

Universidad Autónoma del Estado de México
Centro Universitario UAEM Texcoco



Algoritmos para Segmentación Automática de
Conjunto de Imágenes Utilizando Algoritmos
Evolutivos

TESIS

Que para obtener el Título de
Doctor en Ciencias en Computación

Presenta

M. en C. Jared Cervantes Canales

Director de tesis

Dr. Jair Cervantes Canales

Co-director de tesis

Dr. Farid García Lamont

Texcoco, Estado de México, Mayo 2025

Abstract

The development of computer vision systems for image detection represents a fundamental endeavor across a wide range of scientific domains. Among the various processes involved, image segmentation stands out as a critical step, as it significantly influences the accuracy and efficiency of subsequent tasks. Achieving optimal segmentation parameters with minimal computational cost remains a key challenge, often addressed through the implementation of evolutionary algorithms.

This thesis explores the application of several evolutionary approaches, including Genetic Algorithms, Particle Swarm Optimization (PSO), and Differential Evolution, each of which operates over a broad spectrum of parameter ranges to determine the most effective segmentation strategies. To assess the performance of these algorithms, comprehensive experiments were conducted using publicly available datasets such as PlantVillage, Coffee Diseases, ICL, and LSM. These datasets encompass digital images from diverse domains and are designed for a variety of real-world applications.

Furthermore, a complementary method based on pixel-level characteristics was integrated into the study, yielding highly promising outcomes. The effectiveness of each segmentation technique was evaluated using standard performance metrics, including specificity, sensitivity, accuracy, Probabilistic Rand Index (PRI), Variation of Information (VOI), Global Consistency Error (GCE), and the Jaccard Index. The experimental findings highlight the robustness and efficiency of the proposed methods, particularly in scenarios requiring high precision and adaptability.

Resumen

El desarrollo de sistemas de visión artificial para la detección de imágenes representa un esfuerzo fundamental en una amplia gama de dominios científicos. Entre los diversos procesos involucrados, la segmentación de imágenes destaca como un paso crítico, ya que influye significativamente en la precisión y la eficiencia de las tareas posteriores. Lograr parámetros de segmentación óptimos con un coste computacional mínimo sigue siendo un reto clave, a menudo abordado mediante la implementación de algoritmos evolutivos.

Esta tesis explora la aplicación de diversos enfoques evolutivos, incluyendo algoritmos genéticos, optimización por enjambre de partículas (PSO) y evolución diferencial, cada uno de los cuales opera sobre un amplio espectro de rangos de parámetros para determinar las estrategias de segmentación más eficaces. Para evaluar el rendimiento de estos algoritmos, se realizaron experimentos exhaustivos utilizando conjuntos de datos públicos como PlantVillage, Coffee Diseases, ICL y LSM. Estos conjuntos de datos abarcan imágenes digitales de diversos dominios y están diseñados para diversas aplicaciones del mundo real.

Además, se integró en el estudio un método complementario basado en características a nivel de píxel, con resultados muy prometedores. La eficacia de cada técnica de segmentación se evaluó mediante métricas de rendimiento estándar, como la especificidad, la sensibilidad, la precisión, el Índice Probabilístico de Rand (PRI), la Variación de la Información (VOI), el Error de Consistencia Global (ECG) y el Índice de Jaccard. Los resultados experimentales destacan la robustez y la eficiencia de los métodos propuestos, especialmente en escenarios que requieren alta precisión y adaptabilidad.

