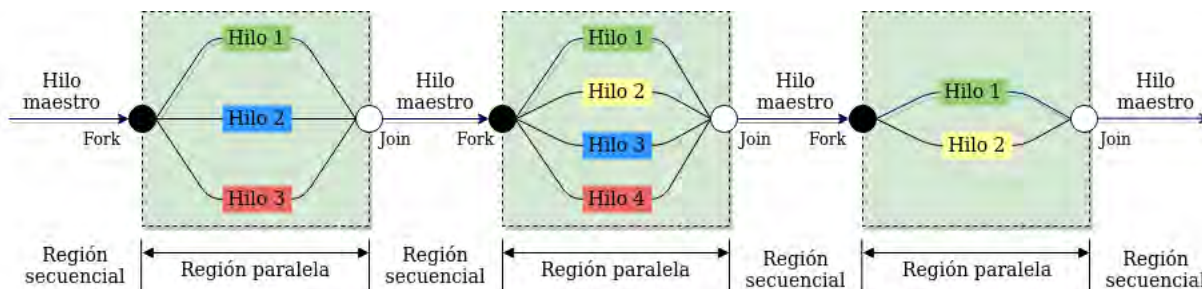
Output: Segmented image Y

- 1: Set m and ε
 - 2: $X_{HFS} \leftarrow X$
 - 3: $H \leftarrow h(x) \in [0, 1]$
 - 4: $v_j \leftarrow \text{rand}([0, 1])$
 - 5: $u_{ij} \leftarrow \frac{1}{\sum_{s=1}^c \left(\frac{d^2(h_i, v_j)}{d^2(h_i, v_s)} \right)^{\frac{2}{(m-1)}}$
 - 6: **while** $\|u_{ij}^a - u_{ij}^{a-1}\| < \varepsilon$ **do**
 - 7: $v_j \leftarrow \frac{\sum_{i=1}^N u_{ij}^m h_i}{\sum_{i=1}^N u_{ij}^m}$
 - 8: $u_{ij} \leftarrow \frac{1}{\sum_{s=1}^c \left(\frac{d^2(h_i, v_j)}{d^2(h_i, v_s)} \right)^{\frac{2}{(m-1)}}$
 - 9: **end while**
 - 10: $Y \leftarrow \text{argmax}_{j=1}^c U_{ij}$
 - 11: **return** Y
-

3.2. Algoritmo *Parallel Hesitant Fuzzy C-Means*

Dado que el algoritmo HFCM utiliza más información al incorporar varios grados posibles de pertenencia para cada pixel, su costo computacional es alto, la estrategia para reducirlo es una implementación paralela utilizando la biblioteca OpenMP (*Open Multi-Processing*). Esta es una interfaz para la programación paralela de memoria compartida independiente de la plataforma en C, C++ y Fortran; se basa en el modelo *fork-join*, donde una tarea muy pesada se divide en subprocesos (*fork*) con menos peso, y luego sus resultados son recopilados al final y se unen en un solo resultado (*join*), Schmidt et al. (2017), este proceso se ilustra en la Figura 3.1, donde se observa un hilo maestro que llega a un punto donde se bifurca en dos, tres o cuatro hilos que realizan tareas simultáneamente para posteriormente unirse y continuar con la región secuencial hasta que encuentre otra región paralela.

OpenMP utiliza un paralelismo a nivel de bucle, es decir, paraleliza aquellos bucles donde las iteraciones se pueden ejecutar de forma independiente. La filosofía básica de OpenMP es aumentar el código secuencial mediante el uso de directivas de compilador especiales similares a comentarios, llamadas *pragmas*, para dar al compilador pistas sobre cómo se puede paralelizar el código. La sintaxis que siguen estos pragmas en lenguaje C y C++ es `#pragma omp construct[clause[clause]. . .]`,

Figura 3.1: Proceso *Fork-Join*.

la palabra clave `omp` hace que los compiladores de OpenMP la procesen, mientras que los que no lo son la ignoren. Esto permite utilizar la misma base del código secuencial para construir la versión paralela, con solo agregar los pragmas apropiados. En la Tabla 3.1 se presenta un ejemplo del código secuencial y paralelo, en este ambos realizan la misma tarea, llenar la matriz difusa titubeante; se observa que la estructura es la misma, la única diferencia es que el código paralelo tiene dos líneas de código extra con las directivas de OpenMP (`#pragma...`).

Tabla 3.1: Ejemplos de código secuencial y paralelo.

Código secuencial	Código paralelo
<pre> 1 void MatHFCM(float *data,float *DataHFS ↪ ,float **Ux) 2 { 3 int index =0; 4 for(int k=0;k<l;k++){ 5 index=k; 6 fcm(data,Ux); 7 for(int i=0;i<N;i++){ 8 for(int j=0;j<c;j++){ 9 DataHFS[index]=Ux[i][j]; 10 index=index+l; 11 } 12 } 13 } 14 } 15 16 </pre>	<pre> 1 void MatHFCM(float *data,float *DataHFS, ↪ float **Ux) 2 { 3 int index=0,k,i,j; 4 #pragma omp parallel 5 #pragma omp for linear(index:l) 6 for(k=0;k<l;k++){ 7 index=k; 8 fcm(data,Ux); 9 for(i=0;i<N;i++){ 10 for(j=0;j<c;j++){ 11 *(DataHFS+index)=*(Ux+i+j); 12 index=index+l; 13 } 14 } 15 } 16 } </pre>

Las directivas utilizadas para realizar la implementación paralela del algoritmo propuesto se describen en la Tabla 3.2, indicando su nombre, su propósito y la sintaxis de cada una.

Tabla 3.2: Directivas de OpenMP utilizadas.

Directiva	Propósito	Sintaxis
parallel	Define una región paralela que puede ejecutarse por varios núcleos/hilos en forma simultánea.	<pre> 1 #pragma omp parallel [clausula[⇔ [,]clausula] ...] 2 { 3 ... 4 }</pre>
for	Indica que las tareas dentro de un ciclo for serán divididas entre los hilos activos. Por lo regular, se utiliza una declaración compuesta de las directivas parallel y for (<code>#pragma omp parallel for</code>).	<pre> 1 #pragma omp parallel 2 { 3 #pragma omp for 4 for (...) 5 { 6 ... 7 } 8 }</pre>
simd	Indica que el bucle se puede transformar en un bucle SIMD (Single-Instruction, Multiple-Data), significa que el procesador realizará múltiples cálculos simultáneamente, utilizando instrucciones especiales que realizan el mismo cálculo para múltiples valores a la vez.	<pre> 1 #pragma omp simd [clause[[,] ⇔ clause] ...] 2 for(...) 3 { 4 ... 5 }</pre>
taskloop	Especifica que las iteraciones de uno o más bucles asociados se ejecutarán en paralelo utilizando tareas explícitas.	<pre> 1 #pragma omp taskloop [clause[⇔ [,]clause] ...] 2 for(...) 3 { 4 ... 5 }</pre>

Además cada una de estas directivas comparten diferentes cláusulas, algunas de estas son descritas a continuación, la mayoría acepta una lista de variables separadas por comas.

- **default(shared | none):** Especifica como se comportan las variables sin cobertura en la región paralela.
- **shared(variable):** Especifica que una o más variables deben ser compartidas a través de todos los núcleos o hilos.
- **private(variable):** Especifica que las variables son privadas para cada hilo, es decir, cada hilo debe tener su propia instancia de la variable.
- **reduction(operación:variable):** Especifica que una o más variables privadas para cada hilo están sujetas a una operación y se acumulan al final de una región paralela. Los operadores pueden ser +, *, -, &, |, ^, && y || .
- **linear(variable:incremento):** Permite la paralelización adecuada del ciclo *for* al contener variables que se incrementan en cantidades fijas en cada iteración.
- **schedule(kind):** Especifica como se dividen las iteraciones de los bucles asociados y como se distribuyen en los núcleos o hilos. Existen distintos tipos: *static*, *dynamic*, *guided*, *auto*, *runtime*.

Finalmente, la implementación del algoritmo *Hesitant Fuzzy C-Means* con las directivas de OpenMP, da como resultado el algoritmo propuesto *Parallel Hesitant Fuzzy C-Means* (PHFCM), solo para ilustrar su funcionamiento se presenta la Figura 3.2, en la que resaltan dos etapas paralelas, la primera se encuentra en la asignación de los valores de la matriz difusa titubeante H y la segunda en la actualización de los centros de los grupos v_j y la matriz de pertenencia difusa u_{ij} .

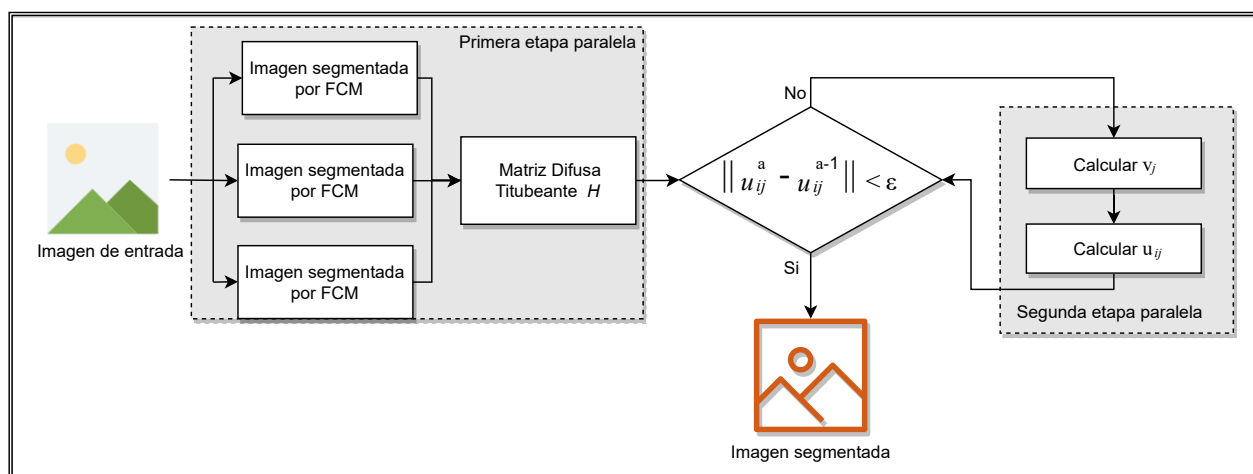


Figura 3.2: Implementación del algoritmo *Parallel Hesitant Fuzzy C-Means*.

Este algoritmo requiere como entrada una imagen (X), la cual debe ser transformada en información difusa titubeante para obtener la matriz H , este proceso se lleva a cabo mediante una región paralela en donde el hilo maestro se divide en tres hilos esclavos y en cada uno se ejecuta el algoritmo FCM, el resultado final de cada hilo es almacenado en la variable de la matriz H .

El siguiente bloque de código presenta la función `MathFCM` que realiza este proceso, utiliza las instrucciones `#pragma omp parallel` y `#pragma omp for linear(index:k)` para indicarle al compilador que debe iniciar una región paralela, además esta contiene un ciclo `for` con una variable `index` que se va incrementando en `l` para ir almacenando los resultados de la matriz `H` con respecto a cada grupo.

```

1 void MathFCM(float *data,float *DataHFS,float **Ux)
2 {
3     int index=0,k,i,j;
4     #pragma omp parallel
5     #pragma omp for linear(index:l)
6     for(k=0;k<l;k++){
7         index=k;
8         fcm(data,Ux);
9         for(i=0;i<N;i++){
10            for(j=0;j<c;j++){
11                *(DataHFS+index)=*(*(Ux+i)+j);
12                index=index+l;
13            }
14        }
15    }
16 }

```

Posteriormente, se inicia un proceso iterativo que esta dividido en regiones paralelas, la primera es para calcular los centros de los grupos v_j (Ecuación 3.11), esta se realiza con la función `Centroids` en la cual se le indica al compilador mediante la instrucción `#pragma omp simd` que debe realizar simultáneamente múltiples cálculos, también se agrega `#pragma omp parallel for schedule(static) reduction(+:protN, protD)` para indicar que las iteraciones del ciclo `for` serán divididas en distintos fragmentos con aproximadamente el mismo tamaño y distribuidos uno en cada hilo, además que las variables `protN` y `protD` estarán sujetas a una operación de suma acumulada al final de la región paralela.

```

1 void Centroids(float *data,float *centroids,float **um)
2 {
3     #pragma omp simd
4     for(j = 0; j<c;j++){
5         for(k = 0;k<T;k++){
6             protN = protD = 0.0;
7             #pragma omp parallel for schedule(static,8) reduction(+:protN, protD)
8             for(i = 0;i<N;i++){
9                 protN += (*(um+i)+j) * data[(i*T)+k];
10                protD += (*(um+i)+j);
11            }
12            *(centroids+index) = protN / protD;
13            index++;
14        }
15    }
16 }

```


Seguido de esto, inicia la segunda región paralela dentro del mismo proceso iterativo para hacer el cálculo de la matriz de pertenencias difusas u_{ij} (Ecuación 3.9) con la función `MembershipMatrix`, esta inicia indicándole al compilador que las variables deben ser compartidas en todos los hilos con la instrucción `#pragma omp parallel default(shared)` y con `#pragma omp taskloop` divide el ciclo en distintos fragmentos y crea tareas para cada uno; estas tareas son para el cálculo de la distancia entre cada dato y los centros de los grupos. Después mediante la instrucción `#pragma omp parallel for private(i,j) reduction(+:aux)` indica que las variables i y j son privadas para cada hilo, además que la variable aux esta sujeta a una suma. Finalmente, cuando se cumple con el criterio de parada, se utiliza el último u_{ij} para obtener la imagen sementada en c regiones.

```

1 void MembershipMatrix(float *data,float *centroids,float **u0)
2 {
3     #pragma omp parallel default(shared)
4     {
5         #pragma omp for
6         for(j=0;j<c;j++){
7             index=0.0;
8             i_n=0;
9             aa =(j*T)+T;
10            for (k = j*T; k<aa;k++)
11                *(Bv+index++) = *(centroids+k);
12            #pragma omp taskloop
13            for(i=0;i<Nbpp;i=i+T){
14                index=0.0;
15                aa =i+T;
16                for (int k = i; k<aa;k++)
17                    *(Av+index++)= *(data+k);
18                *(di+i_n)= HNEuclideanDistance(Av, Bv);
19                (*(dist+i_n)+j) += *(di+i_n);
20                (*(dist+i_n)+j) = pow((*(dist+i_n)+j) + 1e-10,(-1.0/(m-1.0)));
21                i_n++;
22            }
23        }
24    }
25
26    #pragma omp parallel for private(i,j) reduction(+:aux)
27    for(i=0;i<N;i++){
28        aux=0;
29        for(j=0;j<c;j++){
30            aux += (*(dist+i)+j);
31            for(j=0;j<c;j++){
32                (*(sumdist+i)+j) = aux;
33                (*(u0+i)+j) = (*(dist+i)+j) / (*(sumdist+i)+j);
34            }
35        }
36    }

```

3.3. Algoritmo *Automatic Hesitant Fuzzy C-Means*

Uno de los parámetros del algoritmo HFCM que debe ser especificado es el número de grupos c , para no tener que depender de este parámetro se propone el uso del índice de Calinski-Harabasz (CH) (Caliński and Harabasz, 1974), es una medida de que tan similar es un objeto a su propio grupo (cohesión) en comparación a otros grupos (separación). La cohesión esta en función de las distancias de los datos en el grupo al centro del grupo y la separación se basa en la distancia de los centros del grupo al centro global (Maulik and Bandyopadhyay, 2002; Liu et al., 2010).

La formula del índice de Calinski-Harabasz, para K grupos en un conjunto de datos $X = [x_1, x_2, \dots, x_n]$, esta definida como:

$$CH = \left[\frac{\sum_{k=1}^K n_k \|c_k - c\|^2}{(K-1)} \right] / \left[\frac{\sum_{k=1}^K \sum_{i=1}^{n_k} \|x_i - c_k\|^2}{(n-K)} \right] \quad (3.14)$$

donde n_k y c_k son el número de datos y el centroide del k -ésimo grupo, respectivamente y c es el centroide del conjunto de datos, el centroide global. n el número total de datos. Un valor más alto del índice Calinski-Harabasz significa que los grupos son densos y están bien separados, indicando que la agrupación es mejor. En la Figura 3.3 se visualiza el proceso del algoritmo *Automatic Hesitant Fuzzy C-means* (AHFCM) que utiliza este índice para seleccionar el número de centroides.

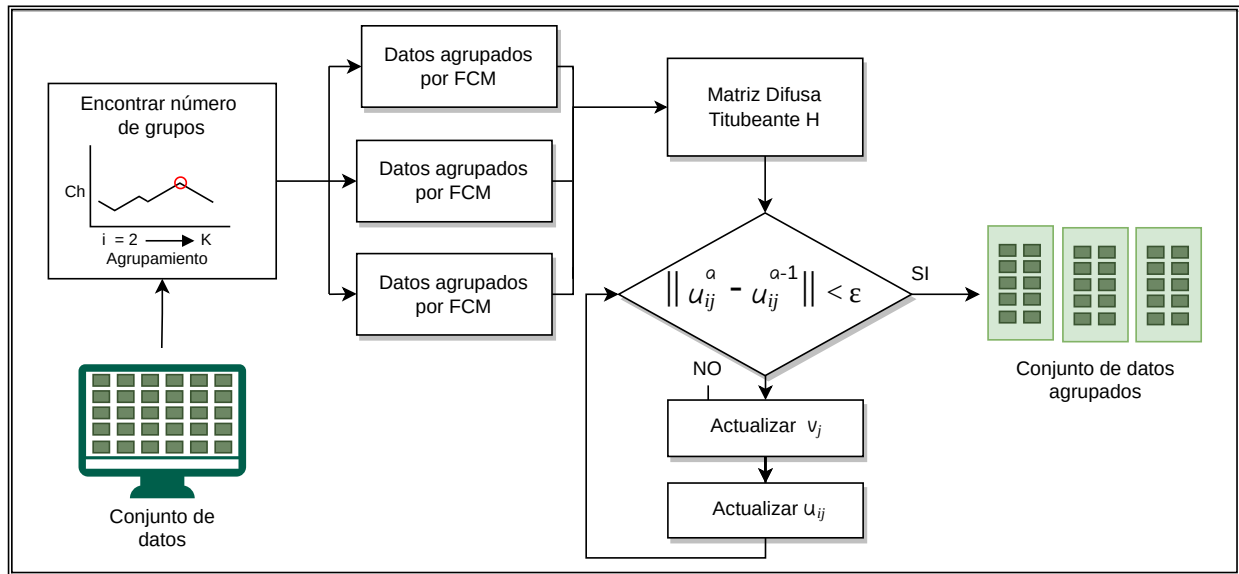


Figura 3.3: Proceso del algoritmo *Automatic Hesitant Fuzzy C-Means*.

La implementación del algoritmo *Automatic Hesitant Fuzzy C-Means*, para el agrupamiento de datos se resume en el Algoritmo 2. Tal como se indica en el pseudocódigo, los datos de entrada son un conjunto de datos X de tamaño $M \times N$. El parámetro fuzzificador o de ponderación y el criterio de parada se sugieren para obtener mejores resultados según la literatura como $m = 2$ y $\varepsilon = 0.00001$, respectivamente. El número de grupos C es determinado por el índice de Calinski-Harabasz descrito en la ecuación 3.14 y es calculado con el resultado del algoritmo FCM para 2 a K grupos y seleccionando el valor más alto; el proceso se detalla en el Algoritmo 3. El valor de K es definido por el usuario bajo distintas consideraciones, algunas de ellas pueden ser: 1)

al ser un agrupamiento por contenido y no por instancias, se tiene un intervalo de pocos objetos diferentes; 2) basadas en el conjunto de datos es decir, se requiere de un análisis de los datos con respecto al *ground truth* o etiquetas; 3) de acuerdo a la capacidad computacional, seleccionando el número de núcleos del procesador. Sin embargo, el valor de K puede ser cualquier número entero ($2 \leq K \leq M \times N$) y dependerá de la perspectiva de cada usuario.

Posteriormente, se sigue el mismo proceso que el algoritmo HFCM para realizar la agrupación, que es transformar los datos a información difusa titubeante, asignar aleatoriamente los valores iniciales a los centros de los grupos V_{HFS} y obtener la matriz de pertenencia difusa U_{ij} para finalmente iniciar el proceso iterativo, al cumplirse el criterio de parada se obtienen los datos agrupados a partir de los valores máximos de la última U_{ij} .

Algorithm 2 Automatic Hesitant Fuzzy C-Means

Input: Dataset X

Output: Clustered dataset Y

- 1: Set m and ε ($m = 2$ and $\varepsilon = 0.00001$ as default)
 - 2: Find number of clusters by Algorithm 3
 - 3: $X_{HFS} \leftarrow X$
 - 4: Initialize v_j and u_{ij}
 - 5: **while** $\|u_{ij}^a - u_{ij}^{a-1}\| < \varepsilon$ **do**
 - 6: $v_j \leftarrow \frac{\sum_{i=1}^N u_{ij}^m h_i}{\sum_{i=1}^N u_{ij}^m}$
 - 7: $u_{ij} \leftarrow \frac{1}{\sum_{c=1}^C \left(\frac{d^2(h_i, v_j)}{d^2(h_i, v_c)} \right)^{\frac{2}{m-1}}}$
 - 8: **end while**
 - 9: $Y \leftarrow \operatorname{argmax}_{j=1}^C U_{ij}$
 - 10: **return** Y
-

Algorithm 3 Find number of clusters

Input: Dataset X

Output: C

- 1: Set K
 - 2: **for** $i \leftarrow 1$ to K **do**
 - 3: $clusters(i) \leftarrow FCM(X, i)$
 - 4: $CH(i) \leftarrow \left[\frac{\sum_{k=1}^K n_k \|c_k - c\|^2}{(K-1)} \right] / \left[\frac{\sum_{k=1}^K \sum_{i=1}^{n_k} \|x_i - c_k\|^2}{(n-K)} \right]$
 - 5: **if** $CH(i) > CH(max_value)$ **then**
 - 6: $max_value \leftarrow i$
 - 7: **end if**
 - 8: **end for**
 - 9: $C = max_value$
 - 10: **return** C
-

3.4. Algoritmo *Fast and Automatic Hesitant Fuzzy Clustering*

La literatura describe que no existe alguna base de datos que contenga información difusa titubeante, ya que esta es expresada a través de expertos. En los algoritmos presentados en esta investigación se ha optado por utilizar el resultado de ejecutar tres veces el algoritmo FCM y se ha explicado que este tipo de información considera múltiples grados de pertenencia para cada dato. En esta sección se propone una alternativa para generar la información difusa titubeante, y es mediante funciones de pertenencia basada en relaciones difusas, descritas a continuación:

- Homogeneidad, calculada dentro de una vecindad de 3×3 , se considera un conjunto difuso y su función de pertenencia es definida como:

$$\mu_h = 1 - \frac{G^{max,l} - G^{min,l}}{G^{max,g} - G^{min,g}} \quad (3.15)$$

donde $G^{min,l}, G^{max,l}, G^{min,g}$ y $G^{max,g}$ son los niveles de gris mínimo y máximo locales y globales, respectivamente.