Índice general

1. Introducción	15
1.1. Problemática	16
1.2. Justificación	17
1.3. Objetivos	18
1.4. Hipótesis	19
1.5. Estado del Arte	19
1.5.1. Segmentación con Algoritmos Genéticos	19
1.5.2. Segmentación Utilizando Optimización por Cúmulo de Partículas	21
1.6. Organización de la Tesis	23
2. Preliminares	25
2.1. Preprocesamiento de Imágenes	25
2.1.1. Filtro Promedio	25
2.1.2. Filtro de Mediana	25
2.1.3. Filtro Gaussiano	25
2.1.4. Ecualización de Histograma	26
2.1.5. CLAHE (Contrast Limited Adaptive Histogram Equalization)	26
2.1.6. Espacio de Color RGB (Red, Green, Blue)	26
2.1.7. Espacio de Color CMYK (Cyan, Magenta, Yellow, Key/Black)	27
2.1.8. Espacio de Color HSV (Hue, Saturation, Value)	27
2.1.9. Espacio de Color CIE Lab	28
2.2. Métodos de Segmentación	28
2.2.1. Métodos de Segmentación Basados en Umbral	33
2.2.2. Métodos de Segmentación Basados en Bordes	34
2.2.3. Métodos de Segmentación Basados en Agrupamiento	37
2.2.4. Métodos de Segmentación Basados en Regiones	38
2.2.5. Métodos de Segmentación Basados en IA	39
2.3. Algoritmos Evolutivos	40
2.3.1. Ventajas de los Algoritmos Evolutivos	43
2.4. Algoritmos Genéticos	43
2.4.1. Población y Selección de Individuos	45
2.4.2. Cruza	46

2.4.3. Mutación	47
2.4.4. Condición de Paro	47
2.5. Optimización por Cúmulos de Partículas	48
3. Segmentación Basada en Algoritmos Genéticos	51
3.1. Parámetros optimizados	51
3.1.1. CLAHE	51
3.1.2. Espacios de Color	52
3.1.3. Canal G	53
3.1.4. Subconjunto de Datos Seleccionados	54
3.1.5. Filtro Gaussiano	54
3.1.6. Filtro Promedio	54
3.1.7. PCA	55
3.1.8. Métodos de Segmentación	55
3.1.9. Operadores Morfológicos	56
3.2. Algoritmo Genético	57
3.2.1. Codificación Genética	63
4. Segmentación Basada en PSO	65
4.1. Componentes del Algoritmo propuesto	65
4.1.1. Espacios de color	66
4.1.2. Canales de color	67
4.1.3. Pre procesamiento	67
4.1.4. Segmentación	69
4.1.5. Operadores morfológicos	69
4.2. Algoritmo propuesto	69
4.3. Exploración y Explotación	72
4.4. Factor de la inercia	72
4.5. Coeficientes de aceleración	73
5. Segmentación Basada en Caracterización de Píxeles	75
5.1. Mejora de la calidad de imagen	75
5.2. Extracción de características multispectrales	76
5.3. Incorporación de características estructurales (bordes)	77
5.4. Entrenamiento del clasificador	78
5.5. Segmentación de nuevas imágenes	78
5.6. Consideraciones finales	78
6. Resultados Experimentales	79
6.1. Métricas de Desempeño	79
6.1.1. Exactitud	80
6.1.2. Sensibilidad	80
6.1.3. Especificidad	80
6.1.4. Probabilistic Rand Index (PRI)	81
6.1.5. Variation of Information (VOI)	81
6.1.6. Global Consistency Error (GCE)	82

6.2. Base de Datos	82
6.3. Hardware y software	87
6.4. Resultados experimentales	87
6.4.1. Resultados con el conjunto de datos ICL	87
6.4.2. Resultados con el conjunto de datos Enfermedades del café	89
6.4.3. Resultados con el conjunto de datos Plant Village	92
6.4.4. Resultados con el conjunto de datos LSM	93
6.4.5. Resultados con el conjunto de datos Celulas	96
7. Conclusiones	99
A. Artículos publicados	111
B. Tablas de resultados	115

Lista de Figuras

2.1. Procesos de un sistema de visión	29
2.2. Segmentación Semántica. a) La imagen original contiene una mesa y ciertos objetos sobre de ella, tres libros y un tazón. b) En el proceso son detectados todos los objetos y se asocian aquellos que pertenecen a una misma clase. c) Se segmentan de un mismo color los objetos pertenecientes a una clase.	31
2.3. Segmentación por instancias. a) La imagen original contiene una mesa y ciertos objetos sobre de ella, tres libros y un tazón. b) En el proceso son detectados todos los objetos en la imagen. c) Se segmentan con distintos colores todos los objetos.	31
2.4. Segmentación binaria. a) La imagen original contiene una mesa y ciertos objetos sobre de ella, tres libros y un tazón. b) En el proceso son detectados los objetos de interes y al mismo tiempo el fondo. c) Se segmentan con un solo color todos los objetos de interes que se distinguen del fondo.	32
2.5. Clasificación de los diferentes tipos de Segmentación	33
2.6. Métodos de Optimización	42
2.7. Representación gráfica de las expresiones que se usa en la genética	45
2.8. Representación gráfica de cruce a un punto	47
2.9. Operador de Mutación	47
2.10. Proceso General del Algoritmo PSO	48
3.1. Segmentación Usando Diferentes Espacios de Color	53
3.2. Técnicas del AG	57
3.3. Método Propuesto AG	58
3.4. Proceso de Selección por el método de Ruleta	60
3.5. Operador de cruce de dos puntos en un Algoritmo genético	61
3.6. Operador de Mutación en un algoritmo genético	62
3.7. Codificado de las variables utilizadas del algoritmo propuesto	63
4.1. Segmentación usando PSO	66
6.1. Elementos de las Métricas de Desempeño	80
6.2. Base de datos ICL	83

6.3. Ground Truth de la base de datos ICL	83
6.4. Base de datos Coffee Diseases	84
6.5. Ground Truth de la base de datos Coffee Diseases	84
6.6. Base de datos Plat Village	85
6.7. Ground Truth de la base de datos Plant Village	85
6.8. Base de datos LSM	86
6.9. Ground Truth de la base de datos LSM	86
6.10. Base de datos LSM	86
6.11. Ground Truth de la base de datos LSM	87
6.12. Resultados de la base de datos ICL	89
6.13. Resultados de la base de datos enfermedades de Cafe	91
6.14. Resultados de la base de datos enfermedades de Cafe	93
6.15. Resultados de la base de datos LSM	96
6.16. Resultados de la base de datos Celulas	98
A.1. Neurocomputing 2023	112
A.2. ESWA 2024	113
A.3. ICIC 2022	114

Índice de Tablas

2.1. Algoritmo K-means	38
2.2. Algoritmo Genético Básico.	44
2.3. Algoritmo de la Técnica de Selección: Ruleta.	46
2.4. Algoritmo Optimización por Cúmulos de Partículas.	50
3.1. Parámetros optimizados con el algoritmo genético propuesto. . .	52
3.2. Algoritmo de Selección Ruleta.	60
4.1. Comportamiento de las partículas con respecto a los valores de los coeficientes.	74
5.1. Espacios de color utilizados en la extracción de características . .	77
6.1. Comparación de Métodos de Segmentación	88
6.2. Comparación de resultados por método de segmentación	90
6.3. Interpretación Técnica por Método de Segmentación	91
6.4. Resultados promedio por método de segmentación	92
6.5. Comparación del desempeño de los algoritmos	94
6.6. Balance entre sensibilidad y especificidad	94
6.7. Selección de métodos por necesidad	95
6.8. Comparación de rendimiento entre métodos de segmentación . .	97
6.9. Resumen promedio de resultados y aplicaciones ideales	98
B.1. Resultados detallados del método de PCA para el conjunto de datos ICL	116
B.2. Resultados detallados del método de Otsu para el conjunto de datos ICL	117
B.3. Resultados detallados del método de AG para el conjunto de da- tos ICL	118
B.4. Resultados detallados del método de PSO para el conjunto de datos ICL	119
B.5. Resultados detallados del método de Caracterización de Píxeles para el conjunto de datos ICL	120
B.6. Resultados del método de segmentación PCA con el conjunto de datos Coffee diseases	121