- Espacio de color, se realiza una transformación al espacio de color IJK, que es una rotación al espacio de color RGB. Primero es necesario transformar cada pixel de la imagen al dominio difuso, mediante la Ecuación 3.12, posteriormente, la conversión del espacio de color RGB a IJK se realiza con las siguientes ecuaciones (Pătraşcu, 2007):

$$I = \frac{R + G + B}{3} \quad (3.16)$$

$$J = \frac{2R - G - B}{2\sqrt{3}} \quad (3.17)$$

$$K = \frac{G - B}{\sqrt{6}} \quad (3.18)$$

- Texturas, se hace uso de enfoques estadísticos para describir la textura de las imágenes, con el objetivo de tener un algoritmo que sea simple de implementar. Las técnicas para describir las texturas que utiliza el algoritmo son:

Entropía difusa, que cuantifica la incertidumbre del contenido de la imagen, definida por:

$$H_{log}(G) = \frac{1}{MN \ln 2} \sum_{m,n} S_n(\mu_{mn}) \quad (3.19)$$

donde:

$$S_n(\mu_{mn}) = -\mu_{mn} \ln(\mu_{mn}) - (1 - \mu_{mn}) \ln(1 - \mu_{mn}) \quad (3.20)$$

Desviación estándar, se utiliza dado que sus valores suelen ser más intuitivos, y se describe mediante:

$$\sigma(z) = \sqrt{\sum_{i=0}^{L-1} (z_i - m)^n p(z_i)} \quad (3.21)$$

donde m es el valor promedio de la intensidad:

$$m = \sum_{i=0}^{L-1} z_i p(z_i) \quad (3.22)$$

Se ha mencionado con anterioridad que la matriz difusa titubeante contiene un conjunto de valores en $[0, 1]$ que denotan los posibles grados de pertenencia de un elemento hacia el conjunto; por lo que la propuesta es hacer uso de las tres relaciones descritas para inicializar esta matriz, tal como se indica en la Ecuación 3.13 donde k representa cada una de las l relaciones establecidas (homogeneidad, color y texturas).

El algoritmo propuesto es aplicado a la segmentación de imágenes y esta dividido en tres fases, 1) filtrar la imagen X , y hacer una reducción, f , para encontrar el número de grupos C mediante el índice de Calinski-Harabasz (CH), en esta fase también se obtiene la matriz difusa titubeante de la imagen f , H_f ; 2) se utiliza la matriz H_f para calcular los centroides los cuales pasarán a la fase 3) como los centroides iniciales que segmentarán la imagen X , para esta tercera fase es necesario calcular la matriz difusa titubeante H_x . Este algoritmo, denominado *Fast and Automatic Hesitant Fuzzy Clustering* (FAHFC), utiliza la base matemática del algoritmo *Hesitant Fuzzy C-Means*. En la Figura 3.4 se visualiza el proceso descrito anteriormente.

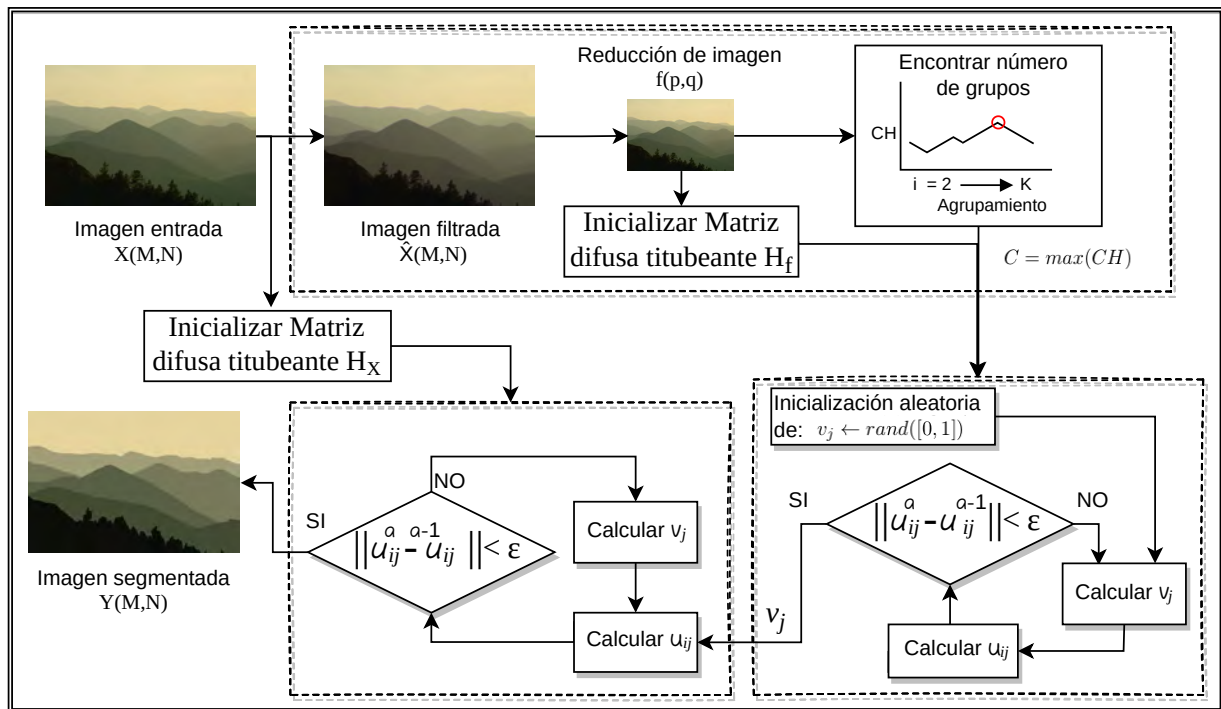


Figura 3.4: Proceso del algoritmo *Fast and Automatic Hesitant Fuzzy Clustering*.

Para mayor detalle, el Algoritmo 4 resume la implementación de este algoritmo, el cual tiene como entrada una imagen a color, que posteriormente será filtrada, en los experimentos se utiliza

el filtro mediana, ya que presenta píxeles reales dentro de la imagen, y es menos sensible a valores extremos. Seguido se hace una reducción de la imagen (Algoritmo 5) para encontrar el número de grupos en los que se segmentará la imagen, así como para encontrar los centroides iniciales, todo esto en la imagen reducida f . En los pasos 4 y 5 se hace la transformación de la imagen X al dominio difuso titubeante, y la inicialización de la matriz difusa titubeante utilizando la alternativa descrita anteriormente, mediante las relaciones difusas de homogeneidad, color y texturas. Finalmente, se realiza un proceso iterativo para actualizar los centroides de los grupos y la matriz de pertenencia difusa y obtener la segmentación de la imagen original, tal como lo realiza el algoritmo HFCM.

Algorithm 4 Fast and Automatic Hesitant Fuzzy Clustering

Input: Image $X(M, N)$
Output: Segmented image $Y(M, N)$

- 1: Set m and ε ($m = 2$ and $\varepsilon = 0.00001$ as default)
 - 2: *Reduction (Algorithm 5)*
 - 3: $X_{HFS} \leftarrow X$
 - 4: $H_X \leftarrow h(x) \in [0, 1]$
 - 5: **while** $\|u_{ij}^a - u_{ij}^{a-1}\| < \varepsilon$ **do**
 - 6: $v_j \leftarrow \frac{\sum_{i=1}^N u_{ij}^m h_i}{\sum_{i=1}^N u_{ij}^m}$
 - 7: $u_{ij} \leftarrow \frac{1}{\sum_{c=1}^C \left(\frac{d^2(h_i, v_j)}{d^2(h_i, v_c)} \right)^{\frac{2}{(m-1)}}$
 - 8: **end while**
 - 9: $Y \leftarrow \operatorname{argmax}_{j=1}^C U_{ij}$
 - 10: **return** Y
-

Algorithm 5 Reduction

Input: $X(M, N)$
Output: v_j

- 1: $\hat{X} \leftarrow \text{Median filter}\{X\}$ (3×3 neighborhood)
 - 2: $F(p, q) \leftarrow \hat{X}(M, N)$
 - 3: $F^{HFS} \leftarrow F$
 - 4: *Find number of clusters (Algorithm 3)*
 - 5: $H_F \leftarrow h(f) \in [0, 1]$
 - 6: $v_j \leftarrow \operatorname{rand}([0, 1])$
 - 7: $u_{ij} \leftarrow \frac{1}{\sum_{s=1}^c \left(\frac{d^2(h_i, v_j)}{d^2(h_i, v_s)} \right)^{\frac{2}{(m-1)}}$
 - 8: **while** $\|u_{ij}^a - u_{ij}^{a-1}\| < \varepsilon$ **do**
 - 9: $v_j \leftarrow \frac{\sum_{i=1}^N u_{ij}^m h_i}{\sum_{i=1}^N u_{ij}^m}$
 - 10: $u_{ij} \leftarrow \frac{1}{\sum_{c=1}^C \left(\frac{d^2(h_i, v_j)}{d^2(h_i, v_c)} \right)^{\frac{2}{(m-1)}}$
 - 11: **end while**
 - 12: **return** v_j
-

En este capítulo se abordó la aportación de esta investigación, primero se definió el algoritmo *Hesitant Fuzzy C-Means* y además de describieron tres mejoras con diferentes características, como el uso de la librería de OpenMP para acelerar el tiempo computacional; agregar el índice de Calinski-Harabasz para encontrar automáticamente el número de grupos y una alternativa para inicializar la matriz de pertenencias difusas. En el siguiente capítulo se presenta la experimentación realizada y los resultados obtenidos, tanto para segmentar imágenes como su aplicación en bases de datos del reconocimiento de patrones.

4 | Experimentación y resultados

4.1. Introducción

En este capítulo se describe cada uno de los experimentos realizados para evaluar la calidad del algoritmo propuesto, así como los resultados obtenidos. Estos experimentos están divididos en dos tipos, en el primero se evalúa la convergencia del algoritmo y en el segundo la calidad del agrupamiento. También se definen los conjuntos de datos que se utilizaron, las métricas empleadas para medir los resultados obtenidos y algunos métodos del Estado del Arte con los que se realiza la comparación de los algoritmos propuestos.

4.2. Conjunto de datos

Los tipos de conjuntos de datos están divididos en dos áreas de acuerdo a su aplicación: segmentación de imágenes y reconocimiento de patrones. Para segmentación se utilizaron dos diferentes tipos de imágenes: reales y médicas; éstas fueron seleccionadas dada su complejidad, ya que tienen diferentes colores, texturas y cambios de iluminación; proporcionando así una evaluación más completa. En primera instancia se evalúa el algoritmo en un conjunto de imágenes reales de las siguientes bases de datos, ISIC 2019 (Combalia et al., 2019), contiene más de 13,000 imágenes dermatoscópicas de tamaño 1920 x 1080 píxeles, para la investigación solo se probó en 80 imágenes porque todas son muy parecidas. *Weizmann Segmentation Evaluation* (Winn et al., 2005), con 200 imágenes en color con un tamaño de 300 x 225 píxeles, *Microsoft Research Cambridge Object Recognition Image Database* (MRCORID) (Criminisi et al., 2004) contiene un conjunto de imágenes con un tamaño de 640 x 480 píxeles y *Berkeley Segmentation Data Set 500* (BSDS500) (Martin et al., 2001), con 500 imágenes con un tamaño de 481 x 321 píxeles. La figura 4.1 presenta una imagen de muestra para cada base de datos.

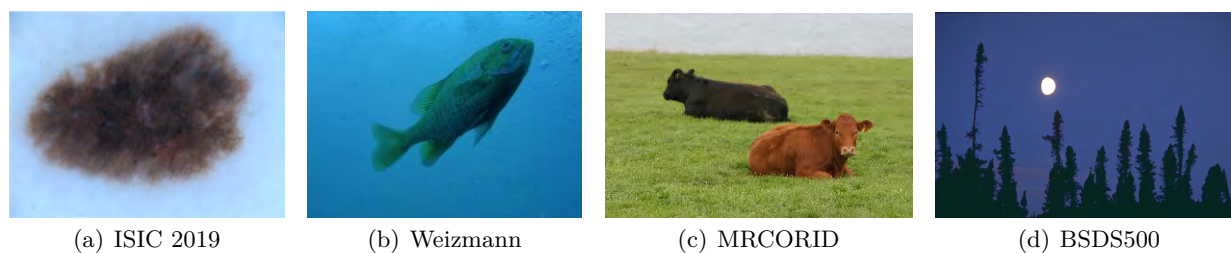


Figura 4.1: Imágenes muestra de cada base de datos.

Para reforzar la evaluación del algoritmo también se utilizó un conjunto de imágenes de glóbulos blancos (WBC) tomado de (Zheng et al., 2018), dos imágenes de muestra se pueden observar en la Figura 4.2. Este conjunto de datos está dividido en dos tipos diferentes de imágenes a color: 300 imágenes de 120×120 píxeles y 100 imágenes de 300×300 píxeles.

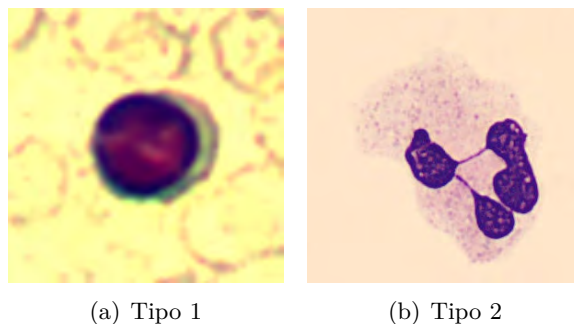


Figura 4.2: Conjunto de imágenes de glóbulos blancos (WBC).

Por otra parte, el algoritmo aplicado al reconocimiento de patrones se evaluó en distintas bases de datos reales obtenidas del repositorio de la UCI Dua and Graff (2017), las cuales se resumen en la Tabla 4.1, con el número de instancias, características y las clases en las que se agrupan.

Tabla 4.1: Bases de datos reales.

Base de datos	Instancias	Características	Clases
Lung-cancer	32	56	3
Iris	150	4	3
Wine	178	13	3
Seeds	210	7	3
Ecoli	336	7	8
Breast Cancer Wisconsin Diagnostic (WDBC)	569	30	2
Yeast	1484	8	10
Banknote authentication (Banknote)	1372	5	2
Image Segmentation (ImgSeg)	2310	19	8

4.3. Métricas

Las métricas utilizadas para evaluar al algoritmo se dividen en dos tipos, las primeras evalúan la calidad de la segmentación de las imágenes, el segundo tipo evalúa la calidad del agrupamiento aplicadas al reconocimiento de patrones. Las métricas consideradas para medir el rendimiento de la segmentación para las bases de datos de ISIC2019, WBC, Weizmann y MRCORID son *Accuracy* (Ecuación 4.1), para medir la calidad de la agrupación. Coeficiente de similitud DICE (Ecuación 4.2), cuantifica la superposición entre los resultados de segmentación con el *ground truth*. Intersección sobre unión (Ecuación 4.3), para evaluar qué tan similar es el resultado de la segmentación con el *ground truth* (Ling et al., 2016).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.1)$$

$$DSC = 2 \bullet \frac{\text{Área}(X \cap Y)}{\text{Área}(X) + \text{Área}(Y)} \quad (4.2)$$

$$IOU = \frac{TP}{TP + FP + FN} \quad (4.3)$$

Donde TP son los verdaderos positivos, TN los verdaderos negativos, FP los falsos positivos y FN los falsos negativos. X y Y representan el *ground truth* y las imágenes segmentadas, respectivamente.

Además, para tener una mejor evaluación de los resultados en la base de datos BSD500, las métricas consideradas para medir el desempeño de la segmentación son Índice Probabilístico *Rand* (PRI), (Unnikrishnan et al., 2007) que mide la similitud entre las imágenes segmentadas y el *ground truth*. Variación de información (VOI), (Meilă, 2007), compara particiones agrupadas, dando la medida de la distancia entre dos clústers, midiendo la cantidad de información perdida o ganada al cambiar de una partición a la otra. Error de consistencia global (GCE), (Martin et al., 2001), evalúa hasta qué punto se puede considerar que la segmentación perfecciona las demás. Error de desplazamiento de límite (BDE), (Freixenet et al., 2002), que evalúa el error de desplazamiento de pixel promedio en los bordes entre dos imágenes segmentadas calculando la distancia entre el pixel y el pixel más cercano en la otra segmentación. Matemáticamente, estas métricas se dan en las siguientes expresiones (4.4), (4.5), (4.6) y (4.7), respectivamente.

$$PRI(S, G_k) = \frac{2}{N(N-1)} \sum_{i,j,i < j} (p_{ij}^{c_{ij}} (1 - p_{ij})^{1-c_{ij}}) \quad (4.4)$$

donde S es el resultado de segmentación del algoritmo probado, N es el número de píxeles, c_{ij} es una función booleana indicando si $l_i^S = l_j^{G_k}$, p_{ij} es el valor esperado de una distribución de *Bernoulli* para el par de píxeles, $l_i^{G_k}$ es el pixel de etiqueta x_i en la imagen k th segmentada manualmente, l_i^S es el pixel de etiqueta x_i en el algoritmo probado. $\{G_1, G_2, \dots, G_L\}$ es el *ground truth*, donde L es el número de imágenes segmentadas manualmente. Las métricas de VOI se definen por:

$$VOI(S, G_k) = H(S) + H(G_k) - 2I(S, G_k) \quad (4.5)$$

donde H es la entropía $-\sum_{i=1}^c \frac{n_i}{n} \log \frac{n_i}{n}$, n_i representa el número de puntos que pertenecen a i -ésimo grupo, $I(S, G_k) = \sum_{i=1}^c \sum_{j=1}^c \frac{n_{i,j}}{n} \log \frac{n_{i,j}}{n} \frac{n_i}{n} \frac{n_j}{n}$ es la información mutua entre dos agrupaciones, $n_{i,j}$ es el número de puntos en la intersección del clúster i de S y j de G_k . El índice GCE se da como:

$$GCE(S, G_k) = \frac{1}{n} \min \left(\sum_{i=1}^n E(S, G_k, x_i), \sum_{i=1}^n E(G_k, S, x_i) \right) \quad (4.6)$$

donde $E(S, G_k, x_i) = \frac{|R(S, x_i) \setminus R(G_k, x_i)|}{|R(S, x_i)|}$ es una medida de error de cada pixel x_i , $|\cdot|$ es la cardinalidad, \setminus es la diferencia del conjunto y $R(S, x_i)$ es el conjunto de píxeles correspondiente a la región de segmentación S que contiene el pixel x_i . Finalmente el BDE está obteniendo por

$$BDE(S, G_k) = \frac{1}{2} (D_S^{G_k} + D_{G_k}^S) \quad (4.7)$$

donde $D_S^{G^k}$ es una distribución de distancia obtenida sumando las distancias sobre todos los puntos de S .

Por su parte, las métricas consideradas para evaluar la calidad del agrupamiento en bases de datos de reconocimiento de patrones se describen a continuación:

- Índice *Rand* ajustado (ARI), calcula una medida de similitud entre dos grupos considerando todos los pares de muestras y contando los pares que se asignan en el mismo grupo o en grupos diferentes entre el resultado del agrupamiento y su *ground truth* (Hubert and Arabie, 1985; Santos and Embrechts, 2009).

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \quad (4.8)$$

donde RI es la versión no ajustada del Índice *Rand* y se describe mediante:

$$RI = \frac{a + b}{C_2^n} \quad (4.9)$$

Esta métrica toma valores entre 0 y 1, los valores más altos indican un mejor agrupamiento.

- *Accuracy* (Acc), descubre la relación uno a uno entre grupos y clases. Sean li e \hat{l}_i el resultado de la agrupamiento y la etiqueta del *ground truth* de xi , respectivamente (Kang et al., 2019).

$$Acc = \frac{\sum_{i=1}^n \delta(\hat{l}_i, \text{map}(l_i))}{n} \quad (4.10)$$

donde n es el número de muestras, la función delta $\delta(x, y)$ es igual a uno si y solo si $x = y$ y cero en caso contrario, $\text{map}(\hat{u})$ es una función de mapeo de permutación que relaciona cada índice del clúster con una etiqueta.

- Información mutua normalizada (NMI) mide la calidad del agrupamiento (Kang et al., 2019).

$$NMI(L, \hat{L}) = \frac{\sum_{l \in L, \hat{l} \in \hat{L}} p(l, \hat{l}) \log\left(\frac{p(l, \hat{l})}{p(l)p(\hat{l})}\right)}{\max(H(L), H(\hat{L}))} \quad (4.11)$$

donde $p(l)$ e $p(\hat{l})$ representan las funciones de distribución de probabilidad marginal de L y \hat{L} , respectivamente, inducidas a partir de la distribución conjunta $p(l, \hat{l})$ de L y \hat{L} . $H(\hat{u})$ es la función de entropía. El valor mayor de NMI significa el mejor rendimiento de agrupación.

- Pureza (*Purity*), mide el grado en que los grupos contengan una sola clase.

$$Pureza = \sum_{i=1}^c \frac{n_i}{n} P(C_i), \quad P(S_i) = \frac{1}{n_i} \max_j(n_i^j) \quad (4.12)$$

donde n_i es el número de puntos en el grupo C_i y n_i^j representa el número total de puntos que el i -ésimo grupo es asignado a la j -ésima categoría. Hay c categorías en total y una pureza mayor indica un mejor rendimiento en la agrupación.

4.4. Métodos de comparación

Para la evaluación del rendimiento de los algoritmos propuestos en esta investigación, se seleccionaron de la literatura únicamente métodos basados en agrupamiento; esto con la finalidad de realizar una comparación adecuada y justa, es decir, utilizar solamente algoritmos dentro de la misma categoría, y no sean de otro enfoque. Estos algoritmos se describen a continuación, además se presenta el pseudocódigo de cada uno.

- **AKFCMS** (*Adaptive kernel-based FCM with spatial constraints*) (Hu and Du, 2019): este algoritmo está basado en el agrupamiento clásico *Kernelized Fuzzy C-Means* (KFCM) y utiliza información espacial para introducir un coeficiente de ponderación adaptativo para controlar la influencia de los píxeles vecinos a los centrales y mejorar la calidad de la agrupación. El Algoritmo 6 describe el pseudocódigo para la segmentación de imágenes, en el que se deben establecer el número de grupos c , la condición de convergencia ε , el total de píxeles vecinos en la ventana de filtrado D_k y el centroide inicial x_i . Después se tienen que calcular las medias de ponderación adaptativas para la imagen filtrada \bar{y}_i , el siguiente paso es un proceso iterativo para calcular la mejor función de pertenencia \bar{z}_{ik} y minimizar la función objetivo usando la mejor función de pertenencia y el centroide, este proceso termina hasta que se cumpla el criterio de parada $\max_i |x_{i+1} - x_i| < \varepsilon$. Finalmente, la segmentación se completa de acuerdo a los mejores grados de pertenencias.

Algorithm 6 Adaptive kernel-based FCM with spatial constraints

Input: Image f , number of clusters c

Output: Clustering segmentation

- 1: Set ε , D_k and x_i
 - 2: Compute $\bar{y}_i \leftarrow \frac{\sum_{j=1}^{D_k} m_j \times y_i}{\sum_{j=1}^{D_k} m_j}$
 - 3: **repeat**
 - 4: Compute $\bar{z}_{ik} \leftarrow \frac{\frac{\alpha}{S_r} \sum_{r \in R_k} \bar{z}_{ir} - \sum_{i=1}^c \frac{1}{(1-K(y_k, x_i)) + \alpha}}{(1-K(y_k, x_i)) + \alpha}$
 - 5: Minimize $O_{KFCMS} \leftarrow \sum_{i=1}^c \sum_{k=1}^D (K(y_k, x_i) \cdot \bar{z}_{ik})^p (1-K(y_k, x_i)) + \frac{\alpha}{S_r} \sum_{i=1}^c \sum_{k=1}^D (K(y_k, x_i) \cdot \bar{z}_{ik})^p \sum_{r \in R_k} (1-K(y_k, x_i))$
 - 6: **until** $\max_i |x_{i+1} - x_i| < \varepsilon$
-

- **FRFCM** (*Significantly Fast and Robust Fuzzy C-Means Clustering Algorithm Based on Morphological Reconstruction and Membership Filtering*) (Lei et al., 2018a): es una mejora al algoritmo FCM que emplea la reconstrucción morfológica para suavizar las imágenes y mejora la inmunidad al ruido preservando los detalles de la imagen, además utiliza un filtrado que modifica la pertenencia de los píxeles sin depender del cálculo de la distancia entre píxeles vecinos y los centros de los grupos. Su pseudocódigo se detalla en el Algoritmo 7, donde f es la imagen, y c el número de grupos, el parámetro de fuzzificación m , el tamaño de la ventana de filtrado ω y el umbral de error mínimo η . Se debe calcular la nueva imagen ξ por reconstrucción morfológica y su histograma, el siguiente paso es inicializar aleatoriamente la matriz de partición de pertenencia $U^{(0)}$ y el contador de bucle $t = 0$. Los centroides y la matriz de partición de pertenencia $U^{(t+1)}$ se actualizan en un proceso iterativo hasta que se cumpla el criterio de paro $\max\{U^{(t)} - U^{(t+1)}\} < \eta$. Finalmente, se implementa el filtro de mediana en la matriz de particiones de pertenencia U' .

Algorithm 7 Fast and Robust Fuzzy C-Means (FRFCM)**Input:** f, c **Output:** Y

- 1: Set m, ω, η and $t = 0$
- 2: Compute $\xi \leftarrow R^c(f)$ and its histogram
- 3: $U^{(0)} \leftarrow \text{rand}([0, 1])$.
- 4: **repeat**
- 5: Update $v_k \leftarrow \frac{\sum_{l=1}^q \gamma_l u_{kl}^m \xi_l}{\sum_{l=1}^q \gamma_l u_{kl}^m}$.
- 6: Update $U^{(t+1)}$ with $u_{kl} \leftarrow \frac{\|\xi_l - v_k\|^{(-2/(m-1))}}{\sum_{j=1}^c \|\xi_l - v_j\|^{(-2/(m-1))}}$
- 7: **until** $\max\{U^{(t)} - U^{(t+1)}\} < \eta$
- 8: $U'' \leftarrow \text{med}\{U'\}$

- DSFCM_N (*Deviation-sparse Fuzzy C-Means with neighbor information constraint*) (Zhang et al., 2018b): este algoritmo depende de las desviaciones entre los valores medidos y los valores teóricos, además introduce información espacial para tener mejores resultados de agrupación al considerar que el uso de valores teóricos de datos es más confiable que simplemente considerar los datos en sí. A continuación, se describe en el Algoritmo 8 el pseudocódigo que como entrada tiene al conjunto de datos $X = \{x_1, x_2, \dots, x_n\}$, el número de grupos c y el parámetro λ es un vector de regularización, para datos de un solo canal genera un escalar. Se debe establecer el factor difuso m , el umbral de convergencia ε , N_j es el tamaño de la ventana local, la matriz de pertenencia inicial aleatoria $U^{(0)}$, la matriz de error $E = 0$ y el contador bucle $t = 0$. Después en un proceso iterativo se actualiza el vector de centroides $V^{(t)}$ la matriz de desviación E y la matriz de pertenencias $U^{(t)}$ hasta que se cumpla con el criterio de convergencia $\max|U^{(t)} - U^{(t+1)}| < \varepsilon$ para obtener la imagen segmentada a partir de la última matriz de partición U .

Algorithm 8 Deviation-sparse Fuzzy C-Means with neighbor information constraint**Input:** $X = \{x_1, x_2, \dots, x_n\}, c$ and λ .**Output:** U, V and E

- 1: Set $m, \varepsilon, N_j, U^{(0)} \leftarrow \text{rand}([0, 1]), E = 0$ and $t = 0$.

2: **repeat**

$$3: \quad \text{Update } V^{(t)} \text{ with } v_i^{(t+1)} \leftarrow \frac{\sum_{j=1}^n \left(u_{ij}^{(t+1)}\right)^m \sum_{r \in N_j} \frac{1}{1+d_{rj}} \left(x_r - e_r^{(t)}\right)}{\sum_{r \in N_j} \frac{1}{1+d_{rj}} \sum_{j=1}^n \left(u_{ij}^{(t+1)}\right)^m}$$

$$4: \quad \text{Update } E^{(t)} \text{ with } e_{jq}^{(t+1)} \leftarrow \frac{\text{soft}\left(\sum_{i=1}^c \sum_{r \in N_j} \left(\frac{\left(u_{ir}^{(t+1)}\right)^m \left(x_j 1 - v_{iq}^{(t+1)}\right)}{1+d_{rj}}\right), \frac{\lambda_q/2}{\sum_{r \in N_j} (1+d_{rj})}\right)}{\sum_{i=1}^c \sum_{r \in N_j} \frac{1}{1+d_{rj}} \left(u_{ir}^{(t+1)}\right)^m}$$

- 5: $t \leftarrow t + 1$

$$6: \quad \text{Update } U^{(t)} \text{ with } u_{ij}^{t+1} \leftarrow \left(\frac{\sum_{k=1}^c \left(\frac{\sum_{r \in N_j} \frac{1}{1+d_{rj}} \left\| x_r - e_r^{(t)} - v_i^{(t)} \right\|_2^2}{\sum_{r \in N_j} \frac{1}{1+d_{rj}} \left\| x_r - e_r^{(t)} - v_k^{(t)} \right\|_2^2} \right)^{\frac{1}{m-1}}}{2} \right)^{-1}$$

- 7: **until** $\max|U^{(t)} - U^{(t+1)}| < \varepsilon$

- AFCCF (*Automatic fuzzy clustering framework*) (Lei et al., 2019b): Basándose en esta idea, el algoritmo AFCCF utiliza superpíxeles para mejorar el algoritmo *Density Peak* (DP), considerando también información espacial para obtener una segmentación de manera efectiva. Los pasos detallados del AFCCF se resumen en el Algoritmo 9 y está dividido en tres partes, pre-procesamiento de superpíxeles, algoritmo DP mejorado (Algoritmo 1 en (Lei et al., 2019b)) y PEFC (*Prior Entropy-based fuzzy clustering algorithm*). El primer paso es calcular los superpíxeles de la imagen X , donde cada área se indica con el valor medio, $\hat{x}_l = \frac{1}{S_l} \sum_{p \in \partial_l} x_p$. Es necesario implementar el algoritmo DP en \hat{x} para obtener γ_j , también el Algoritmo 1 de Lei et al. (2019b) para obtener ϕ_l y el número de grupos. Los valores iniciales de las variables $U^{(0)}$, $V^{(0)}$, $\Sigma^{(0)}$ y $\pi^{(0)}$ se obtienen utilizando el algoritmo FCM con $m = 2$, $\varepsilon = 10^{-5}$ y 50 iteraciones como máximo. Finalmente, en un proceso iterativo se actualiza la matriz de pertenencias $U^{(t)}$, los centros de los grupos $V^{(t)}$, la covarianza $\Sigma^{(t)}$, y la probabilidad $\pi^{(t)}$, hasta cumplir con el criterio de parada. Como resultado, se obtiene un conjunto de datos para clasificar $\hat{x} = \hat{x}_1, \hat{x}_2, \dots, \hat{x}_{N'}$ y un histograma de color correspondiente $S = S_1, S_2, \dots, S_{N'}$, donde S_l denota el número total de píxeles en el área de superpíxeles ∂_l .

Algorithm 9 Automatic fuzzy clustering framework

Input: X .

Output: $\hat{x} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{N'}\}$ and $S = \{S_1, S_2, \dots, S_{N'}\}$

- 1: Compute $\hat{x}_l = \frac{1}{S_l} \sum_{p \in \partial_l} x_p$
 - 2: Obtain γ_j from \hat{x}
 - 3: Obtain ϕ_j and c
 - 4: Initialize $U^{(0)}$, $V^{(0)}$, $\Sigma^{(0)}$ and $\pi^{(0)}$ with FCM algorithm
 - 5: $t = 0$
 - 6: **repeat**
 - 7: Update $U^{(t)}$ with $u_{kl} \leftarrow \frac{\pi_k \exp(-\Phi(\frac{1}{S_l} \sum_{p \in \partial_l} x_p | v_k, \Sigma_k))}{\sum_{h=1}^c \pi_h \exp(-\Phi(\frac{1}{S_l} \sum_{p \in \partial_l} x_p | v_h, \Sigma_h))}$
 - 8: Update $V^{(t)}$ with $v_k \leftarrow \frac{\sum_{l=1}^{N'} u_{lk} \sum_{p \in \partial_l} x_p}{\sum_{l=1}^{N'} u_{lk} S_l}$
 - 9: Update $\Sigma^{(t)}$ with $\Sigma_k \leftarrow \frac{\sum_{l=1}^{N'} S_l u_{kl} ((\frac{1}{S_l} \sum_{p \in \partial_l} x_p) - v_k)^T ((\frac{1}{S_l} \sum_{p \in \partial_l} x_p) v_k)}{\sum_{l=1}^{N'} S_l u_{kl}}$
 - 10: Update $\pi^{(t)}$ with $\pi_k \leftarrow \frac{\sum_{l=1}^{N'} u_{kl} S_l}{\sum_{l=1}^{N'} S_l}$
 - 11: **until** $\max\{U^{(t)} - U^{(t+1)}\} < 10^{-5}$
-

- RSSFCA (*Robust self-sparse fuzzy clustering algorithm*) (Jia et al., 2020): este algoritmo incorpora un término de regularización a su función objetivo para equilibrar la escasez de pertenencias y la falta de claridad, y logra un agrupamiento difuso auto-disperso. Además emplea el CCF-ADB (*connected-component filtering based on area density balance*) para lograr una fusión efectiva de regiones pequeñas, lo que conduce a excelentes resultados de segmentación de imágenes. En el Algoritmo 10 se describe detalladamente cada uno de los pasos de SSFCA. Se requiere establecer el número de grupos c , el parámetro de regularización γ , el parámetro de convergencia η y el número máximo de iteraciones T . Para reducir el número de iteraciones, la matriz de pertenencias $U^{(0)}$, los centros de los grupos $V^{(0)}$ y la matriz de covarianza $\Sigma^{(t)}$ son inicializadas con el algoritmo FCM. Después, estas matrices

deben ser actualizadas junto con la función objetivo $\tilde{J}'^{(t)}$ hasta cumplir con el criterio de convergencia. La salida de este algoritmo es la entrada del Algoritmo 11 donde se van a unir las regiones mas pequeñas para obtener la imagen segmentada por el algoritmo RSSFCA.

Algorithm 10 Self-sparse fuzzy clustering algorithm

Input: X .

Output: $\hat{x} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{N'}\}$ and $S = \{S_1, S_2, \dots, S_{N'}\}$

- 1: Set c, γ, η and T
 - 2: Initialize $U^{(0)}, V^{(0)}, \Sigma^{(0)}$ and $t = 1$
 - 3: **repeat**
 - 4: Update $U^{(t)}$ with $\tilde{J}'_j = \min \|u_{ij} - h_{ij}\|^2$
 - 5: Update $V^{(t)}$ with $v_i = \frac{\sum_{j=1}^n u_{ij} x_j}{\sum_{j=1}^n u_{ij}}$
 - 6: Update $\Sigma^{(t)}$ with $\Sigma_i = \frac{\sum_{j=1}^n u_{ij} (x_j - v_i)^T (x_j - v_i)}{\sum_{j=1}^n u_{ij}}$
 - 7: Update $\tilde{J}'^{(t)} = \sum_{i=1}^c \sum_{j=1}^n u_{ij} \Phi'(x_j | v_i, \sigma_i) + \gamma \sum_{i=1}^c \sum_{j=1}^n u_{ij}^2$
 - 8: **until** $\max |\tilde{J}'^{(t)} - \tilde{J}'^{(t-1)}| \leq \eta$ or $t \geq T$
-

Algorithm 11 CCF-ADB

Input: A connected-component image generated by the SSFCA.

Output: A labeled image containing fewer regions than the input image

- 1: Initialize $\alpha_q, K = 1000$ and $\varepsilon = 0.1$
 - 2: $\varphi_q = \begin{cases} 1, & \|\alpha_q - X_p\| \leq \varepsilon \\ 0, & \text{otherwise} \end{cases}$
 - 3: $\xi_p = \sum_{q=1}^Q \varphi_q$
 - 4: $\kappa_q = \sum_{X_p \leq \alpha_q} \xi_p$
 - 5: Normalization on κ_q to obtain κ'_q .
 - 6: Compute the value of cutoff area M .
 - 7: Merge regions whose areas are smaller than M .
-

- **FCM_SICM** (*Robust Fuzzy C-Means with adaptive spatial \mathcal{E} intensity constraint and membership linking*) (Wang et al., 2020): este algoritmo es un FCM mejorado con restricción espacial e intensidad adaptativa y enlace de miembros. Primero se utiliza un filtro bilateral rápido para adquirir información espacial y de intensidad local; después, se emplea la imagen de diferencia absoluta entre la imagen original y la imagen filtrada bilateralmente, y el recíproco de la imagen de diferencia y la propia imagen de diferencia restringen el FCM convencional, así como la información espacial y de intensidad local, respectivamente; finalmente, para reducir los pasos de iteración, vincula la pertenencia sumando todos los grados de pertenencia dentro de cada cluster calculado en la iteración anterior en forma logarítmica cuadrada y considerándolo como denominador de la función objetivo. En el Algoritmo 12 se resumen el funcionamiento su funcionamiento.

Algorithm 12 FCM_SICM

Input: Image X , cluster numbers K , weighting exponent on membership degree m , minimum error ε , small number $eps = 0.000001$, geometric spread σ_d and photometric spread σ_r

Output: Membership degrees u_{ij}

- 1: \bar{X} be the normalize X into $[0, 1]$.
- 2: \bar{X}_f be the Filter image with fast bilateral filter with σ_d and σ_r
- 3: Find $\Delta I = X - \bar{X}_f$
- 4: Set center vectors $c_i^{(1)}$ to 0
- 5: Randomly initialize all $u_{ij}^{(1)}$.
- 6: Set $u_{ij}^{(0)} = u_{ij}^{(1)}$
- 7: Set $a = 1$
- 8: **repeat**
- 9: Update $c_i^{(a)} = \frac{\sum_{j=1}^N \left(\frac{1}{\Delta y_j} y_j + \Delta y_j \bar{y}_j\right) (u_{ij}^{(a)})^m}{\sum_{j=1}^N \left(\frac{1 + \Delta y_j^2}{\Delta y_j}\right) (u_{ij}^{(a)})^m}$
- 10: Update $u_{ij}^{(a)} = \frac{1}{\sum_{r=1}^K \left[\frac{\frac{1}{\Delta y_j} \|y_j - c_i^{(a)}\|^2 + \Delta y_j \|\bar{y}_j - c_i^{(a)}\|^2}{\ln^2 \left(\sum_{e=1}^N u_{ie}^{(a-1)} + 1 \right)} \right]^{\frac{1}{m-1}}}$
- 11: Update $J_{\text{FCM_SICM}}^{(a)} = \sum_{i=1}^K \frac{\sum_{j=1}^N \frac{1}{\Delta y_j} (u_{ij}^{(a)})^m \|y_j - c_i^{(a)}\|^2 + \sum_{j=1}^N \Delta y_j (u_{ij}^{(a)})^m \|\bar{y}_j - c_i^{(a)}\|^2}{\ln^2 \left(\sum_{e=1}^N u_{ie}^{(a-1)} + 1 \right)}, \sum_{i=1}^K u_{ij}^{(a)}$
- 12: $a = a + 1$
- 13: **until** $\|J_{\text{FCM_SICM}}^{(a)} - J_{\text{FCM_SICM}}^{(a-1)}\| < \varepsilon, a > 1$
- 14: **return** Membership degrees $u_{ij}^{(a)}$
- 15: pixels are applied in the cluster where membership degree is the highest.

En cuanto al reconocimiento de patrones, la evaluación de la calidad del agrupamiento se realiza con algunos algoritmos básicos de agrupamiento, además de otros recientes de la literatura, estos se en listan a continuación:

- FCM (*Fuzzy C-Means*), (Bezdek et al., 1984): es un algoritmo tradicional en el agrupamiento de datos, se caracteriza por asignarle a cada dato cierto grado de pertenencia con respecto a cada uno de los grupos, de tal forma que cada dato puede pertenecer a distintos grupos de forma simultanea. FCM esta basado en una función objetivo que debe ser minimizada en un proceso iterativo en el que se actualiza la matriz de partición difusa U y los centros de los grupos V hasta cumplir con el criterio de paro, cada uno de estos pasos se describen en el Algoritmo 13.

Algorithm 13 Fuzzy C-Means**Input:** Dataset X , number of clusters c **Output:** U, V

- 1: Set m and ε
- 2: $v_j \leftarrow \text{rand}([0, 1])$
- 3: $u_{ij} \leftarrow \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$
- 4: **while** $\|u_{ij}^l - u_{ij}^{l-1}\| < \varepsilon$ **do**
- 5: $v_j \leftarrow \frac{\sum_{i=1}^N u_{ij}^m x_j}{\sum_{i=1}^N u_{ij}^m}$
- 6: $u_{ij} \leftarrow \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$
- 7: **end while**

- Kmediod, (Park and Jun, 2009): Este algoritmo es una técnica de agrupamiento clásica, es una variante de algoritmo *kmeans*, que en lugar de utilizar la media de los datos como centro de los grupos, usa medoides, es decir, un punto real del grupo en donde la disimilitud media con los demás datos del grupo es mínima, es decir, es el punto central grupo. Su pseudocódigo se detalla en el Algoritmo 14.

Algorithm 14 Kmedoid**Input:** Dataset X , number of clusters $1 < c < N$ **Output:** U, V

- 1: $v_j = \text{rand}([0, 1])$
- 2: **repeat**
- 3: $D_{ik}^{2*} = (x_k - v_{*i})^T (x_k - v_{*i})$
- 4: $x_i^* = \text{argmin}_i (D_{ik}^{2*}); v_i^{(l)} = x_i^*$
- 5: $v_i^{(l)*} = \frac{\sum_{j=1}^{N_i} x_i}{N_i}$
- 6: **until** $\max |v_k^{(l)} - v_k^{(l-1)}| \neq 0$

- HPSOFCM (*A Hybrid Clustering Algorithm Based on Fuzzy C-Means and Improved Particle Swarm Optimization*) (Chen et al., 2014): es un algoritmo de agrupamiento híbrido, basado en FCM y ACL-PSO con peso de inercia exponencial, el algoritmo aprovecha al máximo las ventajas de FCM, EPSO y ACL-PSO. Esta combinación permite a HPSOFCM alcanzar una solución más precisa sin quedar atrapado en el óptimo local. El proceso se detalla en el Algoritmo 15.

Algorithm 15 HPSOFCM**Input:** Dataset X , number of clusters $1 < c < N$ **Output:** $gbest$ and $f(gbest)$ 1: Initialize the parameters population size S , c_1 , c_2 , c_3 , $omega_{max}$, v_{min} , v_{max} , maximum iterations CS , and current iteration $t = 0$.

2: Initialize the population and velocity with

$$Population = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_S \end{bmatrix} \quad \begin{array}{l} X_i = \{C_1, C_2, \dots, C_k\} \\ C_j = \{c_1, \dots, c_d, \dots, c_n\} c_d^{min} < c_d < c_d^{max} \\ i = 1, 2, \dots, S \quad j = 1, 2, \dots, k \end{array}$$

$$Velocity = \begin{bmatrix} V_1 \\ V_2 \\ \dots \\ V_S \end{bmatrix} \quad \begin{array}{l} V_i = \{CV_1, CV_2, \dots, CV_k\} \\ CV_j = \{v_1, \dots, v_d, \dots, v_n\} v_d^{min} < v_d < v_d^{max} \\ i = 1, 2, \dots, S \quad j = 1, 2, \dots, k \end{array}$$

3: Calculate $J_m(U, V, X) = \sum_{i=1}^k \sum_{j=1}^N u_{ij}^m d^2(c_i, x_j)$ 4: For each particle is set to $pbest$ and for the population $gbest$.5: Calculate $PopC(t) = \sum_{i=1}^S (1 - R_i(t)) * X_i^{(t)}$ 6: **while** $t \leq CS$ **do**7: Calculate $c_i = \frac{\sum_{j=1}^N u_{ij}^m x_j}{\sum_{j=1}^N u_{ij}^m} \quad i = 1, \dots, k$ 8: Calculate $J_m(U, V, X) = \sum_{i=1}^k \sum_{j=1}^N u_{ij}^m d^2(c_i, x_j)$ 9: Calculate $pbest$ for each particle with

$$p_i^{(t+1)} = \begin{cases} p_i^{(t)} & \text{if } f(X_i^{(t+1)}) \geq f(p_i^{(t)}) \\ X_i^{(t+1)} & \text{if } f(X_i^{(t+1)}) < f(p_i^{(t)}) \end{cases}$$

10: Calculate $gbest$ with $p_g^{(t+1)} = \underset{p_i}{\operatorname{argmin}} f(p_i^{(t+1)}) \quad 1 \leq i \leq S$ 11: Calculate $PopC$ for the swarm using $PopC(t) = \sum_{i=1}^S (1 - R_i(t)) * X_i^{(t)}$

12: For each particle update the velocity with:

if $P \leq R_t$

update velocity using:

$$v_{id}^{(t+1)} = w(t) \times v_{ij}^{(t)} + c_1 \times r_1 \times (p_{id}^{(t)} - x_{id}^{(t)}) + c_2 \times r_2 \times (\alpha * PopC_d^{(t)} - x_{id}^{(t)})$$

else

update velocity using:

$$v_{id}^{(t+1)} = v_{id}^{(t)} + c_1 \times r_1 \times (p_{id}^{(t)} - x_{id}^{(t)}) + c_2 \times r_2 \times (p_{gd}^{(t)} - x_{id}^{(t)})$$

13: The position of each particle is updated through $x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)}$ 14: $t = t + 1$ 15: **end while**

- KMM (*K-Multiple-Means*), (Nie et al., 2019): este algoritmo agrupa datos con múltiples medias de subgrupos en k grupos específicos. El pseudocódigo se describe en el Algoritmo 16, que tiene como entrada el conjunto de datos $A = [a_1, a_2, \dots, a_m]^T \in R^{m \times d}$, el número de grupos k , el número de subgrupos m , el parámetro de regularización γ que se utiliza para

controlar la escasez de la conexión de los datos a los múltiples subgrupos. Se debe inicializar las múltiples medias A , eligiendo muestras aleatorias de m . El resultado de este algoritmo es el conjunto de datos agrupados en k grupos.