B.7. Resultados del método de segmentación Otsu con el conjunto de datos Coffee diseases	122
B.8. Resultados del método de segmentación GA con el conjunto de datos Coffee diseases	123
B.9. Resultados del método de segmentación PSO con el conjunto de datos Coffee diseases	124
B.10. Resultados del método de segmentación CP con el conjunto de datos Coffee diseases	125
B.11. Resultados del método de segmentación PCA con el conjunto de datos Plant Village	126
B.12. Resultados del método de segmentación OTSU con el conjunto de datos Plant Village	127
B.13. Resultados del método de segmentación GA con el conjunto de datos Plant Village	128
B.14. Resultados del método de segmentación PSO con el conjunto de datos Plant Village	129
B.15. Resultados del método de segmentación CP con el conjunto de datos Plant Village	130
B.16. Resultados del método de segmentación PCA con el conjunto de datos LSM	131
B.17. Resultados del método de segmentación Otsu con el conjunto de datos LSM	132
B.18. Resultados del método de segmentación GA con el conjunto de datos LSM	133
B.19. Resultados del método de segmentación PSO con el conjunto de datos LSM	134
B.20. Resultados del método de segmentación CP con el conjunto de datos LSM	135

Capítulo 1

Introducción

En los últimos años, el uso de sistemas de visión y técnicas de procesamiento digital de imágenes ha experimentado una proliferación significativa en una amplia gama de campos de aplicación. Desde áreas como ingeniería civil [83], medicina [39] [28] [90], seguridad [85], Astronomía [76], robótica [56] y agricultura [56], estos sistemas y técnicas se han convertido en herramientas fundamentales para procesar y analizar información crucial en la toma de decisiones.

A pesar de los avances notables en las técnicas de procesamiento de imágenes digitales en los últimos años, persisten varios desafíos cuando se trata de su implementación.

Los sistemas de visión están compuestos por múltiples etapas, cada una de las cuales se lleva a cabo utilizando una técnica específica que depende de una serie de variables. Aspectos como el preprocesamiento, las variables de las técnicas de segmentación, la iluminación, la intensidad de las imágenes e incluso operaciones morfológicas como la dilatación o la erosión, que a menudo se utilizan como procesamiento finales, pueden tener un impacto significativo en la calidad de la segmentación.

La segmentación en uno de sus diversos conceptos, se puede definir como el proceso de dividir o agrupar la imagen en múltiples regiones. En esta imagen dividida se puede encontrar muchas regiones si es una segmentación compleja, donde hay varias regiones de interés y el fondo de la imagen o si es una segmentación básica, donde podemos encontrar solo dos regiones, la región de interés y el fondo de la imagen. Este procedimiento facilita el análisis de cualquier imagen, mostrando las secciones que son representativas y de interés, y eliminando las secciones que no contengan información relevante.

En esencia, la segmentación implica etiquetar cada píxel de una imagen de entrada, agrupándolos en regiones homogéneas y conectadas que comparten características y propiedades similares. Estas regiones homogéneas se definen por propiedades como la intensidad, textura y color de los píxeles [34] [31].

Actualmente, existen diversos métodos de segmentación, cada uno con sus propias variables y ajustes. Elegir todos los componentes y parámetros para segmentar un determinado conjunto de datos no es un trabajo sencillo. Distin-

tos factores pueden influenciar en la calidad de la segmentación como lo son los preprocesamientos y post procesamientos, cambios de intensidad, variaciones en la iluminación y demás. En muchos casos, se requiere la intervención de un experto en el campo para evaluar exhaustivamente todos los aspectos del conjunto de datos, aplicar su experiencia en experimentos pertinentes y seleccionar las variables adecuadas. No obstante, el desafío surge cuando nos enfrentamos a conjuntos de datos con un número significativo de variables, lo que resulta en una complejidad combinatoria abrumadora. Además, no existe una solución universal que pueda aplicarse a todos los conjuntos de datos, ya que cada uno presenta sus propias particularidades y desafíos únicos.

Es precisamente debido a esta complejidad y diversidad que se vuelve imperativo recurrir a algoritmos para la segmentación automática de imágenes. Estos algoritmos pueden analizar eficientemente conjuntos de datos complejos y grandes, identificar patrones relevantes y segmentar imágenes de manera precisa y efectiva. Al utilizar estos algoritmos, podemos superar las limitaciones impuestas por la complejidad combinatoria y obtener resultados consistentes y confiables en la segmentación de imágenes.