Algorithm 16 K-Multiple-Means

Input: $X \in \mathbb{R}^{n \times d}$, k , m, γ

Output: k clusters

- 1: $A \leftarrow \text{rand}([0, 1])$
 - 2: **repeat**
 - 3: Calculate S by optimal solution to $\min_{S_i} \left\| s_i - \frac{d_i^X}{2\gamma} \right\|_2^2$
 $s.t. s_i \geq 0, s_i^T \mathbf{1} = 1$
 - 4: **while not converge do**
 - 5: Update $F = \begin{bmatrix} U \\ V \end{bmatrix}$, where U and V are $\frac{\sqrt{2}}{2}$ of the leading k left and right singular vectors of $\tilde{S} = D_U^{-\frac{1}{2}} S D_V^{-\frac{1}{2}}$ respectively and $D = \begin{bmatrix} D_U & \\ & D_V \end{bmatrix}$
 - 6: For each i , update the i -th row of S by solving $\min_{s_{ij} \geq 0, s_i^T \mathbf{1} = 1} \left\| s_i + \frac{1}{2\gamma} \tilde{d}_i \right\|_2^2$, where $\tilde{d}_i \in \mathbb{R}^{m \times 1}$
 as a vector with j -th element as $\tilde{d}_{ij} = \left\| x_i - a_j \right\|_2^2 + \lambda \left\| \frac{f_i}{\sqrt{d_i}} - \frac{f_{(n+j)}}{\sqrt{d_{(n+j)}}} \right\|_2^2$
 - 7: **end while**
 - 8: For each j , update the j -th row of A with $a_j = \frac{\sum_{i=1}^n s_{ij} x_i}{\sum_{i=1}^n s_{ij}}$
 - 9: **until converge**
 - 10: The k clusters are on the basis of the connectivity of the bipartite graph.
-

- FCM-ELPSO (*Hybrid Fuzzy Clustering Method Based on FCM and Enhanced Logarithmical PSO*) (Zhang and Ma, 2020): este algoritmo combina a FCM con una versión mejorada de PSO, llamada ELPSO, que adopta una nueva estrategia mejorada de pesos logarítmicos de inercia para proporcionar un mejor equilibrio entre exploración y explotación; además este método utiliza el índice PBM(F) y el valor de la función objetivo como índices de validez del cluster para evaluar el efecto de la agrupación. Este método híbrido introduce ELPSO para el proceso de entrenamiento y utiliza la exploración global de ELPSO para encontrar un prototipo de agrupación inicial adecuado para FCM y la exploración local para evitar caer en el óptimo local y utiliza la rápida convergencia de FCM para mejorar los resultados y el tiempo de convergencia.

Algorithm 17 FCM-ELPSO**Input:** Dataset S , number of clusters C **Output:** The best position $gbest$

- 1: Create a swarm with P particles
 - 2: Initialize parameters for ELPSO including size of population P ; ω_i initial for each particle ($l = 1, 2, 3, \dots, P$); c_{1l} and c_{2l} , m
 - 3: Initialize X_l , V_l , $pbest_l(t)$ and $\omega_l = \omega_i$ initial for each particle ($l = 1, 2, 3, \dots, P$) and $gbest(t)$ for the swarm
 - 4: **repeat**
 - 5: **repeat** ▷ ELPSO
 - 6: Calculate membership matrix U_l of each particle
 - 7: Calculate the criterion J_l of each particle
 - 8: Calculate the $pbest_l(t)$ of each particle
 - 9: Calculate the $gbest(t)$ of the swarm
 - 10: Update velocity of each particle using:

$$V_l(t+1) = \omega(t)V_l(t) + c_1R_{1l}(pbest_l(t) - X_l(t)) + c_2R_{2l}(gbest(t) - X_l(t))$$
 - 11: Update position of each particle using:

$$X_l(t+1) = X_l(t) \oplus V_l(t)$$
 - 12: For each particle ($l = 1, 2, 3, \dots, P$) update $\omega_l(t) = (\ln(2.1 + t))^{(-z)}$
 - 13: Store $gbest(t)$ as $gbest$
 - 14: $t = t + 1$
 - 15: **until** ELPSO termination condition is met (\star).
 - 16: **repeat** ▷ FCM
 - 17: Calculate membership degrees using:

$$u_{ij} = \frac{1}{\sum_{k=1}^C (d_{ij}/d_{ki})^{(2/(m-1))}}$$
 - 18: Calculate cluster prototypes using:

$$\beta_i = \frac{\sum_{j=1}^N (u_{ij})^m s_j}{\sum_{j=1}^N (u_{ij})^m}$$
 - 19: Calculate the $pbest_l(t)$ of each particle;
 - 20: Calculate the $gbest(t)$ of the swarm;
 - 21: store $gbest(t)$ as $gbest$
 - 22: **until** ELPSO termination condition is met ($\star\star$)
 - 23: **until** While stop condition is not met ($\star\star\star$)
 - 24: **return** $gbest$ matrix
- (\star) when 95 iterations are achieved(T_PSO) or the J criterion has a variation lower or equivalent to 0.00001.
- ($\star\star$) when 5 iterations are achieved (T_FCM)or the J criterion has a variation lower or equivalent to 0.00001.
- ($\star\star\star$) when the total number of iterations achieves 500 (ELPSO + FCM) or there is no variation in the $gbest$ in two runs of FCM-PSO in a row (FPSO continued FCM).

4.5. Experimentos

Con el objetivo de evaluar el rendimiento de los algoritmos propuestos en esta investigación, en términos de calidad de agrupamiento y convergencia, se realizaron los siguientes experimentos:

1. *Tiempo de ejecución e iteraciones*: Este experimento se realizó con la finalidad de demostrar la rapidez de la convergencia de los algoritmos y esta dividido en tres secciones:
 - a Comparar el algoritmo secuencial *Hesitant Fuzzy C-Means* (HFCM) y el algoritmo implementado en OpenMP *Parallel Hesitant Fuzzy C-Means* (PHFCM) para evidenciar que el uso de la librería de OpenMP si genera un cambio significativo en el tiempo de ejecución; siendo implementados en bases de datos de segmentación de imágenes reales y médicas.
 - b Comparar el algoritmo *Automatic Hesitant Fuzzy C-Means* (AHFCM) con algunos algoritmos seleccionados de la literatura (FCM, Kmediod, HPSOFCM, FRFCM, KMM, FCM-ELPSO), esta comparación se realizó en distintas bases de datos reales obtenidas del repositorio de la UCI (Dua and Graff, 2017)
 - c Comparar el algoritmo *Fast and Automatic Hesitant Fuzzy Clustering* con algoritmos seleccionados de la literatura, y aplicados a la segmentación de imágenes, específicamente la base de datos BSDS500.
2. *Calidad del agrupamiento*: Analizar la calidad del agrupamiento comparando el algoritmo propuesto con algunos algoritmos recientes de la literatura con el objetivo de demostrar la superioridad para agrupar datos con respecto a otros algoritmos, tanto en bases de datos de segmentación de imágenes como para las de reconocimiento de patrones.

4.6. Resultados

En esta sección se describen los resultados obtenidos de la experimentación del algoritmo propuesto PHFCM, AHFCM y FAHFC; los experimentos son aplicados a las tareas de segmentación de imágenes y reconocimiento de patrones y están divididos en dos secciones principales, que son la evaluación del tiempo de ejecución y la calidad del agrupamiento. El objetivo de esta sección es demostrar que el rendimiento de los algoritmos propuestos es superior al compararlo con otros algoritmos de la literatura.

4.6.1. Tiempo de ejecución

Los resultados de este experimento son presentados en términos del número de iteraciones y el tiempo de ejecución, de cada uno de los algoritmos propuestos (PHFCM, AHFCM y FAHF), la primera parte se realiza una comparación entre el algoritmo HFCM implementado en modo secuencial y *Parallel Hesitant Fuzzy C-Means* que es acelerado mediante OpenMP, aplicado a bases de datos de imágenes reales y médicas; en la segunda parte de esta sección se evalúa el rendimiento del algoritmo AHFCM con otros de la literatura, aplicado a bases de datos del reconocimiento de patrones, y finalmente, la última sección se analiza la ejecución del algoritmo FAHFC aplicado a la segmentación de imágenes reales.

Algoritmo *Parallel Hesitant Fuzzy C-Means*

Los resultados obtenidos para bases de datos de imágenes médicas se resumen en la Tabla 4.2, que muestra el tiempo de ejecución promedio obtenido en las bases de datos ISIC2019 y WBC, se observa que el tiempo se reduce considerablemente con el algoritmo PHFCM, en esta tabla también se destaca que el número de iteraciones es afectado, ya que se reduce considerablemente a cuatro iteraciones, mientras que el secuencial requiere más de ocho para converger.

Tabla 4.2: Rendimiento de ejecución promedio en bases de datos de imágenes médicas.

Base de datos	Algoritmo	Tiempo (s)	Iteraciones
ISIC2019	HFCM	6.273	9
	PHFCM	2.326	4
WBC	HFCM	0.8292	8
	PHFCM	0.1562	4

En cuanto al área de segmentación de imágenes reales, los resultados de este experimento para la bases de datos MRCORID se presentan en la Tabla 4.3, una comparación de estos revela que el algoritmo PHFCM es más rápido que el HFCM secuencial, así como existe una diferencia relevante entre el número de iteraciones para converger entre los dos algoritmos, donde el algoritmo PHFCM converge en menos iteraciones, reduciéndose aproximadamente a la mitad.

Tabla 4.3: Rendimiento de ejecución promedio de la base de datos MRCORID.

Algoritmo	Tiempo (s)	Iteraciones
HFCM	5.465	17
PHFCM	0.811	8

Los resultados de la bases de datos Weizman, se resumen en la Tabla 4.4, en donde se aprecia que el comportamiento del algoritmo PHFCM es superior al del algoritmo secuencial, es decir, es más rápido. Igualmente, en esta tabla se distingue que existe una reducción en cuanto al número de iteraciones requeridas para converger del algoritmo propuesto, que lo hace en un promedio de cinco iteraciones a diferencia del secuencial que necesita de 14.

Tabla 4.4: Rendimiento de ejecución promedio de la base de datos Weizman.

Algoritmo	Tiempo (s)	Iteraciones
HFCM	0.669	14
PHFCM	0.149	5

Por su parte, los resultados obtenidos para la base de datos BSDS500 se presentan en la Tabla 4.5, se observa que el rendimiento del algoritmo PHFCM es superior a la versión secuencial, reduciendo considerablemente el tiempo de ejecución, además también logra converger en menor número de iteraciones.

Tabla 4.5: Rendimiento de ejecución promedio de la base de datos BSDS500.

Algoritmo	Tiempo (s)	Iteraciones
HFCM	1.575	11
PHFCM	0.917	8

Con los resultados de las bases de datos de segmentación de imágenes, la implementación (PHFCM) a beneficiado la convergencia reduciéndose a por lo menos la mitad del número de iteraciones necesarias para converger con respecto al algoritmo secuencial, tal como se ha mencionado este resultado no se esperaba, ya que la paralelización solo debería de afectar el tiempo de procesamiento; sin embargo también logro alterar las iteraciones requeridas para converger. Una posible explicación del porqué se reducen las iteraciones puede deberse al hecho de que en la primera etapa paralela, los datos asignados a la matriz difusa titubeante impactan significativamente para que el algoritmo PHFCM logré converger en menor número de iteraciones, en otras palabras, los datos generados en esa etapa logran que el algoritmo alcance más rápido los centroides finales.

Algoritmo *Automatic Hesitant Fuzzy C-Means*

Los resultados cuantitativos de este experimento se presentan en términos del número de iteraciones y el tiempo de ejecución, la comparación es entre el algoritmo AHFCM y algunos algoritmos seleccionados de la literatura (FCM, Kmediod, HPSOFCM, FRFCM, KMM, FCM-ELPSO). Los resultados obtenidos se presentan en la Tabla 4.6, ésta revela que el algoritmo AHFCM no necesita más de 15 iteraciones para converger a diferencia del algoritmo FCM que requiere más de 50 iteraciones en algunas bases de datos. También se observa que el algoritmo Kmediod converge en menos de 6 de iteraciones, sin embargo, el tiempo de ejecución es superior al del algoritmo propuesto y al de los demás algoritmos; siendo el algoritmo AHFCM el más rápido en este experimento.

Tabla 4.6: Tiempo de ejecución e iteraciones de cada algoritmo para cada base de datos.

Base de datos	Rendimiento	FCM	Kmediod	HPSOFCM	FRFCM	KMM	FCM-ELPSO	AHFCM
Lung-cancer	Tiempo (s)	0.015	0.484	25.255	0.113	0.390	0.004	0.080
	Iteraciones	20	2	70	62	9	17	6
Iris	Tiempo (s)	0.012	0.904	17.290	0.117	0.662	0.058	0.018
	Iteraciones	19	4	70	36	11	12	11
Wine	Tiempo (s)	0.015	0.594	9.697	0.118	0.833	0.009	0.030
	Iteraciones	13	3	70	27	10	9	8
Seed	Tiempo (s)	0.012	0.902	24.427	0.117	0.992	0.010	0.026
	Iteraciones	36	5	70	17	11	11	13
Ecoli	Tiempo (s)	0.097	0.542	205.735	0.260	0.848	0.105	0.013
	Iteraciones	172	2	70	100	7	82	12
WDBC	Tiempo (s)	0.006	1.636	38.141	0.132	3.274	0.062	0.034
	Iteraciones	15	3	70	19	20	11	7
Yeast	Tiempo (s)	0.232	0.564	1355.889	0.363	2.047	0.392	0.019
	Iteraciones	185	2	70	100	14	90	2
Banknote	Tiempo (s)	0.055	1.387	57.084	0.385	4.567	0.071	0.011
	Iteraciones	25	6	70	37	20	23	14
ImgSeg	Tiempo (s)	0.039	0.491	98.253	0.182	0.730	0.030	0.021
	Iteraciones	60	2	70	95	6	37	11

Algoritmo *Fast and Automatic Hesitant Fuzzy Clustering*

Los resultados de este experimento se presentan en la Tabla 4.7, se observa que los algoritmos AFCE y RSSFCA, obtienen el mayor número de iteraciones, esto se debe a que se establecen por defecto al inicio del algoritmo, por lo tanto, el tiempo de ejecución también es alto. Otro dato importante que se observa es que a pesar que el algoritmo FCM es más rápido, el número de iteraciones es mayor. Por su parte el algoritmo propuesto, converge en menor número de iteraciones y en menor tiempo que los demás algoritmos, incluso tiene un mejor rendimiento que el algoritmo PHFCM, esto es debido a la etapa de inicialización en la segunda fase, donde se calculan los centroides iniciales, a través de una reducción de la imagen original, dado que la imagen es de menor tamaño el proceso iterativo para actualizar centroides y matriz de pertenencias se realiza en menor tiempo, los centroides obtenidos en esta etapa son utilizados en la fase 3, dado el acercamiento que tienen a los valores óptimos, hacen al algoritmo converger en menor número de iteraciones, en un promedio de 5.

Tabla 4.7: Rendimiento de ejecución promedio.

Algoritmos	FCM	FRFCM	AFCE	RSSFCA	FCM_SICM	PHFCM	FAHFC
Tiempo (s)	0.015	2.746	2.306	61.994	0.803	0.439	0.387
Iteraciones	20	36	50	50	15	6	5

4.6.2. Calidad del agrupamiento

Los resultados de la evaluación de la calidad del agrupamiento en bases de datos de reconocimiento de patrones, imágenes médicas y reales se comparan a continuación, los resultados presentados contienen el valor promedio de cada métrica con respecto a cada algoritmo propuesto (PHFCM, AHFCM y FAHFCM), así como cada algoritmo utilizado para la comparación. Además, para visualizar el resultado de la segmentación se presentan dos imágenes muestra de cada base de datos, sugiriendo mediante el parámetro c el número de regiones en las que se segmentaron.

Algoritmo *Parallel Hesitant Fuzzy C-Means*

La evaluación del algoritmo PHFCM en bases de datos de imágenes médicas (ISIC2019 y WBC) se presentan en la Tabla 4.8.

Tabla 4.8: Resultado promedio del agrupamiento para bases de datos médicas.

Base de datos	Métricas	AKFCMS	FRFCM	DSFCM_N	AFCE	RSSFCA	PHFCM
ISIC2019	ACC	0.9461	0.9593	0.9561	0.9422	0.9436	0.9601
	DSC	0.9518	0.9628	0.9629	0.9516	0.9552	0.9635
	IOU	0.9103	0.9296	0.9295	0.9088	0.9157	0.9310
WBC	ACC	0.9015	0.9049	0.9135	0.8856	0.9109	0.9171
	DSC	0.9366	0.9390	0.9439	0.9262	0.9404	0.9459
	IOU	0.8870	0.8917	0.9009	0.8676	0.8929	0.9029

En la tabla se observa que el algoritmo es capaz de superar el rendimiento de los algoritmos usados para la comparación, ya que en ambas bases de datos el resultado de las métricas es mayor. Solo para visualizar y corroborar estos resultados, se agrega la Figura 4.3, que muestra dos gráficos de los resultados de cada base de datos con respecto a cada una de las métricas, con esta figura se aprecia la superioridad del algoritmo propuesto sobre los algoritmos con los que se comparó.

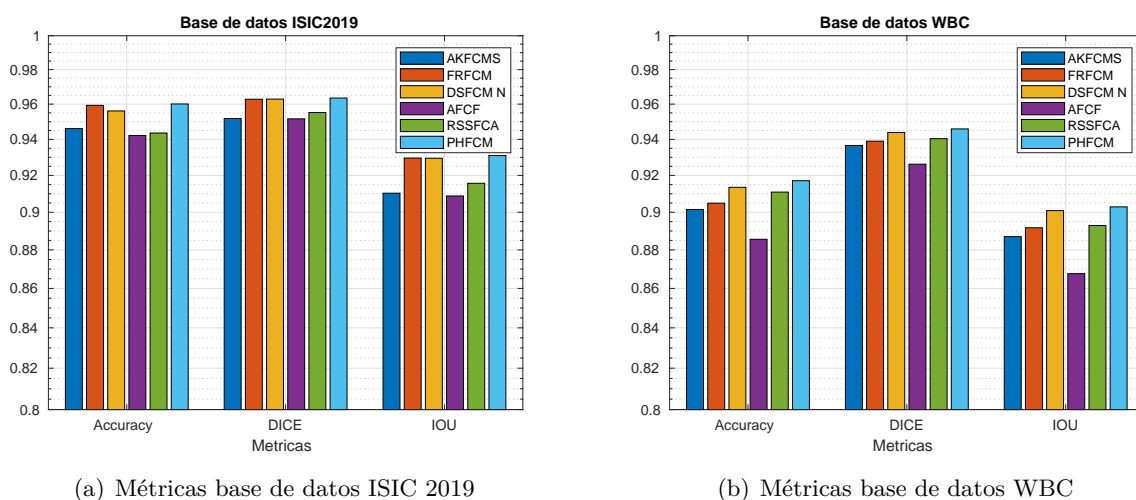


Figura 4.3: Resultados en bases de datos de imágenes médicas.

Solo para ejemplificar, en la Figura 4.4 se ilustran los resultados de dos imágenes de muestra para esta base de datos, cada una es segmentada por los diferentes algoritmos (AKFCMS, FRFCM, DSFCM_N, AFCF, RSSFCA y PHFCM), en estas imágenes se colocó el número de regiones establecidas para segmentar cada imagen sugerida por el parámetro c además también se muestra el *ground truth* de cada una de ellas. Además, para apreciar el contenido de cada imagen, de su *ground truth* y de cada segmentación obtenida por los algoritmos se optó por hacer un aumento a ciertas regiones de cada imagen y así realizar una mejor comparación de los resultados.

En la segmentación de las dos bases de datos de imágenes médicas, el algoritmo propuesto PHFCM es capaz de definir los límites de las regiones, además de generar regiones más compactas, a diferencia de otros algoritmos, como ejemplo está el algoritmo AKFCMS y RSSFCA que no delimitan correctamente las regiones, además de no asignar correctamente todos los píxeles de una región. Una de las complejidades de este tipo de imágenes radica en los límites de las regiones, que no están bien definidos, sino que se van desvaneciendo, y es aquí en donde surge la ventaja del uso de conjuntos difusos titubeantes, ya que permiten tomar en cuenta distintos valores de pertenencia posibles para tomar una mejor decisión al ubicar cada píxel.

La evaluación de la calidad del agrupamiento para la bases de datos MRCORID se presenta en la Tabla 4.9, estos resultados sugieren que el algoritmo propuesto tiene un mejor desempeño en dos de las métricas consideradas, mientras que el resultado promedio de la métrica ACC del algoritmo AFCF es superior al del algoritmo propuesto, sin embargo la ventaja es mínima; una posible explicación a esto se debe al hecho de que el resultado de este algoritmo (AFCF) es superior al del algoritmo propuesto en dos o tres imágenes, como máximo, por lo que al obtener un promedio

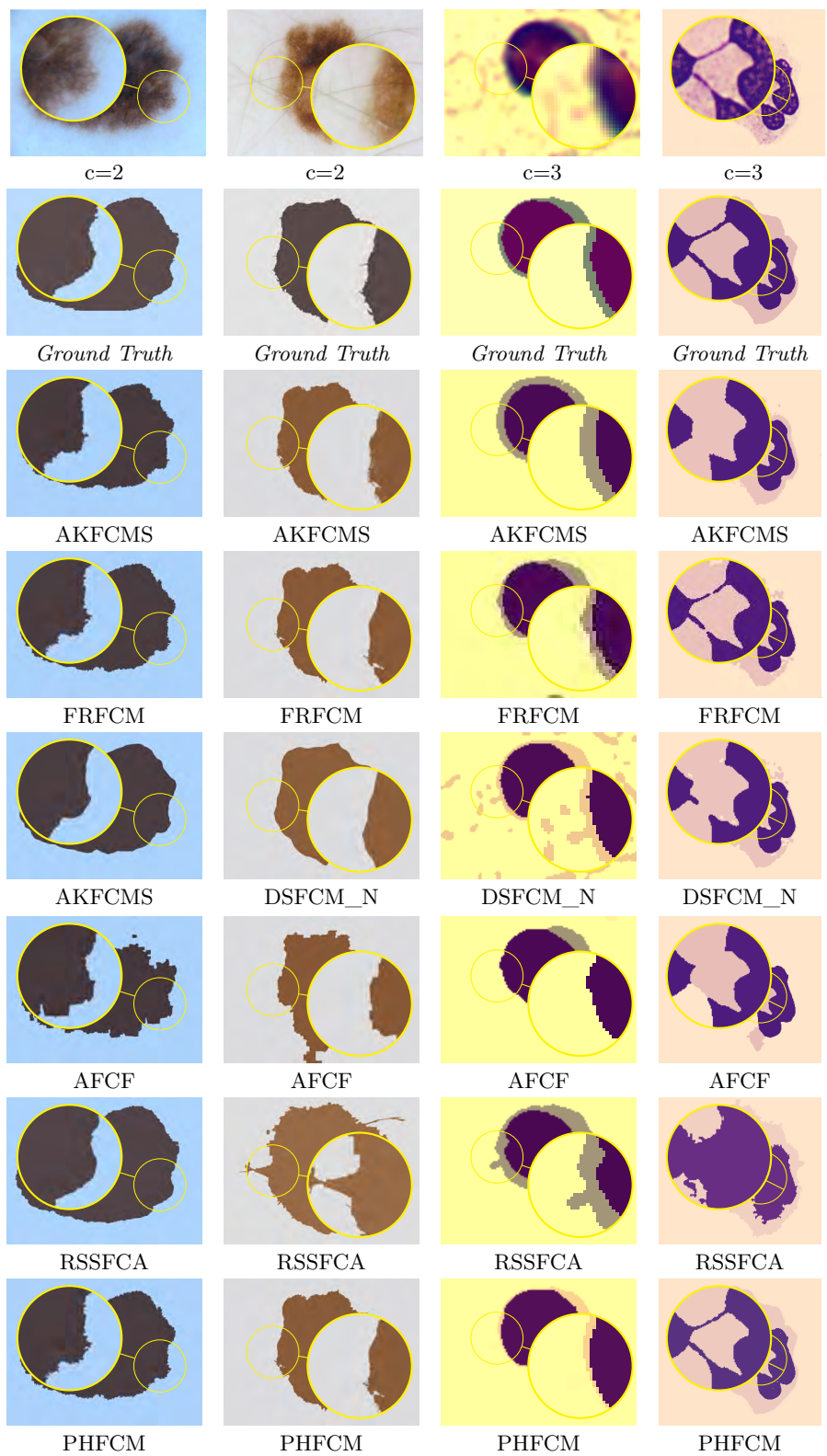


Figura 4.4: Resultado de la segmentación en base de datos médicas (ISIC2019 y WBC).

superan el resultado del algoritmo propuesto. Para visualizar detalladamente esta tabla, se presenta la Figura 4.5 con un gráfico de barras para cada métrica considerada en esta base de datos.

Tabla 4.9: Resultado promedio del agrupamiento para base de datos MRCORID.

Base de datos	Métricas	AKFCMS	FRFCM	DSFCM_N	AFCF	RSSFCA	PHFCM
MRCORID	ACC	0.9280	0.9350	0.8485	0.9356	0.9320	0.9353
	DSC	0.9480	0.9544	0.8240	0.9545	0.9507	0.9546
	IOU	0.9085	0.9202	0.7524	0.9204	0.9131	0.9208

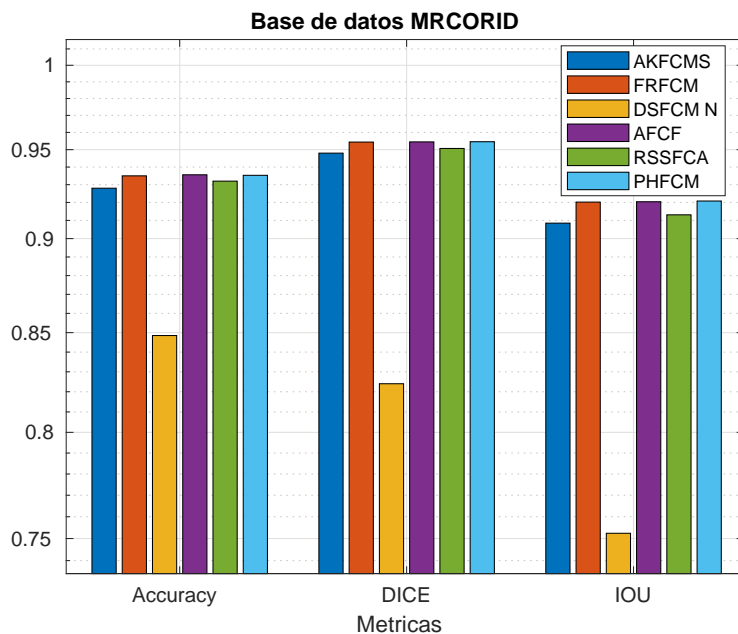


Figura 4.5: Resultados de la base de datos MRCORID.

Por otro lado, la Figura 4.6 ilustra los resultados de dos imágenes de muestra de esta base de datos, estas son segmentadas en tres y cuatro regiones. Una de las principales complejidades para esta base de datos son las texturas que contiene cada imagen. De esta figura se destaca que PHFCM es capaz de agrupar los píxeles en más de dos regiones, además, estas son más homogéneas comparándolas con las que generan los demás algoritmos, como es el caso de AKFCMS que no es capaz de agrupar los píxeles de una misma región. Otro resultado interesante es la superioridad que tiene para delimitar las regiones, a diferencia de los demás, como por ejemplo DSFCM_N que no detalla los límites de las regiones.

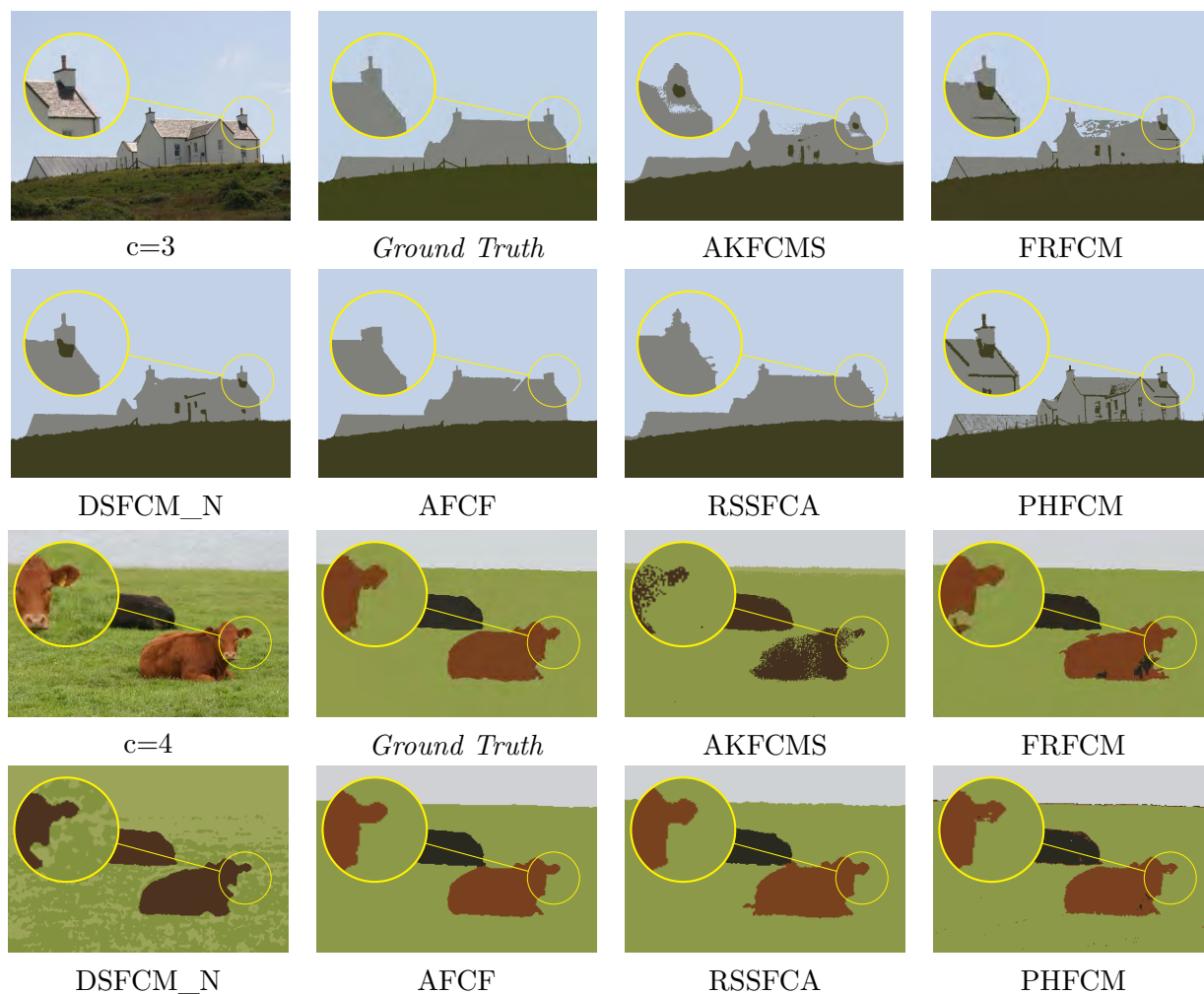


Figura 4.6: Resultado de la segmentación en base de datos MRCORID.

Para esta base de datos Weizman, los resultados cuantitativos de la calidad del agrupamiento se resumen en la Tabla 4.10, donde destaca el algoritmo PHFCM con un rendimiento superior al de los algoritmos con los que se realizó la comparación. Para observar a detalle la superioridad del algoritmo PHFCM se presenta la Figura 4.7.

Tabla 4.10: Resultado promedio del agrupamiento para base de datos Weizman.

Base de datos	Métricas	AKFCMS	FRFCM	DSFCM_N	AFCF	RSSFCA	PHFCM
Weizmann	ACC	0.8792	0.8890	0.7360	0.8976	0.8736	0.9085
	DSC	0.8481	0.8635	0.7157	0.8369	0.7621	0.9021
	IOU	0.7568	0.8067	0.6016	0.7628	0.6757	0.8374

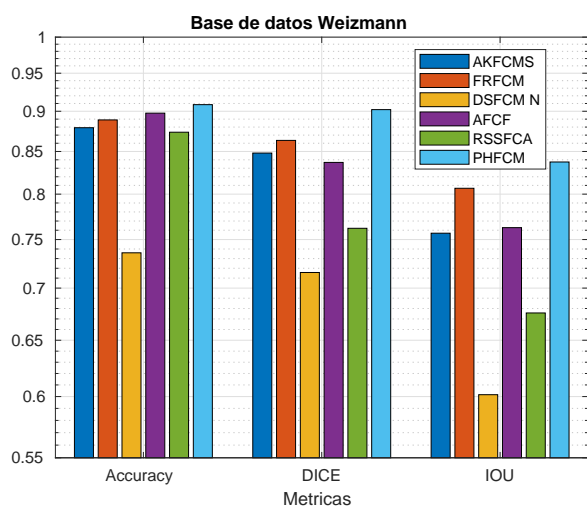


Figura 4.7: Resultados de la base de datos Weizman.

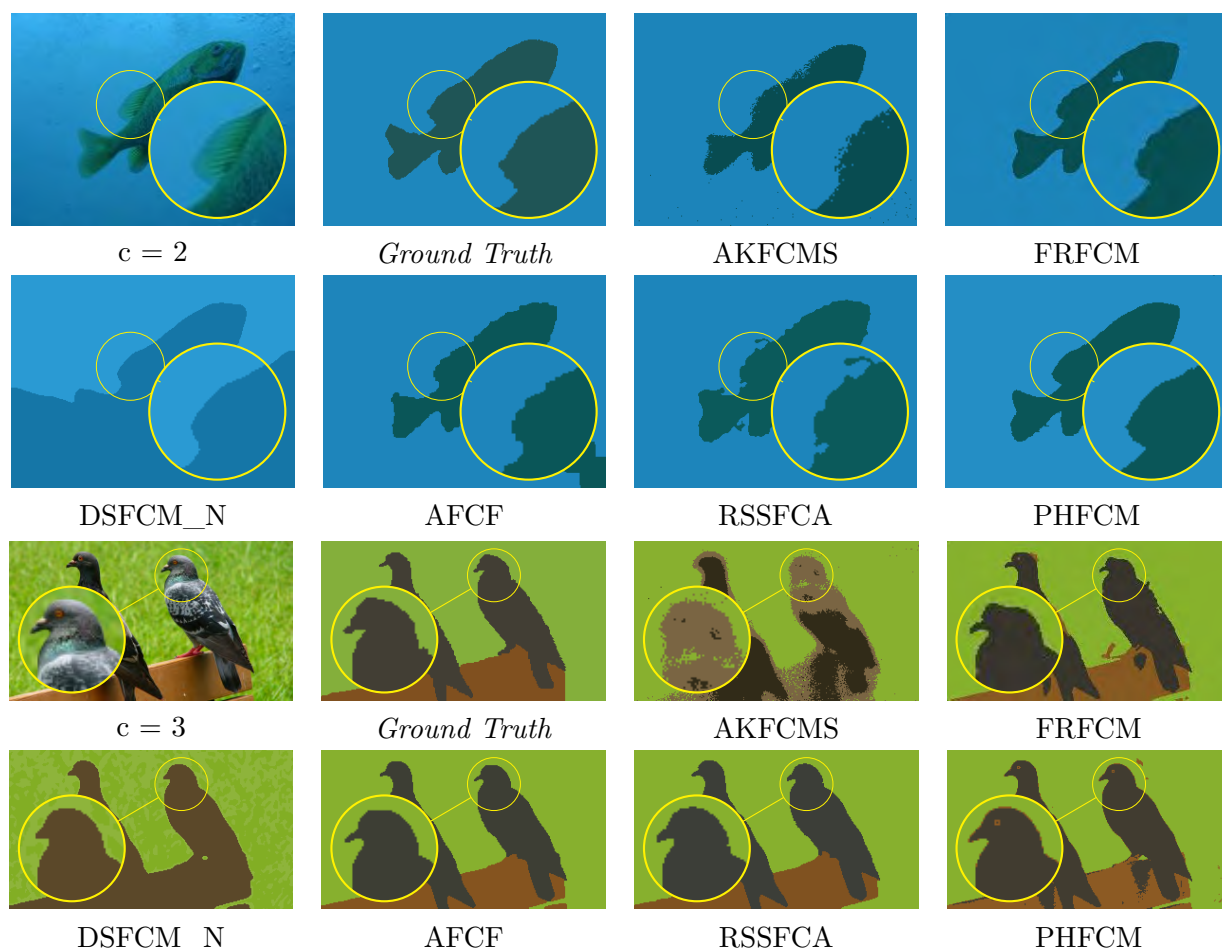


Figura 4.8: Resultado de la segmentación en base de datos Weizmann.

En cuanto a los resultados de la segmentación en esta base de datos se presenta la Figura 4.8, en la que al comparar el algoritmo propuesto se observa que sí es capaz de segmentar adecuadamente las regiones, a diferencia del algoritmo DSFCM_N que no logra agrupar correctamente los píxeles de cada región. También se puede destacar que el algoritmo PHFCM conserva detalles finos en los límites de las regiones, mientras que RSSFCA u AKFCMS no son capaces de preservar esos detalles.

Finalmente, los resultados cuantitativos de los experimentos realizados en la bases de datos BSDS500 se presentan en la Tabla 4.11, donde el algoritmo es capaz de superar a los demás algoritmos con los que se comparó. Sin embargo, el algoritmo RSSFCA obtiene una ventaja mínima sobre el algoritmo PHFCM en la métrica BDE, esto puede deberse a que los resultados presentados son el promedio de todas las imágenes, entonces en un par de imágenes esta métrica dio un mejor resultado que la de PHFCM. Por último, la Figura 4.9 ilustra los resultados de esta tabla en un gráfico de barras para observar detalladamente los resultados cuantitativos.

Tabla 4.11: Resultado promedio del agrupamiento para base de datos BSDS500.

Base de datos	Métricas	AKFCMS	FRFCM	DSFCM_N	AFCF	RSSFCA	PHFCM
BSDS500	PRI	0.8087	0.8944	0.7621	0.8848	0.8469	0.9013
	VOI	1.0883	0.7609	1.2188	0.7719	0.8921	0.7348
	GCE	0.1453	0.1041	0.1645	0.1048	0.1386	0.1030
	BDE	18.4835	10.5203	17.0073	10.8065	9.2459	10.2092

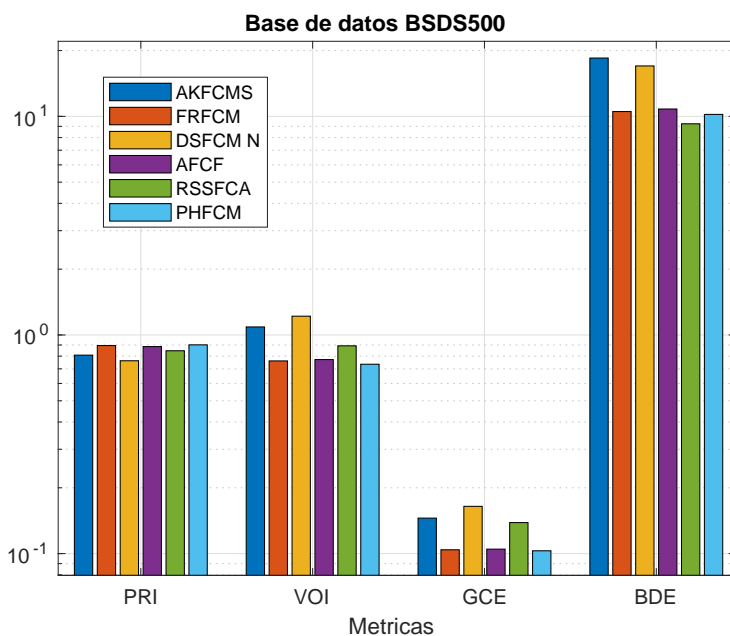


Figura 4.9: Resultados de la base de datos BSDS500.

En la Figura 4.10 se ilustran los resultados de la segmentación en esta base de datos, estas imágenes revelan un rendimiento superior del algoritmo PHFCM sobre los algoritmos utilizados para la comparación, al observar detalladamente el aumento a las regiones se aprecia que las regiones son homogéneas, además los límites están definidos, es decir, el algoritmo propuesto es capaz de definir desde donde empieza y/o termina una región, mientras que algunos no son capaces de segmentar en las regiones indicadas. Un ejemplo de esto se presenta al momento de separar la luna del fondo de la imagen, en donde la mayoría de los algoritmos con los que se realiza la comparación no la segmentan, lo que hacen es considerarla parte del fondo. Es importante destacar la razón por la que se consideró esta base de datos, ya que es compleja debido a que las imágenes tienen diferentes texturas y cambios de iluminación.

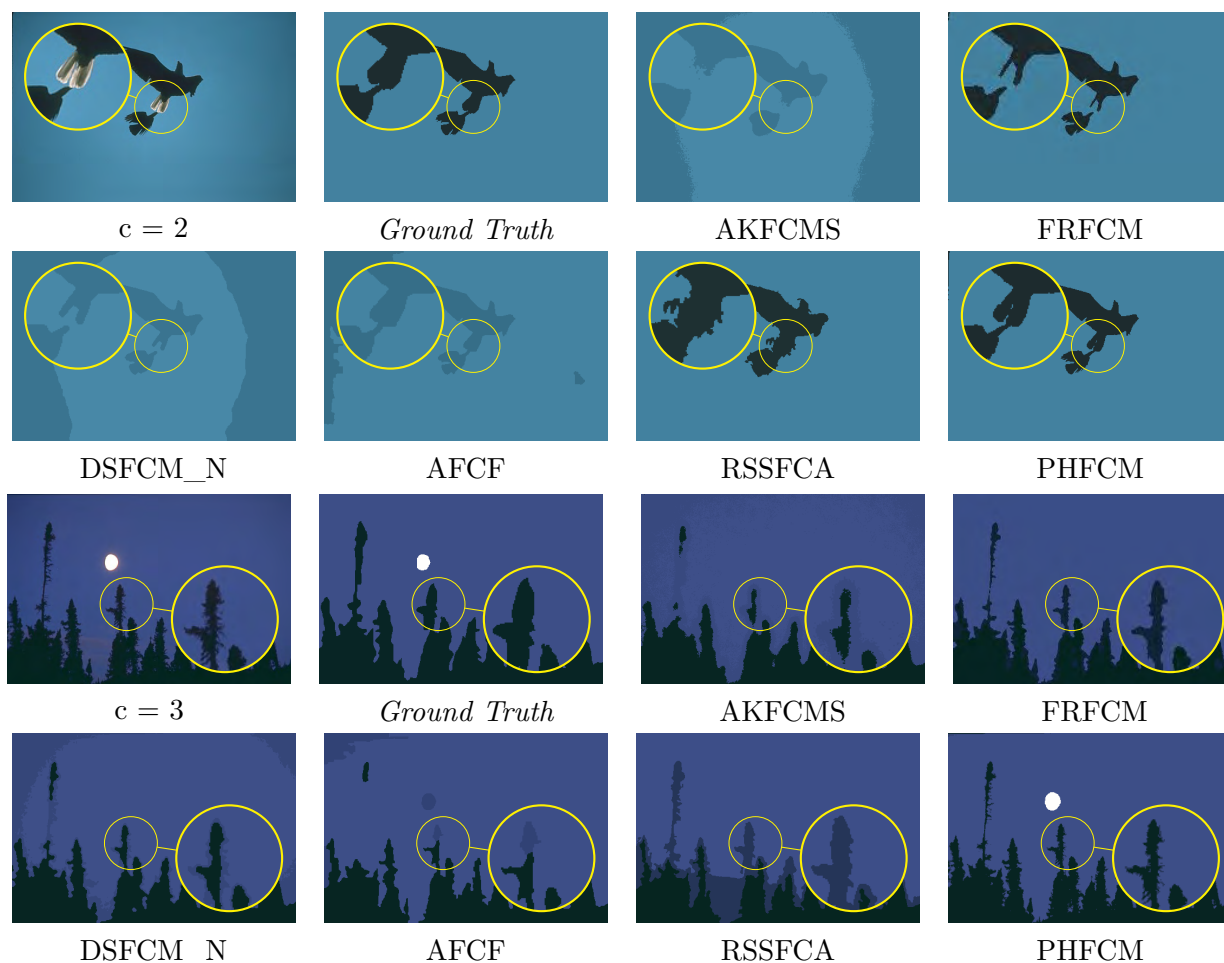


Figura 4.10: Resultado de la segmentación en base de datos BSDS500.

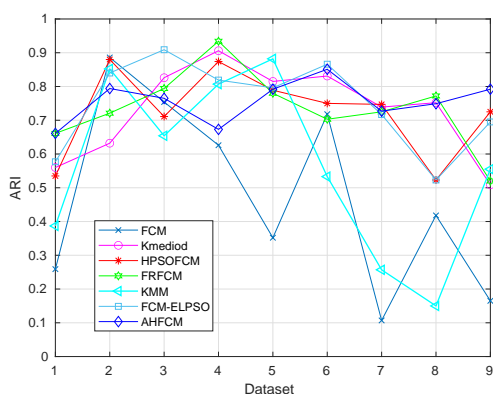
Algoritmo *Automatic Hesitant Fuzzy C-Means*

El valor promedio de cada métrica se resume en la Tabla 4.12, aquí se observa que el rendimiento del algoritmo AHFCM es superior comparado con los demás algoritmos, teniendo una ventaja en las cuatro métricas seccionadas para evaluar la calidad del agrupamiento.