En este contexto, los algoritmos evolutivos han surgido como una herramienta fundamental y revolucionaria para resolver problemas de optimización y toma de decisiones. Los algoritmos evolutivos han sido la base de un gran número de investigaciones donde se han desarrollado muchas aplicaciones [45] [49] [70] [53] [81] [82]. Los podemos encontrar aplicándose en todas las disciplinas y campos de la ciencia, incluyendo la economía, el diseño de estructuras, modelado, bioinformática, maquinaria, mediciones, modelado, automatización y demás. [45] [49].

Es fundamental destacar que estos algoritmos han sido aplicados con éxito en una amplia gama de situaciones del mundo real, podemos agilizar procesos, mejorar la precisión de los diagnósticos médicos, optimizar la producción agrícola, fortalecer la seguridad en entornos críticos y mucho más. En esencia, facilitan la transformación de datos visuales en conocimientos útiles y accionables, lo que nos capacita para tomar decisiones informadas y eficaces en una variedad de contextos del mundo real.

El objetivo principal de esta investigación es desarrollar algoritmos evolutivos para la segmentación de conjuntos de imágenes. Se proponen algoritmos automáticos para la selección óptima de variables en las etapas de preprocesamiento, procesamiento y post procesamiento. Estas técnicas serán descritas, comparadas y evaluadas utilizando conjuntos de datos actuales. La tesis detalla la justificación, los objetivos, el estado del arte y presenta la metodología propuesta, junto con los resultados y conclusiones obtenidas.

1.1. Problemática

Todo proceso tiene un camino hacia su culminación, y cada etapa de ese camino es tan crucial como el resultado final. La eficacia de un desarrollo está directamente ligada a la calidad del resultado obtenido. Por ende, podemos

afirmar que el proceso de desarrollo y su resultado final son dos aspectos que guardan una relación proporcional directa.

La detección de objetos es el resultado de un proceso que abarca diversas fases, cada una de las cuales se centra en la segmentación como piedra angular para lograr una extracción de características precisa y relevante.

No obstante, alcanzar una segmentación óptima es una tarea ardua. Requiere, en muchos casos, un profundo conocimiento de las técnicas de segmentación y los preprocesamientos de imágenes asociados. La obtención de un conjunto adecuado de variables y procedimientos para una segmentación efectiva solo se adquiere a través del estudio continuo, la dedicación y la experiencia acumulada en el tema. Además, es esencial experimentar con una amplia gama de situaciones para llegar a una solución satisfactoria, teniendo en cuenta el alto número de combinaciones posibles que implican estos experimentos (un vasto espacio de soluciones). Por último, no existe una solución universal que se aplique a todos los conjuntos de datos; se requiere un nuevo enfoque cada vez que cambia el tema o el conjunto de imágenes a analizar.

Por estas razones, resulta crucial el desarrollo de técnicas automáticas de segmentación. En un trabajo previo, se han diseñado dos algoritmos para la selección automática de técnicas de segmentación y sus variables asociadas. En esta tesis, el objetivo es profundizar en la investigación utilizando técnicas de segmentación más avanzadas, como se han propuesto en estudios recientes [42] [99] [91] [33], y además, emplear diferentes algoritmos evolutivos para mejorar la optimización [7] [80] [66]. Finalmente, se compararán los resultados obtenidos con las técnicas de segmentación de imágenes más actuales [5] [99] [41] [67].

1.2. Justificación

La segmentación de imágenes representa una etapa crítica en el procesamiento digital de imágenes y en los sistemas de visión, cuyo estudio se ha intensificado en los últimos tiempos para abordar una amplia gama de problemas en nuestra vida cotidiana. La calidad de un sistema de visión se ve directamente afectada por la precisión de la segmentación de imágenes. Por lo tanto, el avance, la mejora y la aplicación de nuevos algoritmos en este ámbito son de suma importancia.

El desarrollo de algoritmos de segmentación automáticos ofrece una ventaja significativa al reducir el tiempo y la complejidad asociados con el trabajo de un experto en el tema. Estos algoritmos permiten una manipulación más fácil y rápida de conjuntos de imágenes sin requerir un conocimiento exhaustivo del tema. Además, el uso de algoritmos automáticos podría contribuir a mejorar la segmentación incluso en conjuntos de datos con condiciones de iluminación y fondos similares, lo que resultaría en una segmentación óptima y más confiable.

Para lograr un modelo adecuado para la segmentación de un conjunto de imágenes, es imprescindible comprender la influencia de cada paso del proceso, desde las etapas previas, durante y posteriores a la segmentación. Es necesario entender cómo afectan las variables de manera individual y en combinación con

otras al obtener una segmentación precisa y coherente. Este enfoque holístico garantiza que se seleccionen y ajusten los parámetros óptimos para cada situación específica, lo que resulta en una segmentación efectiva y de alta calidad.

1.3. Objetivos

El objetivo general de esta tesis es desarrollar algoritmos avanzados y eficaces para la segmentación automática de conjuntos de imágenes, utilizando como base los principios y técnicas de los algoritmos evolutivos. Este objetivo implica la investigación, el diseño y la implementación de métodos innovadores que permitan una segmentación precisa y robusta de una amplia variedad de imágenes, abordando desafíos como la variabilidad en las condiciones de iluminación, la presencia de ruido y la complejidad de las estructuras visuales.

Se pretende alcanzar los siguientes objetivos específicos:

1. Desarrollar e implementar técnicas para cálculo de aptitud para múltiples regiones de segmentación.
2. Desarrollar algoritmos para segmentación binaria de conjuntos de datos utilizando no solo algoritmos básicos de segmentación, sino los algoritmos actuales para segmentación binaria.
3. La implementación de algoritmos de segmentación automática, como los Algoritmos Genéticos, la Optimización por Cúmulos de Partículas (PSO) y la segmentación basada en características de píxeles, resulta crucial en el campo de la visión por computadora.
4. Obtener un análisis comparativo con las técnicas más actuales utilizadas para segmentación de imágenes.
5. Implementar los algoritmos desarrollados en diferentes conjuntos de datos y obtener un comparativo de los resultados.

1.4. Hipótesis

¿Se puede mejorar la calidad de la segmentación de conjunto de imágenes en comparación con las técnicas más modernas implementando algoritmos automáticos de segmentación?

Se plantea que, si se ajustan adecuadamente los parámetros del algoritmo evolutivo, se logrará una segmentación mejorada y más precisa de los conjuntos de imágenes, lo que resultará en una mejor identificación y extracción de características relevantes en los datos visuales.

1.5. Estado del Arte

En los últimos años, uno ha sido testigo de un notable avance en el desarrollo de algoritmos evolutivos destinados a la segmentación de imágenes. Este progreso ha dado lugar a una amplia gama de técnicas innovadoras y sofisticadas que han revolucionado nuestra capacidad para analizar y procesar datos visuales con precisión y eficacia. La diversidad y la complejidad de estos algoritmos reflejan el dinamismo y la vitalidad de la investigación en este campo.

En el contexto de esta tesis, se ha llevado a cabo una exhaustiva exploración de las múltiples técnicas desarrolladas, las cuales pueden ser clasificadas en diferentes categorías en función de sus enfoques y características distintivas. Esta clasificación proporciona una estructura organizativa que nos permite comprender y analizar de manera sistemática las diversas estrategias empleadas para abordar el desafío de la segmentación de imágenes.

Al categorizar las técnicas desarrolladas en esta investigación, podemos identificar patrones, tendencias y áreas de innovación que nos ayudan a comprender mejor el panorama actual de la segmentación de imágenes basada en algoritmos evolutivos. Además, esta clasificación nos brinda una base sólida para evaluar y comparar el rendimiento de las diferentes técnicas en función de criterios específicos, lo que nos permite determinar cuáles son más adecuadas para aplicaciones particulares o escenarios específicos. En resumen, esta tesis representa un importante paso adelante en nuestro entendimiento y aprovechamiento de los algoritmos evolutivos para la segmentación de imágenes, proporcionando una contribución valiosa al campo y sentando las bases para futuras investigaciones y desarrollos en esta área emocionante y en constante evolución.