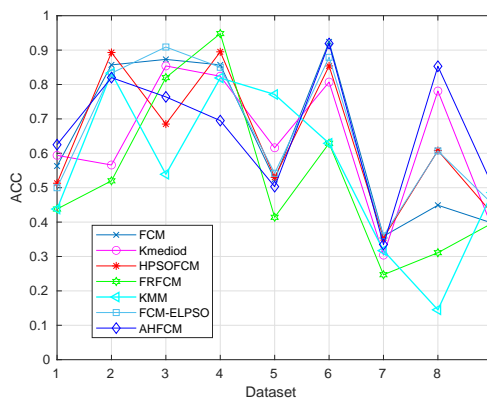
Tabla 4.12: Resultado promediado de cada algoritmo

Métrica	FCM	Kmediod	HPSOFCM	FRFCM	KMM	FCM-ELPSO	AHFCM
ARI	0.476	0.730	0.726	0.734	0.564	0.749	0.756
ACC	0.645	0.636	0.640	0.525	0.553	0.659	0.669
NMI	0.446	0.448	0.397	0.438	0.305	0.430	0.453
Pureza	0.677	0.684	0.681	0.614	0.575	0.709	0.713

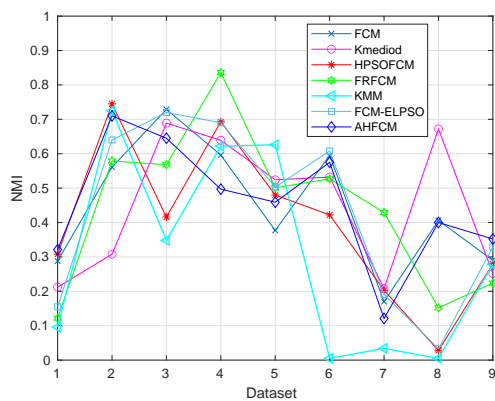
Para visualizar esta evaluación, se presenta la Figura 4.11 con el resultado de cada una de las métricas en cada base de datos; en esta figura se observa que el algoritmo AHFCM obtiene los valores más altos en la mayoría de las bases de datos, corroborando los resultados cuantitativos.



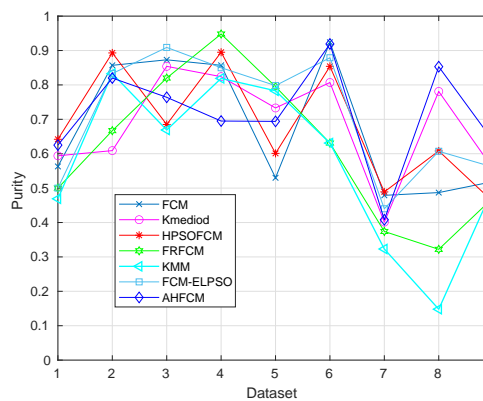
(a) Índice *Rand* ajustado



(b) *Accuracy*



(c) Información mutua normalizada



(d) Pureza

Figura 4.11: Gráficos de los resultados cuantitativos.

Anteriormente, se mencionó que una de las principales aportaciones lograda con este algoritmo es la capacidad de determinar el número de grupos en los que se va a agrupar el conjunto de datos; por lo que es importante mostrar que el índice CH encuentra correctamente el número de grupos para cada conjunto de datos, detallando los resultados obtenidos en la Figura 4.12, en la que se marca con un punto rojo el valor máximo obtenido para cada base de datos.

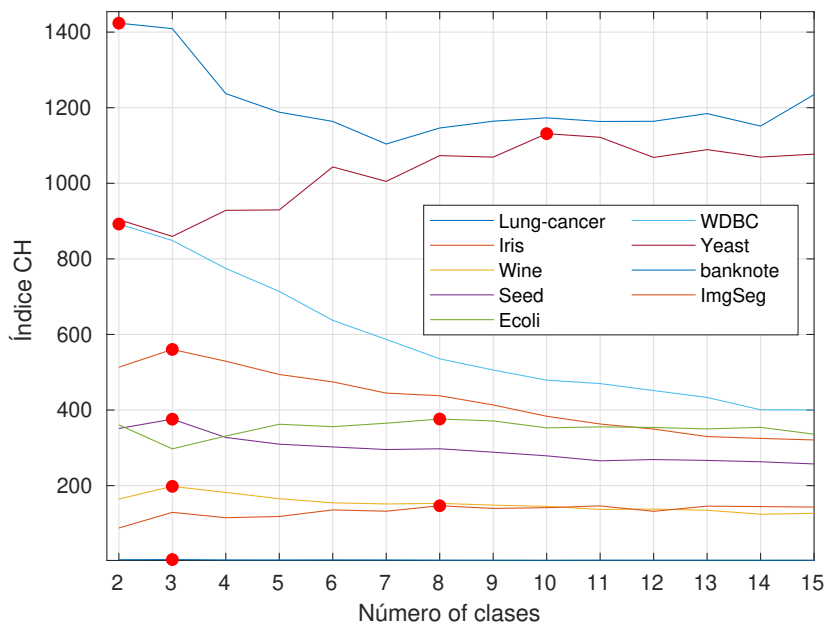


Figura 4.12: Resultado del índice CH.

Algoritmo *Fast and Automatic Hesitant Fuzzy Clustering*

El resultado promedio de cada una de las métricas se resumen en la Tabla 4.13, donde se observa la ventaja del algoritmo propuesto FAHFC sobre los demás algoritmos en la mayoría de las métricas, por ejemplo es superior al algoritmo PHFCM en las métricas PRI y GCE con diferencias de 0.02 y 0.06, respectivamente; mientras que en la métrica VOI tiene una diferencia de 0.12 sobre RSSFCA, solo para visualizar los datos de la tabla también se presenta el gráfico de la Figura 4.13.

Tabla 4.13: Resultado promedio de la segmentación de imágenes.

Algoritmos	PRI	VOI	GCE	BDE
FCM	0.769	1.147	0.210	9.563
FRFCM	0.833	0.855	0.150	7.969
AFCF	0.868	0.735	0.132	7.113
RSSFCA	0.875	0.663	0.106	9.135
FCM_SICM	0.802	0.966	0.181	14.388
PHFCM	0.901	0.735	0.103	10.209
FAHFC	0.923	0.546	0.097	9.478

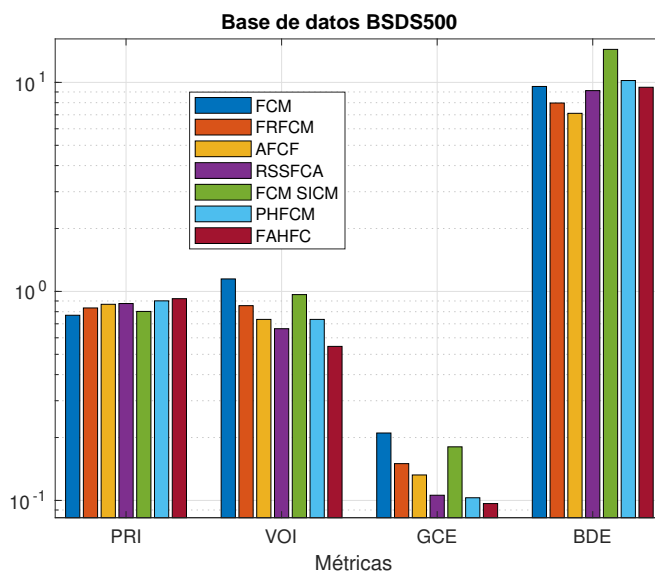


Figura 4.13: Gráfica de los resultados cuantitativos.

Finalmente, en la Figura 4.14 se ejemplifica el resultado de la segmentación de una imagen, con diferentes algoritmos para comparar el rendimiento de cada uno. Esta imagen es complicada de segmentar, debido a sus texturas y cambios de iluminación, sin embargo el algoritmo propuesto, FAHFC supera los resultados de los demás algoritmos con los que se realiza la comparación, siendo visualmente más parecido al *ground truth*. La imagen también revela la diferencia que existe entre los algoritmos PHFCM y el propuesto FAHFC, a pesar que ambos están basados en la teoría de

los conjuntos difusos titubeantes, no generan los mismos resultados, la ventaja la tiene FAHFC, que utiliza tres diferentes relaciones difusas para inicializar la matriz difusa titubeante y lograr un mejor rendimiento, mientras que PHFCM sigue utilizando FCM.

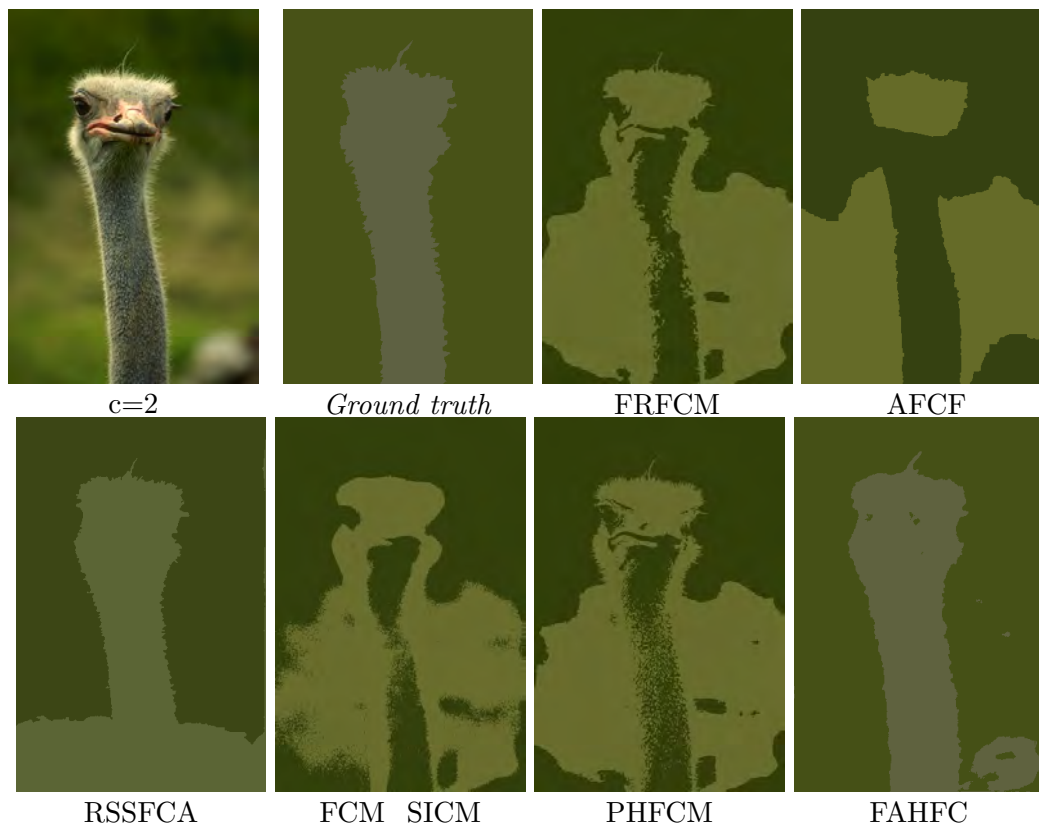


Figura 4.14: Resultado de la segmentación de la imagen 66075.

Para comparar los resultados con otras imágenes de diferentes características, se presenta la Figura 4.15, en las que el parámetro c sugiere el número de regiones en que es segmentada cada una de las imágenes, además de compararlas con su *ground truth*. En esta figura se observa que algunos algoritmos no logran segmentar la imagen en las regiones sugeridas, mientras que el algoritmo propuesto logra una mejor delimitación de las regiones. Aquí también se muestra la diferencia entre los algoritmos PHFCM y FAHFC, en donde el algoritmo propuesto tiene mejor rendimiento, en la calidad de la segmentación, generando regiones más homogéneas y delimitándolas mejor.

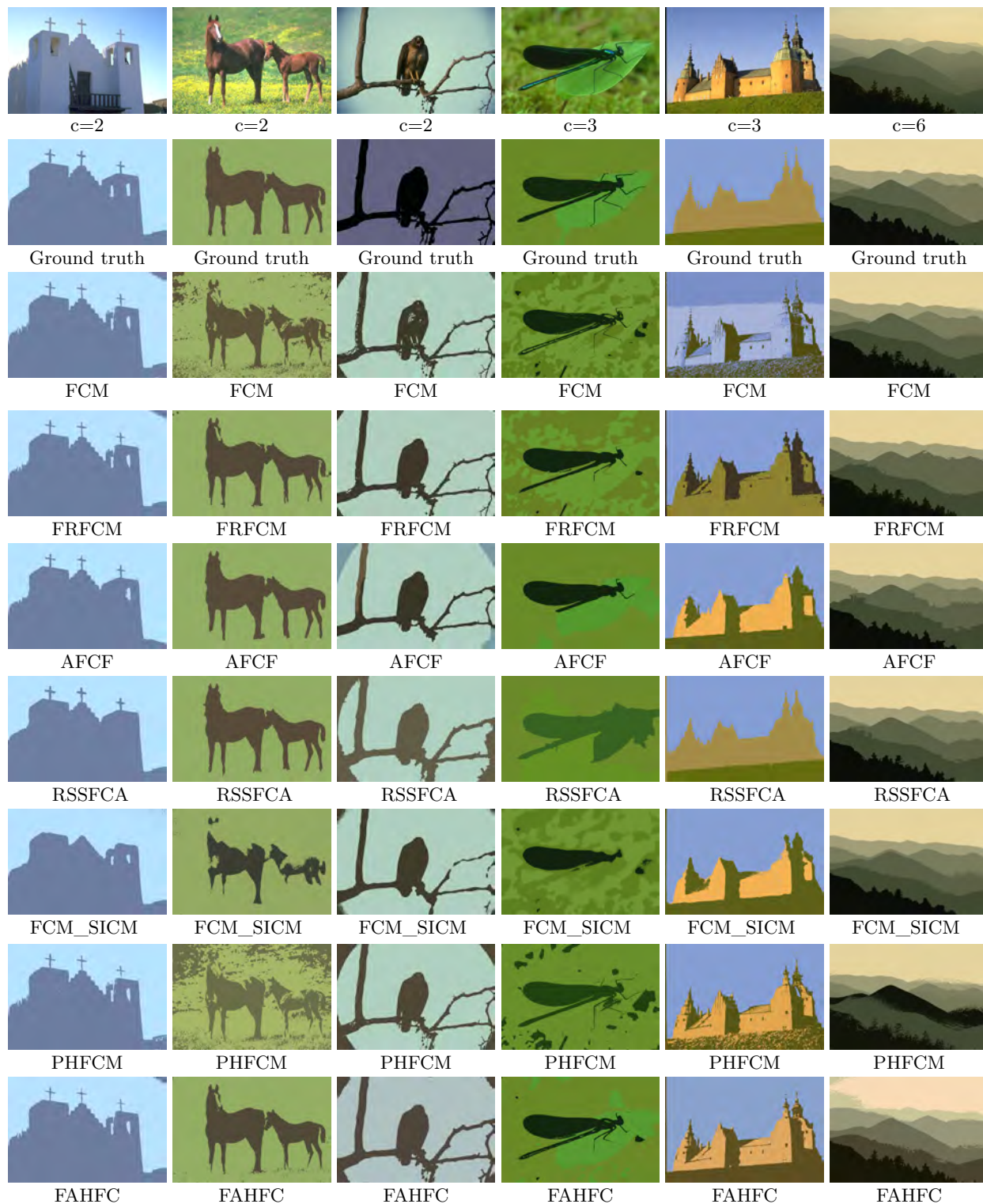


Figura 4.15: Resultado de la segmentación.

4.6.3. Discusión de los resultados

En esta sección se presentaron tres algoritmos de agrupamiento que fueron aplicados en bases de datos para segmentar imágenes y en bases de datos del reconocimiento de patrones; cada algoritmo está basado en la teoría de los conjuntos difusos titubeantes, además de otras características para mejorar el rendimiento de cada uno. El primer algoritmo presentado fue el PHFCM, cuya ventaja es su implementación en OpenMP para acelerar su tiempo de procesamiento, reduciéndose considerablemente comparándolo con el algoritmo secuencial. Por otra parte, con respecto a la calidad de agrupamiento, en los experimentos aplicados a la segmentación de imágenes, en las métricas consideradas, el algoritmo obtuvo el mejor desempeño frente a otros métodos de la literatura; además, demostraron que el algoritmo propuesto tiene una mejora en la asignación de píxeles encontrados en los bordes de las imágenes, teniendo una mayor precisión para delimitar las regiones en las imágenes, también fue capaz de obtener regiones más homogéneas y preservar ciertos detalles finos que los otros métodos omiten.

El segundo algoritmo es el AHFCM, que además de utilizar la base matemática del agrupamiento difuso titubeante, agrega un índice de validación para definir el número de clústers en que se van a agrupar los datos; con este algoritmo se realizaron experimentos en bases de datos aplicadas al reconocimiento de patrones, concretamente del repositorio de la UCI. Los resultados obtenidos se compararon con algunos algoritmos mencionados en la literatura, éstos mostraron que el algoritmo AHFCM tiene un mejor rendimiento en comparación con los demás. En el primer experimento se mostró la rapidez del algoritmo propuesto, con un tiempo medio de 0.028 segundos, además de converger en pocas iteraciones, menos de 15, se pudo observar que el algoritmo Kmediod converge en menos de 6 iteraciones pero su tiempo promedio de ejecución es de 0.8 segundos, superior al del algoritmo propuesto, asimismo, el tiempo promedio de ejecución del algoritmo FCM es el doble que el del algoritmo propuesto. En cuanto a las métricas para evaluar la calidad del agrupamiento, el algoritmo AHFCM obtuvo un mejor rendimiento sobre los otros algoritmos, respecto a la métrica ARI tuvo una diferencia de 0.021 sobre el algoritmo FRFCM, para ACC y Pureza la diferencia sobre el algoritmo FCM-ELPSO fue de 0.009 y 0.004, respectivamente; finalmente respecto a la métrica NMI, la diferencia con FCM fue de 0.007.

Finalmente, se propuso el algoritmo FAHFC, que también está basado en los conjuntos difusos titubeantes, y es capaz de encontrar el número de regiones en que se segmentarán las imágenes, de la misma forma que el algoritmo AHFCM, además, este algoritmo sugirió una alternativa para inicializar la matriz difusa titubeantes sin la necesidad de otros algoritmos de agrupamiento, usando relaciones difusas de diferentes características, como la homogeneidad, el cambio de espacio de color, y las texturas de las imágenes. En los experimentos realizados se observó que esta inicialización ayudó a mejorar la segmentación comparada con la versión PHFCM que también utiliza los HFS para hacer el agrupamiento, la diferencia es que inicializa la matriz difusa titubeante a partir del algoritmo FCM y dentro del espacio de color RGB. Además, los resultados de las métricas muestran también la superioridad del algoritmo FAHFC sobre los demás algoritmos.

Con los resultados obtenidos, en el siguiente capítulo se desglosan las conclusiones de la investigación de acuerdo a los objetivos planteados en el capítulo 1; además se detallan las aportaciones y los productos resultantes.

5 | Conclusiones

En este capítulo después de haber realizado las evaluaciones correspondientes se hace un análisis de las conclusiones de la investigación a partir de los resultados obtenidos y teniendo en cuenta los objetivos planteados. Además se describen los productos generados y las principales aportaciones. Por último se complementa con ideas y sugerencias para trabajos futuros.

5.1. Objetivos y alcances logrados

Las actividades realizadas con respecto a los objetivos y alcances planteados para el desarrollo de esta investigación se presentan en las Tablas 5.1 y 5.2.

Tabla 5.1: Objetivos realizados.

Objetivo	Actividad
Estudiar el algoritmo de agrupamiento <i>Fuzzy C-Means</i> y sus variantes.	Se estudio dicho algoritmo y se logró implementar una versión robusta " <i>Spatial α-Trimmed Fuzzy C-Means</i> "(STrFCM).
Estudiar y comprender la teoría de los Conjuntos Difusos Titubeantes y su aplicación en la segmentación de imágenes y el reconocimiento de patrones.	Se estudiaron conceptos de la teoría de los conjuntos difusos titubeantes, además de una revisión de la literatura donde se aplica esta teoría a las tareas de segmentación de imágenes y el reconocimiento de patrones. Se redactó un reporte del estado del arte.
Proponer un algoritmo de agrupamiento basado en la teoría de los conjuntos difusos titubeantes.	Se formuló el algoritmo " <i>Hesitant Fuzzy C-Means</i> basado en esta teoría, además de tres mejoras (PHFCM, AHFCM, FAHFC).
Evaluar la calidad del agrupamiento con datos sintéticos.	Se optó por evaluar los algoritmo propuestos en datos reales.
Aplicar y evaluar el algoritmo propuesto en el reconocimiento de patrones y la segmentación de imágenes médicas.	Se aplicó el algoritmo a bases de datos de reconocimiento de patrones y segmentación de imágenes reales y médicas, la evaluación se plasmó en un artículo (Figura 5).

Tabla 5.2: Alcances realizados.

Alcance	Actividad
Analizar dos algoritmos de agrupamiento difusos existentes.	Del análisis del algoritmo <i>Fuzzy C-Means</i> se generaron dos artículos, uno es la comparación de los métodos de inicialización (Figura 3) y el otro es una versión robusta (Figura 1).
Analizar la teoría de los conjuntos difusos titubeantes.	Se desarrolló el marco conceptual con el análisis de los conceptos de ésta teoría.
Plantear y desarrollar un algoritmo de agrupamiento basado en los conjuntos difusos titubeantes.	Se implementó el algoritmo <i>Parallel Hesitant Fuzzy C-Means</i> aceptado en la revista <i>Signal, Image and Video Processing</i> (Figura 5).
Evaluar la calidad de agrupamiento del algoritmo propuesto con al menos dos métricas definidas en la literatura.	La evaluación de la calidad del agrupamiento para bases de datos de segmentación de imágenes se realizó con las métricas: ACC, DSC, IOU, PRI, VOI, GCE y BDE, en cuanto al reconocimiento de patrones se usaron: ARI, ACC, NMI y <i>Purity</i> .
Aplicar el algoritmo propuesto al menos a dos bases de datos especializadas en imágenes médicas y dos más para reconocimiento de patrones y poder evaluar su desempeño.	El algoritmo propuesto fue aplicado a tres bases de datos de imágenes reales, dos médicas y a nueve bases de datos del repositorio de la UCI para el reconocimiento de patrones.
La comparación de los resultados se hará con al menos cinco algoritmos de agrupamiento por cada una de las áreas de aplicación mencionadas en esta propuesta.	Los resultados fueron comparados con siete algoritmos de la literatura aplicados a la segmentación de imágenes (FCM, AKFCMS, FRFCM, DSFCM_N, AFCF, RSSFCA y FCM_SICM) y seis para el reconocimiento de patrones (FCM, Kmediod, HPSOFCM, FRFCM, KMM, FCM-ELPSO).

5.2. Resultados de la investigación

5.2.1. Productos

Durante el desarrollo de esta investigación se obtuvieron los siguientes productos:

1. Reporte del estado del arte: documento en el que se presenta un resumen de diversas publicaciones recientes que tratan de temas importantes para esta investigación, estos fueron divididos en: algoritmo *Fuzzy C-Means* y sus variantes, bases teóricas de los conjuntos difusos titubeantes, propuestas actuales de segmentación de imágenes y reconocimiento de patrones, y las medidas de distancia.
2. Formulación e implementación de algoritmo para segmentar imágenes mediante la teoría de los conjuntos difusos titubeantes.
3. Formulación e implementación de algoritmo para agrupar datos mediante la teoría de los conjuntos difusos titubeantes.

4. Formulación e implementación de algoritmo de agrupamiento automático (no requiere del parámetro para definir el número de clusters).
5. Artículo para el congreso *12th Mexican Conference on Pattern Recognition MCPR 2020* (Figura 1 del Apéndice).
6. Artículo para el congreso *19th Mexican International Conference on Artificial Intelligence* (Figura 3 del Apéndice).
7. Artículo publicado en la revista *Signal, Image and Video Processing* (Figura: 5 del Apéndice).

5.2.2. Aportaciones

Las aportaciones obtenidas con esta investigación son:

- Bases matemáticas del algoritmo *Hesitant Fuzzy C-Means*.
- Algoritmo de agrupamiento basado en Conjuntos Difusos Titubeantes aplicado al Reconocimiento de Patrones.
- Algoritmo para segmentar imágenes reales y médicas basado en los Conjuntos Difusos Titubeantes.
- Algoritmo automático para agrupar datos, no requiere especificar el número de grupos.
- Generación de información difusa titubeante mediante relaciones difusas.

5.2.3. Conclusiones

En esta investigación se desarrolló la justificación matemática del algoritmo de agrupamiento *Hesitant Fuzzy C-Means*, basado en los conjuntos difusos titubeantes, los cuales a diferencia de la teoría difusa tradicional permite el uso de varios grados de pertenencia de un dato hacia cada uno de los grupos. Debido a que este algoritmo ocupa mas información de la que comúnmente utilizan en el agrupamiento, se optó por acelerarlo con una técnica de paralelización a nivel de *hardware*, utilizando OpenMP, resultando el algoritmo *Parallel Hesitant Fuzzy C-Means* (PHFCM). Uno de los inconvenientes en el agrupamiento es la inicialización de los parámetros, tal es el caso del número de grupos en que se va agrupar el conjunto de datos, por tal razón se propuso agregar al algoritmo HFCM, un índice de validación para no tener la necesidad de especificar este parámetro, este algoritmo lleva por nombre *Automatic Hesitant Fuzzy C-Means* (AHFCM). Por otra parte, la información difusa titubeante según la literatura es definida por expertos, una alternativa para generarla fue mediante el uso de relaciones difusas, surgiendo así el algoritmo *Fast and Automatic Hesitant Fuzzy Clustering* (FAHFC).

Con los experimentos realizados se logró la evaluación de los tres algoritmos en términos de calidad de agrupamiento y convergencia, mostrando ser superiores a algunos algoritmos seleccionados de la literatura. Por su parte, PHFCM fue evaluado en bases de datos en la segmentación de imágenes como: Weizmann, MRCORID y BSDS500, y algunas especializadas en imágenes médicas como: ISIC 2019 y WBC; los principales descubrimientos en la segmentación de estas imágenes fueron que el algoritmo fue capaz de generar regiones más homogéneas que los algoritmos con los

que fue comparado, conservando detalles finos además de delimitar mejor las regiones. Otro resultado impactante fue que además de reducir el tiempo de ejecución con la técnica de paralelización se logró reducir el número de iteraciones requeridas para converger, la explicación posible a este resultado debió darse con la asignación de la matriz difusa titubeante, generando valores aptos para lograr converger en menos iteraciones.

En cuanto al algoritmo AHFCM, fue evaluado en bases de datos del reconocimiento de patrones, específicamente tomadas del repositorio de la UCI, obteniendo resultados favorables sobre los algoritmos con los que se realizó la comparación, además de ser capaz de encontrar el número de grupos para agrupar los datos en todas las bases de datos en donde se realizó la evaluación.

Finalmente, el algoritmo FAHFC fue resultado de la estancia académica (Figura 6) cuyo objetivo fue la inicialización de la matriz difusa. Este algoritmo fue evaluado en bases de datos para segmentación de imágenes y comparado con algunos algoritmos del estado del arte, mostrando ser superior, en la mayoría de las métricas utilizadas para la comparación. Además que es más rápido que la versión paralela PHFCM, ya que hace una reducción de la imagen para hacer la agrupación y encontrar los centroides iniciales que son utilizados para hacer la segmentación de la imagen original. En cuanto a la calidad del agrupamiento, se observó que el uso de las relaciones difusas mejora la calidad del agrupamiento, generando regiones mas homogéneas, con bordes delimitados.

Finalmente, la producción científica obtenida en esta investigación, tal como se ha mencionado, fueron tres artículos publicados, dos en congresos internacionales (1 y 3) y uno publicado en la revista *Signal, Image and Video Processing* indizada en el *Journal Citation Report (JCR)*. Además, se enviaron dos artículos y se esperan comentarios por parte de las dos revistas JCR (7 y 9)

5.2.4. Trabajos futuros

- Extender las bases matemáticas hacia otras teorías de los conjuntos difusos, por ejemplo los conjuntos difusos intuitivos, en lugar de tener un solo valor de pertenencia, no pertenencia e indecisión, se podrían considerar varios valores posibles.
- Dado que se describe la formulación de cada algoritmo, estos pueden ser aplicados en diferentes áreas que requieran agrupación de datos, incluso también clasificación, por ejemplo en el entorno médico se podría aplicar para clasificar los tipos de tumores en una imagen.
- Incluir información espacial a los algoritmos para hacerlos robustos al ruido de las imágenes.

Referencias

- Albashah, N. L. S. B., Dass, S. C., Asirvadam, V. S., and Meriaudeau, F. (2018). Segmentation of blood clot mri images using intuitionistic fuzzy set theory. In *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, pages 533–538. IEEE.
- Aliahmadipour, L., Torra, V., and Eslami, E. (2017). On hesitant fuzzy clustering and clustering of hesitant fuzzy data. In *Fuzzy sets, rough sets, multisets and clustering*, pages 157–168. Springer.
- Aljebory, K. M. and Mohammed, T. S. (2018). Modified fuzzy c-means clustering algorithm application in medical image segmentation. *JEA Journal of Electrical Engineering*, 2(1):1–9.
- Alom, M. Z., Yakopcic, C., Hasan, M., Taha, T. M., and Asari, V. K. (2019). Recurrent residual u-net for medical image segmentation. *Journal of Medical Imaging*, 6(1):014006.
- Arora, J. and Tushir, M. (2018). Robust spatial intuitionistic fuzzy c-means with city-block distance clustering for image segmentation. *Journal of Intelligent & Fuzzy Systems*, 35(5):5255–5264.
- Atannasov, K. (1999). Intuitionistic fuzzy sets: Theory and applications. *Physica-Verlag, New York*.
- Athertya, J. S., Kumar, G. S., and Govindaraj, J. (2019). Detection of modic changes in mr images of spine using local binary patterns. *Biocybernetics and Biomedical Engineering*, 39(1):17–29.
- Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495.
- Bezdek, J. C., Ehrlich, R., and Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3):191–203.
- Bezdek, J. C., Keller, J., Krisnapuram, R., and Pal, N. (1999). *Fuzzy models and algorithms for pattern recognition and image processing*, volume 4. Springer Science & Business Media.
- Bharati, S. K. (2018). Hesitant fuzzy computational algorithm for multiobjective optimization problems. *International Journal of Dynamics and Control*, 6(4):1799–1806.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.
- Bow, S.-T. (2002). *Pattern recognition and image preprocessing*. Marcel Dekker New York.
- Bullock, J., Cuesta-Lázaro, C., and Quera-Bofarull, A. (2019). Xnet: A convolutional neural network

- (cnn) implementation for medical x-ray image segmentation suitable for small datasets. In *Medical Imaging 2019: Biomedical Applications in Molecular, Structural, and Functional Imaging*, volume 10953, pages 453–463. SPIE.
- Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.
- Chaira, T. (2021). An intuitionistic fuzzy clustering approach for detection of abnormal regions in mammogram images. *Journal of Digital Imaging*, 34(2):428–439.
- Chen, J., Zheng, H., Lin, X., Wu, Y., and Su, M. (2018). A novel image segmentation method based on fast density clustering algorithm. *Engineering Applications of Artificial Intelligence*, 73:92–110.
- Chen, N., Xu, Z., and Xia, M. (2013). Correlation coefficients of hesitant fuzzy sets and their applications to clustering analysis. *Applied Mathematical Modelling*, 37(4):2197–2211.
- Chen, S., Xu, Z., and Tang, Y. (2014). A hybrid clustering algorithm based on fuzzy c-means and improved particle swarm optimization. *Arabian Journal for Science and Engineering*, 39(12):8875–8887.
- Combaila, M., Codella, N. C., Rotemberg, V., Helba, B., Vilaplana, V., Reiter, O., Halpern, A. C., Puig, S., and Ma2ehy, J. (2019). Bcn20000: Dermoscopic lesions in the wild. *arXiv preprint arXiv:1908.02288*.
- Criminisi, A. et al. (2004). Microsoft research cambridge object recognition image database.
- De Oliveira, J. V. and Pedrycz, W. (2007). *Advances in fuzzy clustering and its applications*. John Wiley & Sons.
- de Oliveira, P. V. and Yamanaka, K. (2018). Image segmentation using multilevel thresholding and genetic algorithm: An approach. In *2018 2nd International Conference on Data Science and Business Analytics (ICDSBA)*, pages 380–385. IEEE.
- Domínguez, D. (2019). Extracción de superpíxeles y fusión local mediante un algoritmo de agrupamiento difuso intuicionista.
- Dougherty, G. (2011). *Medical image processing: techniques and applications*. Springer Science & Business Media.
- Dua, D. and Graff, C. (2017). Uci machine learning repository.
- Fan, R., Bocus, M. J., Zhu, Y., Jiao, J., Wang, L., Ma, F., Cheng, S., and Liu, M. (2019). Road crack detection using deep convolutional neural network and adaptive thresholding. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 474–479. IEEE.
- Fang, L., Liu, J., Liu, J., and Mao, R. (2018). Automatic segmentation and 3d reconstruction of spine based on fcn and marching cubes in ct volumes. In *2018 10th International Conference on Modelling, Identification and Control (ICMIC)*, pages 1–5. IEEE.
- Farhadinia, B. (2017). A hesitant fuzzy based medical diagnosis problem. *International Journal*

- on *Data Science and Technology*, 3(1):1–17.
- Freixenet, J., Muñoz, X., Raba, D., Martí, J., and Cufí, X. (2002). Yet another survey on image segmentation: Region and boundary information integration. In *European conference on computer vision*, pages 408–422. Springer.
- Gan, G., Ma, C., and Wu, J. (2007). Data clustering: Theory, algorithms, and applications (asa-siam series on statistics and applied probability).
- Gonzalez, R. C., Woods, R. E., et al. (2002). Digital image processing.
- Hu, G. and Du, Z. (2019). Adaptive kernel-based fuzzy c-means clustering with spatial constraints for image segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 33(01):1954003.
- Huang, S., Kang, Z., Xu, Z., and Liu, Q. (2021a). Robust deep k-means: An effective and simple method for data clustering. *Pattern Recognition*, 117:107996.
- Huang, Y.-P., Singh, P., Kuo, W.-L., and Chu, H.-C. (2021b). A type-2 fuzzy clustering and quantum optimization approach for crops image segmentation. *International Journal of Fuzzy Systems*, 23(3):615–629.
- Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.
- Jähne, B. (2005). *Digital Image Processing, 6th Revised and Extended Edition*. Springer.
- Jia, X., Lei, T., Du, X., Liu, S., Meng, H., and Nandi, A. K. (2020). Robust self-sparse fuzzy clustering for image segmentation. *IEEE Access*, 8:146182–146195.
- Junhua Hu, Yan Yang, X. Z. and Chen, X. (2017). Similarity and entropy measures for hesitant fuzzy sets. *International Transactions in Operational Research*.
- Kala, R. and Deepa, P. (2021). Spatial rough intuitionistic fuzzy c-means clustering for mri segmentation. *Neural Processing Letters*, 53(2):1305–1353.
- Kang, Z., Wen, L., Chen, W., and Xu, Z. (2019). Low-rank kernel learning for graph-based clustering. *Knowledge-Based Systems*, 163:510 – 517.
- Karimi, D. and Salcudean, S. E. (2019). Reducing the hausdorff distance in medical image segmentation with convolutional neural networks. *IEEE Transactions on medical imaging*, 39(2):499–513.
- Kaur, P. and Chaira, T. (2021). A novel fuzzy approach for segmenting medical images. *Soft Computing*, 25(5):3565–3575.
- Khorram, B. and Yazdi, M. (2019). A new optimized thresholding method using ant colony algorithm for mr brain image segmentation. *Journal of digital imaging*, 32(1):162–174.
- Kpalma, K. and Ronsin, J. (2007). An overview of advances of pattern recognition systems in computer vision.
- Kumar, D., Agrawal, R., and Verma, H. (2020). Kernel intuitionistic fuzzy entropy clustering for mri image segmentation. *Soft Computing*, 24(6):4003–4026.

- Kumar, D., Agrawal, R. K., and Kirar, J. S. (2019). Intuitionistic fuzzy clustering method with spatial information for mri image segmentation. In *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–7. IEEE.
- Lei, B. and Fan, J. (2019). Image thresholding segmentation method based on minimum square rough entropy. *Applied Soft Computing*, 84:105687.
- Lei, T., Jia, X., Liu, T., Liu, S., Meng, H., and Nandi, A. K. (2019a). Adaptive morphological reconstruction for seeded image segmentation. *IEEE Transactions on Image Processing*, 28(11):5510–5523.
- Lei, T., Jia, X., Zhang, Y., He, L., Meng, H., and Nandi, A. K. (2018a). Significantly fast and robust fuzzy c-means clustering algorithm based on morphological reconstruction and membership filtering. *IEEE Transactions on Fuzzy Systems*.
- Lei, T., Jia, X., Zhang, Y., Liu, S., Meng, H., and Nandi, A. K. (2018b). Superpixel-based fast fuzzy c-means clustering for color image segmentation. *IEEE Transactions on Fuzzy Systems*, 27(9):1753–1766.
- Lei, T., Liu, P., Jia, X., Zhang, X., Meng, H., and Nandi, A. K. (2019b). Automatic fuzzy clustering framework for image segmentation.
- Leondes, C. T. (1998). *Image processing and pattern recognition*, volume 5. Elsevier.
- Li, C., Zhao, H., and Xu, Z. (2018). Kernel c-means clustering algorithms for hesitant fuzzy information in decision making. *International Journal of Fuzzy Systems*, 20(1):141–154.
- Li, D., Zeng, W., and Li, J. (2015). New distance and similarity measures on hesitant fuzzy sets and their applications in multiple criteria decision making. *Engineering Applications of Artificial Intelligence*, 40:11–16.
- Li, F., Liu, M., Initiative, A. D. N., et al. (2019). A hybrid convolutional and recurrent neural network for hippocampus analysis in alzheimer’s disease. *Journal of neuroscience methods*, 323:108–118.
- Li, M., Wang, L., Deng, S., and Zhou, C. (2020). Color image segmentation using adaptive hierarchical-histogram thresholding. *PloS one*, 15(1):e0226345.
- Liao, H. and Xu, Z. (2017). *Hesitant fuzzy decision making methodologies and applications*. Springer.
- Ling, C., Diyana, W. M., Zaki, W., Hussain, A., and Hamid, H. A. (2016). Semi-automated vertebral segmentation of human spine in mri images. In *2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES)*, pages 120–124. IEEE.
- Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. (2010). Understanding of internal clustering validation measures. In *2010 IEEE international conference on data mining*, pages 911–916. IEEE.
- Luna, D. C. (2021). Sobresegmentación de imágenes a color mediante un grafo difuso basado en una medida de distancia espacial y color.
- Lv, J.-h., Guo, S.-c., and Guo, F.-f. (2019). Study on hesitant fuzzy information measures and their

- clustering application. *Computational Intelligence and Neuroscience*.
- Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423.
- 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):1650–1654.
- Meilă, M. (2007). Comparing clusterings—an information based distance. *Journal of multivariate analysis*, 98(5):873–895.
- Minnema, J., van Eijnatten, M., Kouw, W., Diblen, F., Mendrik, A., and Wolff, J. (2018). Ct image segmentation of bone for medical additive manufacturing using a convolutional neural network. *Computers in biology and medicine*, 103:130–139.
- Mújica Vargas, D. (2013). *Segmentación de imágenes utilizando Algoritmos Robustos de Agrupamiento Difuso*. PhD thesis, Escuela Superior de Ingeniería Mecánica y Eléctrica, Instituto Politécnico Nacional.
- Naz, S. and Akram, M. (2019). Novel decision-making approach based on hesitant fuzzy sets and graph theory. *Computational and Applied Mathematics*, 38(1):7.
- Ngan, R. T., Ali, M., Fujita, H., Abdel-Basset, M., Giang, N. L., Manogaran, G., Priyan, M., et al. (2019). A new representation of intuitionistic fuzzy systems and their applications in critical decision making. *IEEE Intelligent Systems*, 35(1):6–17.
- Nie, F., Wang, C.-L., and Li, X. (2019). K-multiple-means: A multiple-means clustering method with specified k clusters. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 959–967.
- Park, H.-S. and Jun, C.-H. (2009). A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2):3336–3341.
- Pătrașcu, V. (2007). Fuzzy image segmentation based on triangular function and its n-dimensional extension. In *Soft Computing in Image Processing*, pages 187–207. Springer.
- Pei, H.-X., Zheng, Z.-R., Wang, C., Li, C.-N., and Shao, Y.-H. (2017). D-fcm: Density based fuzzy c-means clustering algorithm with application in medical image segmentation. *Procedia Computer Science*, 122:407–414.
- Reis, F. A., Almeida, R., Kijak, E., Malinowski, S., Guimarães, S. J. F., and do Patrocínio, Z. K. (2019). Combining convolutional side-outputs for road image segmentation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.
- Rockafellar, R. T. (1993). Lagrange multipliers and optimality. *SIAM review*, 35(2):183–238.
- Russ, J. C. (2016). *The image processing handbook*. CRC press.
- Sadad, T., Munir, A., Saba, T., and Hussain, A. (2018). Fuzzy c-means and region growing based

- classification of tumor from mammograms using hybrid texture feature. *Journal of computational science*, 29:34–45.
- Santos, J. M. and Embrechts, M. (2009). On the use of the adjusted rand index as a metric for evaluating supervised classification. In *International conference on artificial neural networks*, pages 175–184. Springer.
- Schmidt, B., Gonzalez-Dominguez, J., Hundt, C., and Schlarb, M. (2017). *Parallel programming: concepts and practice*. Morgan Kaufmann.
- Snyder, W. E. and Qi, H. (2010). *Machine vision*. Cambridge University Press.
- Sun, G., Guan, X., Yi, X., and Zhou, Z. (2018). Grey relational analysis between hesitant fuzzy sets with applications to pattern recognition. *Expert Systems with Applications*, 92:521–532.
- Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.
- Tan, Z. and Zhang, D. (2020). A fuzzy adaptive gravitational search algorithm for two-dimensional multilevel thresholding image segmentation. *Journal of Ambient Intelligence and Humanized Computing*, 11(11):4983–4994.
- Theodoridis, S. and Koutroumbas, K. (2009). *Pattern Recognition*. Academic Press, Boston, fourth edition.
- Tomita, N., Cheung, Y. Y., and Hassanpour, S. (2018). Deep neural networks for automatic detection of osteoporotic vertebral fractures on ct scans. *Computers in biology and medicine*, 98:8–15.
- Torra, V. (2010). Hesitant fuzzy sets. *International Journal of Intelligent Systems*, 25(6):529–539.
- Tuan, T. M., Fujita, H., Dey, N., Ashour, A. S., Ngoc, V. T. N., Chu, D.-T., et al. (2018). Dental diagnosis from x-ray images: An expert system based on fuzzy computing. *Biomedical Signal Processing and Control*, 39:64–73.
- Unnikrishnan, R., Pantofaru, C., and Hebert, M. (2007). Toward objective evaluation of image segmentation algorithms. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):929–944.
- Wang, M., Wan, Y., Gao, X., Ye, Z., and Chen, M. (2018a). An image segmentation method based on fuzzy c-means clustering and cuckoo search algorithm. In *Ninth International Conference on Graphic and Image Processing (ICGIP 2017)*, volume 10615, page 1061525. International Society for Optics and Photonics.
- Wang, Q., Wang, X., Fang, C., and Yang, W. (2020). Robust fuzzy c-means clustering algorithm with adaptive spatial & intensity constraint and membership linking for noise image segmentation. *Applied Soft Computing*, 92:106318.
- Wang, Y., Zhang, R., and Qian, L. (2018b). An improved a* algorithm based on hesitant fuzzy set theory for multi-criteria arctic route planning. *Symmetry*, 10(12):765.

- Wierchoń, S. T. and Kłopotek, M. A. (2018). *Modern algorithms of cluster analysis*, volume 34. Springer.
- Winn, J., Criminisi, A., and Minka, T. (2005). Object categorization by learned universal visual dictionary. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1800–1807. IEEE.
- Wollmann, T., Gunkel, M., Chung, I., Erfle, H., Rippe, K., and Rohr, K. (2019). Gruu-net: Integrated convolutional and gated recurrent neural network for cell segmentation. *Medical image analysis*, 56:68–79.
- Wu, C. and Yang, X. (2020). Robust credibilistic fuzzy local information clustering with spatial information constraints. *Digital Signal Processing*, 97:102615.
- Wu, C., Zhang, L., Zhang, H., and Yan, H. (2019). Improved superpixel-based fast fuzzy c-means clustering for image segmentation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1455–1459. IEEE.
- Xia, M. and Xu, Z. (2011). Hesitant fuzzy information aggregation in decision making. *International journal of approximate reasoning*, 52(3):395–407.
- Xiong, S.-H., Chen, Z.-S., and Chin, K.-S. (2018). A novel magdm approach with proportional hesitant fuzzy sets. *International Journal of Computational Intelligence Systems*, 11(1):256–271.
- Xu, Z. (2016). *Hesitant fuzzy sets theory*. Springer.
- Xu, Z. and Wu, J. (2010). Intuitionistic fuzzy c-means clustering algorithms. *Journal of Systems Engineering and Electronics*, 21(4):580–590.
- Xu, Z. and Xia, M. (2011). Distance and similarity measures for hesitant fuzzy sets. *Information Sciences*, 181(11):2128–2138.
- Yang, M.-S. and Hussain, Z. (2018). Distance and similarity measures of hesitant fuzzy sets based on hausdorff metric with applications to multi-criteria decision making and clustering. *Soft Computing*, pages 1–14.
- Zeng, W., Li, D., and Yin, Q. (2016). Distance and similarity measures between hesitant fuzzy sets and their application in pattern recognition. *Pattern Recognition Letters*, 84:267–271.
- Zhang, J. and Ma, Z. (2020). Hybrid fuzzy clustering method based on fcm and enhanced logarithmical pso (elpso). *Computational intelligence and neuroscience*, 2020.
- Zhang, S., Jiang, W., and Satoh, S. (2018a). Multilevel thresholding color image segmentation using a modified artificial bee colony algorithm. *IEICE TRANSACTIONS on Information and Systems*, 101(8):2064–2071.
- Zhang, Y., Bai, X., Fan, R., and Wang, Z. (2018b). Deviation-sparse fuzzy c-means with neighbor information constraint. *IEEE Transactions on Fuzzy Systems*, 27(1):185–199.
- Zhang, Z. and Song, J. (2019). An adaptive fuzzy level set model with local spatial information for medical image segmentation and bias correction. *IEEE Access*, 7:27322–27338.