1. Segmentación con algoritmos genéticos
2. Segmentación utilizando optimización por cúmulo de partículas
3. Segmentación basada en caracterización de píxeles

1.5.1. Segmentación con Algoritmos Genéticos

La aplicación de algoritmos evolutivos ha sido de gran utilidad en el campo de la inteligencia artificial, con una variedad de trabajos que exploran su uso en

diferentes contextos. Desde la optimización de parámetros hasta la mejora de sub-técnicas, se han encontrado diversas aplicaciones en la literatura.

Por ejemplo, en el artículo de Khan et al. [51], se presenta un enfoque que combina un algoritmo genético con la técnica de segmentación fuzzy C-means para segmentar imágenes. El objetivo principal del algoritmo genético es encontrar el cromosoma que contenga los centroides de los k segmentos. Este enfoque se distingue por dos características particulares. En primer lugar, utiliza un mapa autoorganizado para determinar el número de segmentos y la longitud del cromosoma. En segundo lugar, el proceso de inicialización aleatoria no es completamente aleatorio; en cambio, se emplea una estrategia basada en la oposición, donde se seleccionan tanto los mejores individuos como sus opuestos. Esto se hace con el fin de evitar caer en mínimos locales y promover una exploración más amplia del espacio de búsqueda.

En el estudio realizado por Xie et al. [92], se lleva a cabo la segmentación automática de imágenes de dermatoscopia. Para lograrlo, se emplea un enfoque que utiliza un algoritmo genético como paso inicial para obtener muestras óptimas de semillas. Estas muestras se utilizan para generar un bosque neuronal autogenerado, el cual se emplea para entrenar el resto de las muestras. En este proceso, las redes neuronales autogeneradas son las responsables de la segmentación propiamente dicha, mientras que el algoritmo genético actúa como un preproceso para la segmentación.

Por otro lado, en el trabajo presentado por Nematzadeh et al. [65], se propone un enfoque que utiliza múltiples algoritmos genéticos de manera consecutiva. La razón de emplear varios algoritmos radica en la utilización de diferentes longitudes de cromosomas en cada uno de ellos, lo que permite buscar el número óptimo de superpíxeles y las características principales ideales en una imagen de entrada para la explicación automática. El proceso implica ejecutar los algoritmos genéticos en secuencia, y luego recopilar los resultados a través de un consenso y votación mayoritaria. El objetivo final de este enfoque es mejorar el método LIME (Explicaciones Model-agnostic Interpretable Locales, por sus siglas en inglés) y, en última instancia, lograr una segmentación precisa de las imágenes de melanoma.

La investigación realizada por Ghosh et al. [36] introduce un enfoque innovador que incorpora un algoritmo genético para la segmentación de la próstata en imágenes de tomografía computarizada pélvica. Este algoritmo aprovecha la información previa aprendida sobre las texturas, formas y posiciones relativas de los órganos en las imágenes. La metodología fue evaluada con éxito en el contexto de la segmentación de la próstata, demostrando su efectividad en este desafío particular.

Por otro lado, en el trabajo de Abdel et al. [2], se aborda el problema de la segmentación de imágenes como un ejercicio de optimización en lugar de simplemente aplicar umbralización. Se propone un enfoque que considera el histograma de niveles de gris de una imagen como una distribución de probabilidad, y se utilizan las entropías de Tsallis y Renyi como funciones de aptitud para maximizar. Esta aproximación muestra una perspectiva novedosa y efectiva para abordar la segmentación de imágenes.

Además, en el estudio presentado por Yue et al. [97], se introduce un algoritmo híbrido de murciélago que combina características clave de los algoritmos genéticos con un método de umbralización multinivel. Este enfoque utiliza la capacidad de cruzamiento de los algoritmos genéticos para seleccionar umbrales óptimos, lo que demuestra una mejora significativa en la precisión de la segmentación.

Por último, en el artículo de Hung et al. [44], se propone un enfoque que combina el algoritmo Fuzzy C-means con un algoritmo genético para la segmentación de resonancias magnéticas cerebrales. El algoritmo genético se utiliza para encontrar los mejores individuos que representan las posiciones óptimas de los centroides de los grupos, lo que resulta en una segmentación más precisa y mejorada en comparación con otros métodos de agrupamiento. Es importante destacar que este enfoque se centra exclusivamente en métodos de segmentación de agrupamiento, sin incluir técnicas de preprocesamiento ni operadores morfológicos en la comparación.