- Zheng, X., Wang, Y., Wang, G., and Liu, J. (2018). Fast and robust segmentation of white blood cell images by self-supervised learning. *Micron*, 107:55–71.
- Zhou, W. and Xu, Z. (2018). Extended intuitionistic fuzzy sets based on the hesitant fuzzy membership and their application in decision making with risk preference. *International Journal of Intelligent Systems*, 33(2):417–443.
- Zhou, X., Li, Z., Xie, H., Feng, T., Lu, Y., Wang, C., and Chen, R. (2020). Leukocyte image segmentation based on adaptive histogram thresholding and contour detection. *Current Bioinformatics*, 15(3):187–195.
- Zhu, Z., Li, D., Hu, Y., Li, J., Liu, D., and Li, J. (2021). Indoor scene segmentation algorithm based on full convolutional neural network. *Neural Computing and Applications*, 33(14):8261–8273.

Apéndice

Producción científica



This is to Certify that

*Virna V. Vela-Rincón, Dante Mújica-Vargas, Manuel Mejía Lavalle, and
Andrea Magadán Salazar*

**Attended the
12th Mexican Conference on Pattern Recognition
and present the conference**

*Spatial α -Trimmed Fuzzy C-means Algorithm to image
segmentation*



**Karina Mariela Figueroa Mora
General Chair**

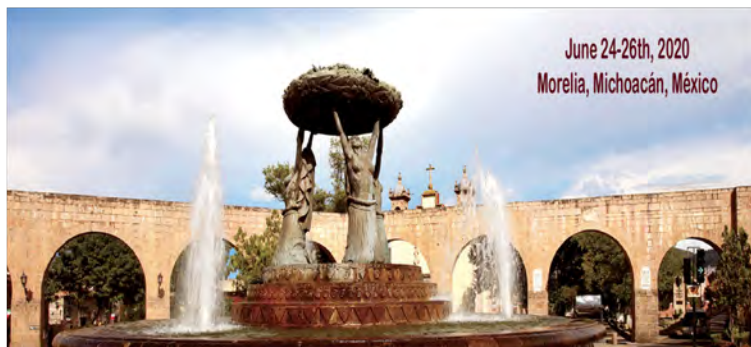


Figura 1: Constancia de autor del artículo publicado en el congreso MCPR 2020.



Spatial α -Trimmed Fuzzy C-Means Algorithm to Image Segmentation

Virna V. Vela-Rincón^(*), Dante Mújica-Vargas, Manuel Mejía Lavalle,
and Andrea Magadán Salazar

Tecnológico Nacional de México/CENIDET, Cuernavaca, Morelos, Mexico
{viryvela,dantemv,mlavalle,magadan}@cenidet.edu.mx

Abstract. An important aspect should be taken into account, when an image is segmented, the presence of atypical information. In this investigation an algorithm is proposed that is noise tolerant in the segmentation process. A method to image segmentation that combines Fuzzy C-Means (FCM) algorithm and Trimmed Means filter, called Spatial α Trimmed Fuzzy C-means, using local information to achieve better segmentation. The FCM is very sensitive to noise, and the Trimmed Means filter is used to eliminate outliers with a lower computational cost. Compared to some state-of-the-art algorithms, the proposed is faster and noise tolerant, demonstrating better performance in the metrics considered.

Keywords: Image segmentation · Fuzzy C-Means · α -Trimmed means filter · Spatial information

1 Introduction

Image segmentation is defined as a process by which an image is divided into homogeneous areas, each belonging to different objects with similar features (i.e. color, contrast, brightness, texture and so on) [16]. Segmentation has a many applications medical image analysis, autonomous vehicles, video surveillance, and augmented reality to count a few [8]. Generally, the image segmentation method are divided into histogram-based approaches [11], edge detection approaches [19], region-based approaches [15], clustering approaches [12].

One of the most popular image segmentation algorithms is Fuzzy C-means (FCM), which gives each data a degree of membership through a distance measure to assign to the nearest group [6,17]. FCM is a noise sensitive algorithm, some investigations try to improve this algorithm, some guarantee noise immunity and preserve image details by incorporating local spatial and gray information together to enhances the clustering performance [5]; other try to reduce the high computational complexity that surge an iterative calculation of the distance between pixels within local spatial neighbors and clustering centers, based on morphological reconstruction and membership filtering [10]. There algorithms used to determinate the parameters by introducing density for each sample,

© Springer Nature Switzerland AG 2020
K. M. Figueroa Mora et al. (Eds.): MCPR 2020, LNCS 12088, pp. 118–128, 2020.
https://doi.org/10.1007/978-3-030-49076-8_12

Figura 2: Primera página del artículo publicado en el congreso MCPR 2020.

**THE MEXICAN SOCIETY FOR ARTIFICIAL INTELLIGENCE (SMIA)
AND THE UNIVERSIDAD PANAMERICANA**




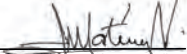
Award this certificate to
Virna Viridiana Vela Rincón, Celia Ramos-Palencia and Dante Mújica-Vargas


for presentation of the paper entitled
Comparison methods for Fuzzy C-means initialization applied to image segmentation

at the 19th Mexican International Conference of Artificial Intelligence, MICAI 2020.
Mexico City, Mexico, October 12 - 17, 2020.




Dr. Félix Castro Espinoza
SMIA President


Dra. Lourdes Martínez Villaheñor
Program Chair


Dr. Oscar Herrera Aleántara
Program Chair



Dr. Miram Ponce Espinosa
MICAI Local Chair



Figura 3: Constancia de autor del artículo publicado en el congreso MICAI 2020.



Comparison Methods for Fuzzy C-Means Initialization Applied to Image Segmentation

Virna V. Vela-Rincón¹ (✉), Celia Ramos-Palencia¹, Dante Mújica-Vargas¹,
Jean Marie Vianney Kinani², and Eduardo Ramos-Díaz²

¹ Tecnológico Nacional de México/CENIDET, Cuernavaca, Morelos, Mexico
viryvela@cenidet.edu.mx

² Instituto Politécnico Nacional-UPIIH, San Agustín Tlaxiaca, Hidalgo, Mexico

Abstract. Fuzzy C-means (FCM) is one of the most used clustering algorithms, some research seeks to achieve a better quality in the results. It is well known that an adequate selection of the initial centroids will achieve a better clustering result. In this paper, a comparison of some initialization methods applied to the FCM algorithm is made, the experimental results suggest that the K-means++ initialization method is the most suitable for image segmentation, since it produces a better initialization condition, in addition to improve convergence times.

Keywords: Initialization methods · Fuzzy C-means · Image segmentation · K-means++

1 Introduction

Clustering breaks down a given set of objects into subgroups or clusters based on similarity. Its objective is to divide the data set in such a way that the objects that belong to the same cluster are as similar as possible, while the objects that belong to different clusters are as different as possible [1]. Clustering can be applied to image segmentation, one of the most popular methods is the Fuzzy C-means algorithm, which allows the gradual belonging in a closed interval $\mu \in [0, 1]$ of the data with respect to the clusters or regions of interest.

Fuzzy C-means cannot guarantee a unique clustering result because the initial centroids are chosen at random, this yields a higher number of iterations and the clustering result unstable. Many investigations try to find the best way to initialize this centroids, such is the case in [2] that extracts the most vivid and distinguishable colors called **dominant colors** from a given set of colors, this are obtained by computing the similarities between color points in the set and well-known reference colors. These similarities are used to calculate the degree of belonging of each color point to the reference colors. Based on the degrees of membership, we identify the dominant colors and then select to guide the selection of the initial centroids. [3] presents a initialization scheme called Hierarchical

© Springer Nature Switzerland AG 2020

L. Martínez-Villaseñor et al. (Eds.): MICAI 2020, LNAI 12469, pp. 350–362, 2020.

https://doi.org/10.1007/978-3-030-60887-3_31

Figura 4: Primera página del artículo publicado en el congreso MICAI 2020.

Signal, Image and Video Processing
<https://doi.org/10.1007/s11760-021-01957-8>

ORIGINAL PAPER



Parallel hesitant fuzzy C-means algorithm to image segmentation

Virna V. Vela-Rincón¹ · Dante Mújica-Vargas¹ · Jose de Jesus Rubio²

Received: 12 December 2020 / Revised: 6 April 2021 / Accepted: 29 May 2021
 © The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract

Hesitant fuzzy information allows clustering data with multiple possible membership values for a single item in a reference set. Hesitant fuzzy sets have been applied in many decision-making problems, obtaining better results against others kinds of fuzzy sets. So, in this paper a method for image segmentation based on the hesitant fuzzy set theory is investigated. Additionally, processing time is sped up with a hardware-level parallelization technique using OpenMP. Comparing the experimental results, it can be seen that the segmentation by the propose algorithm is superior, compared to some of the state of the art. The most striking feature to emerge from this algorithm is its ability to preserve the details of the boundaries of the region, in addition to the fact that the regions are more homogeneous.

Keywords Color image segmentation · Hesitant fuzzy set · Parallelization technique · OpenMP

1 Introduction

Image segmentation can be defined as a process by which an image is divided into homogeneous areas, each belonging to different objects with similar features (i.e., color, brightness, texture and so on) [1,2]. There are many applications in this field, from medical applications such as dental diagnosis from X-ray images, agricultural applications like pest identification and detection until the location of objects in aerial or satellite images [3–5]. This research is focused on an algorithm to image segmentation, clustering pixel on datasets of real images, based on hesitant fuzzy set theory to obtain a better performance to delimit the region in an image.

In image segmentation task, specifically the algorithms based on clustering, important contributions have been made to the state of the art. In [6] and [7], algorithms based on FCM are proposed: The first uses morphological reconstruc-

tion and filtering of members (FRFCM), and the other an adaptive kernel with spatial restrictions (AKFCMS); both use spatial information to improve the efficiency of image segmentation. [8] presents two clustering methods; DSFCM uses the theoretical data estimated from the measured data for clustering and takes the deviations between the measured data and their theoretical values into consideration. The neighborhood information is introduced to improve the clustering performance in DSFCM_N. Both algorithms acquire more accurate cluster centers by considering that the use of theoretical values of data are more reliable than just considering the data itself. There is more research that uses spatial information, such as [9–11]. Other effective method for image segmentation is superpixel-based fast fuzzy C-means (SFFCM) clustering [12,13]. An automatic fuzzy clustering framework (AFCF) is proposed in [14], based on the idea of use superpixel for the density peak (DP) algorithm, which improves the computational efficiency of the DP algorithm. Because neighboring spatial information from both pixels and membership is considered, the result of segmentation is effectively improved. Finally, [15] proposed a robust self-sparse fuzzy clustering algorithm (RSSFCA) that incorporates a regularization term to its objective function to balance the sparsity of membership and fuzziness, and it thus achieves self-sparse fuzzy clustering and also employs the CCF-ADB to achieve effectively small region merging.

Analyzing the aforementioned methods, all demonstrate effectiveness when segmenting images using different tools

✉ Dante Mújica-Vargas
 dante.mv@cenidet.tecnm.mx
 Virna V. Vela-Rincón
 virnyvela@cenidet.edu.mx
 Jose de Jesus Rubio
 rubio.josedejesus@gmail.com

¹ Tecnológico Nacional de México/CENIDET, Cuernavaca, Morelos, Mexico

² Sección de Estudios de Posgrado e Investigación, Esime Azcapotzalco, Instituto Politécnico Nacional, Ciudad de México, Mexico

Published online: 19 June 2021

Springer

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Figura 5: Primera página del artículo publicado en la revista *Signal, Image and Video Processing*.

Estancia académica



INSTITUTO POLITÉCNICO NACIONAL
 ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA, UNIDAD AZCAPOTZALCO
 SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
 Doctorado en Ingeniería de Sistemas Robóticos y Mecatrónicos



Ciudad de México, a 12 de enero de 2022

Dr. Juan Gabriel González Serna
Jefe del Departamento de Ciencias Computacionales
Tecnológico Nacional de México/CENIDET

Por este medio permito informarle que la **M.C. VIRNA VIRIDIANA VELA RINCÓN** estudiante del programa de Doctorado en Ciencias de la Computación del Tecnológico Nacional de México/CENIDET con número de control D15CE032, ha **concluido SATISFACTORIAMENTE** su estancia académica en el área de Doctorado en Ingeniería en Sistemas Robóticos y Mecatrónicos, SEPI, ESIME Azcapotzalco, Instituto Politécnico Nacional, con una duración de tres meses y realizando las actividades descritas en el siguiente cronograma:

Actividades	2021																		
	Septiembre				Octubre				Noviembre				Diciembre						
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4			
Estudio de la matriz difusa titubeante			x	x															
Estudio de técnicas de inicialización para algoritmos de agrupamiento			x	x															
Planteamiento de técnica de inicialización de la matriz difusa titubeante				x	x	x	x	x	x										
Implementación de técnica de inicialización de la matriz difusa titubeante								x	x	x									
Evaluación de la calidad del agrupamiento del algoritmo HFCM con la técnica propuesta										x	x	x	x						
Artículo de los resultados obtenidos																x	x	x	x

Sin otro particular por el momento, le envío un cordial saludo.

ATENTAMENTE

Dr. José de Jesús Rubio Ávila
 Doctorado en Ingeniería de Sistemas Robóticos y Mecatrónicos,
 SEPI, ESIME Azcapotzalco, Instituto Politécnico Nacional

Edificio de Posgrado y Campus Virtual | Av. de las Granjas 682, Col. Santa Catarina, C.P. 2250, Deleg. Azcapotzalco, México D.F., .Tel.: 5729 6000-64497. Email: rubio.josedejesus@gmail.com; jrubioa@ipn.mx

Figura 6: Constancia de satisfacción de la Estancia Académica.

Artículos en revisión

The screenshot displays the journal's header with the title 'Computación y Sistemas' and ISSN 2007-9737. A navigation menu includes HOME, ABOUT, USER HOME, SEARCH, CURRENT, ARCHIVES, and ANNOUNCEMENTS. The breadcrumb trail is 'Home > User > Author > Submissions > #4149 > Summary'. The main heading is '#4149 Summary', with sub-tabs for SUMMARY, REVIEW, and EDITING. The 'Submission' section lists authors (Virna V. Vela-Rincón, Dante Mújica-Vargas, Noé Alejandro Castro-Sánchez, Antonio Luna Álvarez, Arturo Rendón Castro), title ('Automatic Clustering Based on Hesitant Fuzzy Sets Theory'), original file ('4149-8540-4-SM.PDF' dated 2022-02-24), and submitter ('Virna V. Vela-Rincon'). The 'Status' section shows 'In Review' with initiation and modification dates of 2022-02-24. A 'Submission Metadata' section with an 'EDIT METADATA' link is also present.

Computación y Sistemas
ISSN 2007-9737
AN INTERNATIONAL JOURNAL OF COMPUTING SCIENCE AND APPLICATIONS

HOME ABOUT USER HOME SEARCH CURRENT ARCHIVES ANNOUNCEMENTS

Home > User > Author > Submissions > #4149 > Summary

#4149 Summary

SUMMARY REVIEW EDITING

Submission

Authors	Virna V. Vela-Rincón, Dante Mújica-Vargas, Noé Alejandro Castro-Sánchez, Antonio Luna Álvarez, Arturo Rendón Castro
Title	Automatic Clustering Based on Hesitant Fuzzy Sets Theory
Original file	4149-8540-4-SM.PDF 2022-02-24
Supp. files	None ADD A SUPPLEMENTARY FILE
Submitter	Virna V. Vela-Rincon
Date submitted	February 24, 2022 - 09:34 PM
Section	Articles
Editor	Alejandra Ramos

Status

Status	In Review
Initiated	2022-02-24
Last modified	2022-02-24

Submission Metadata

[EDIT METADATA](#)

Figura 7: Sometimiento de un artículo en la revista Computación y Sistemas.

Automatic Clustering Based on Hesitant Fuzzy Sets Theory

Virna V. Vela-Rincón¹, Dante Mújica-Vargas¹, Noé Alejandro Castro-Sánchez¹,
Antonio Luna Álvarez¹, Arturo Rendón Castro¹

¹ Tecnológico Nacional de México/CENIDET, Cuernavaca, Morelos
México

viryvela@cenidet.edu.mx, dante.mv@cenidet.tecnm.mx, noe.cs@cenidet.tecnm.mx,
jesus.luna18ce@cenidet.edu.mx, angel.rendon18ce@cenidet.edu.mx

Abstract. Hesitant fuzzy sets allow clustering data with multiple possible membership values for a single element of a reference set; this type of sets has been applied mainly in decision making problems, obtaining better results than other types of fuzzy sets. This research proposes a clustering algorithm based on the theory of hesitant fuzzy sets; the proposal is mathematically justified. Moreover, the algorithm is parameter-free, using the Calinski-Harabasz index to select the optimal number of clusters in which the data will be grouped, solving the initialization problem in clustering. Comparing the experimental results, on databases applied to pattern recognition, it is observed that the proposed algorithm is able to obtain a better performance compared to some of the state-of-the-art algorithms; it has the ability to converge in few iterations, in addition to doing it in less time than the comparison.

Keywords. Fuzzy clustering, Hesitant fuzzy sets, Pattern Recognition, Automatic clusters

1 Introduction

Clustering is an unsupervised learning task, which aims to divide a set of data according to their similarity, so that objects belonging to the same cluster are as similar as possible, while those belonging to different clusters are as different as possible [9]. This kind of learning, unsupervised, has been used in different fields, such as, machine learning, image segmentation, pattern recognition, among others [17, 2, 12, 23, 26, 7, 15, 34, 5]. There are different types of clustering algorithms, which have been divided into

density-based clustering, the most representative algorithm is DBSCAN [11] which considers that all cluster points must be connected by density. With the popularity of deep learning in recent years, network-based clustering algorithms have been successively proposed, like DSC [1] and DSCDAN, [32], however, this clustering is computationally expensive with many parameters to be configured. Another type of clustering is hierarchical clustering which organizes the data in a hierarchical structure, sometimes this can be combined with other types of clustering [20].

Finally, centroid-based clustering is one of the most intuitive clustering approaches, Fuzzy C-means (FCM) [3] is one of the most used to clustering; allows the gradual membership in a closed interval $\mu \in [0, 1]$ of the data with respect to the groups. This flexibility makes it possible to express the membership of a data in all groups simultaneously [4, 9]. FCM has been widely studied, as it has certain weaknesses such as falling into local minima, so some improvements have been proposed in the literature such as the FRFCM algorithm [15] that integrates morphological reconstruction to smooth the data, in addition to a filtering that modifies the data membership without relying on distance computation. Others, such as FCM-ELPSO [33], which combines FCM with an improved version of PSO, called ELPSO, that adopts a strategy of logarithmic inertia weights to provide a better balance between exploration and exploitation.

Figura 8: Primera página del artículo sometido a la revista *Computación y Sistemas*.

Engineering Applications of Artificial Intelligence
Fast and Automatic Hesitant Fuzzy Clustering applied to image segmentation
 --Manuscript Draft--

Manuscript Number:	
Article Type:	Research paper
Keywords:	Fuzzy clustering; Hesitant fuzzy Sets; Image segmentation; Calinski-Harabasz index
Corresponding Author:	Dante Mujica-Vargas CENIDET: Centro Nacional de Investigacion y Desarrollo Tecnológico Cuernavaca, Morelos MEXICO
First Author:	Virna Vela-Rincón
Order of Authors:	Virna Vela-Rincón Dante Mujica-Vargas José de Jesús Rubio Blanca Esthera Carvajal-Gómez Jean Marie Vianney Kinani
Abstract:	Image segmentation is a very studied area, looking for the best clustering of pixels. However, it is sometimes a complicated task, especially when these pixels are at the edges of regions, where there is a gradient and it is difficult to decide to which region to assign it. Hesitant fuzzy sets (HFS) allow to better describe these situations, allowing to have multiple possible values, giving more flexibility. In this research we propose a fast and automatic method based on hesitant fuzzy clustering (FAHFC), which besides being able to determine the number of clusters in which the segmentation is performed, proposes an alternative to build the HFS by using fuzzy relations (without relying on experts), as well as a step to speed up the processing time. The experiments show the superiority in terms of clustering quality and convergence with respect to some selected state-of-the-art algorithms.
Suggested Reviewers:	Jesús Olivares Mercado jolivares@ipn.mx Jorge Fuentes Pacheco jorge.fuentes@uaem.mx

Figura 9: Sometimiento de artículo en la revista *Engineering Applications of Artificial Intelligence*.

Fast and Automatic Hesitant Fuzzy Clustering applied to image segmentation

Virna V. Vela-Rincón^a, Dante Mújica-Vargas^{a,*}, José de Jesús Rubio^b, Jean Marie Vianney Kinani^c

^a*Tecnológico Nacional de México/CENIDET, Cuernavaca, Morelos, México.*

^b*Instituto Politécnico Nacional, SEPI-ESIME-Azcapotzalco, Ciudad de México, México.*

^c*Instituto Politécnico Nacional-UPIIH, San Agustín Tlaxiaca-Hidalgo, México.*

Abstract

Image segmentation is a very studied area, looking for the best clustering of pixels. However, it is sometimes a complicated task, especially when these pixels are at the edges of regions, where there is a gradient and it is difficult to decide to which region to assign it. Hesitant fuzzy sets (HFS) allow to better describe these situations, allowing to have multiple possible values, giving more flexibility. In this research we propose a fast and automatic method based on hesitant fuzzy clustering (FAHFC), which besides being able to determine the number of clusters in which the segmentation is performed, proposes an alternative to build the HFS by using fuzzy relations, as well as a step to speed up the processing time. The experiments show the superiority in terms of clustering quality and convergence with respect to some selected state-of-the-art algorithms.

Keywords: Fuzzy clustering, Hesitant fuzzy Sets, Image segmentation, Calinski-Harabasz index

1. Introduction

Image segmentation is an area of computer vision that has been studied for many years, with the objective of separating the image into more homogeneous regions that have similar characteristics, i.e., that belong to the same object [1]. It is a complicated task, because the images have different characteristics.

One of the most studied algorithms in this task is the Fuzzy C-means algorithm, which allows a fuzzy clustering analysis, in other words, each data is assigned a degree of membership in the interval $[0, 1]$ with respect to each of the groups or regions, simultaneously, in such a way that each data belongs to a certain degree to each of the groups [2, 3]. There are different algorithms based on the FCM algorithm, making improvements to achieve better segmentation quality performance, such as the FRFCM algorithm that employs morphological reconstruction to smooth images [4]. AFCF [5] is inspired by the image superpixels, the DP algorithm and the prior entropy-based fuzzy clustering.

*Corresponding author:

Email address: dantemv@cenidet.edu.mx (Dante Mújica-Vargas)

Preprint submitted to *Journal of L^AT_EX Templates*

February 24, 2022

Figura 10: Primera página del artículo sometido a la revista *Engineering Applications of Artificial Intelligence*.