En este análisis del estado del arte, se identificó un estudio que guarda similitudes con el método propuesto, el artículo es de Tosta et al. [86], ya que incluye un preprocesamiento, un proceso de segmentación intermedio y un postprocesamiento. En el estudio mencionado se implementa un ecualizador de histograma en los canales de color del modelo RGB, seguido por un filtro gaussiano para eliminar el ruido de la imagen. Posteriormente, se emplea la técnica de entropía difusa de tres particiones para definir los valores de cada individuo. Luego, se utiliza un algoritmo genético para determinar el valor óptimo de umbral, finalizando con operadores morfológicos como dilatación y apertura.

Aunque hay diversos estudios que utilizan diferentes algoritmos genéticos, no se han encontrado investigaciones que aborden la segmentación óptima incorporando una variedad de parámetros, como diferentes técnicas de preprocesamiento, distintos métodos de segmentación y técnicas de refinamiento en el proceso de acabado. Este vacío en la literatura resalta la singularidad y la innovación del enfoque propuesto, el cual integra múltiples etapas y considera una amplia gama de factores para lograr una segmentación óptima y precisa.

1.5.2. Segmentación Utilizando Optimización por Cúmulo de Partículas

El uso del algoritmo de optimización por enjambre de partículas (PSO) ha sido fundamental para el avance de sistemas de aprendizaje automático en la tarea de segmentación visual. Una de las investigaciones más recientes en este campo es la de Chen et al. [24], quienes introdujeron una estrategia de élite y mejoraron la optimización integrada del PSO para encontrar el umbral de segmentación óptimo en imágenes. Del mismo modo, Zhang et al. [100] emplearon el PSO en combinación con la agrupación híbrida de K-medias para segmentar datos de productores agrícolas en el espacio de color YCbCr, mitigando así los efectos de la iluminación baja y las sombras. Esta combinación tiene como objetivo abordar las limitaciones de velocidad de convergencia del PSO y la determinación del centro de agrupación óptimo inicial.

Otro enfoque innovador se presenta en el artículo de Feng et al. [32], donde se propone una combinación de técnicas para reducir el costo computacional. En este estudio, se utiliza el PSO para obtener un vector umbral que define tanto la intensidad de los píxeles como la intensidad promedio local de los píxeles, basándose en un enfoque de umbral de imagen que maximiza la entropía del histograma en escala de grises 2-D, especialmente útil en imágenes infrarrojas.

Además, el trabajo de Maitra et al. [60] presenta una variante mejorada del PSO llamada HCOCLPSO, que emplea un enfoque híbrido al clonar las partículas más aptas. Este método híbrido utiliza tanto el aprendizaje cooperativo como el aprendizaje integral para abordar el problema de la dimensionalidad, superando así las limitaciones tradicionales del PSO.

La investigación realizada por Chander et al. [22] presenta un esquema innovador de segmentación utilizando la técnica de Otsu. En este esquema, los umbrales resultantes del proceso iterativo se consideran como umbrales principales iniciales, alrededor de los cuales se generan partículas de manera aleatoria. El algoritmo PSO propuesto aporta una contribución significativa al adaptar ciertos complementos, como el impulso y la interacción social, en las actualizaciones de movimiento entre las partículas.

Asimismo, en el trabajo de Zhang et al. [101], se propone la sustitución de la distancia euclidiana por la distancia de Mahalanobis dentro del algoritmo de agrupamiento posibilitado de c-medias. Además, se optimizan los centros de agrupamiento iniciales utilizando el algoritmo evolutivo de optimización por enjambre de partículas. Por otro lado, Pham et al. [71] destacan por su contribución en la segmentación de imágenes cerebrales de resonancia magnética. Su enfoque se centra en la implementación de una nueva función objetivo que emplea agrupación de entropía difusa kernelizada con corrección de sesgo e información espacial local, la cual se incorpora al algoritmo PSO para mejorar el proceso de segmentación.

En este campo, también destaca el artículo de Borjigin et al. [8], donde proponen inicialmente un modelo de umbral multinivel basado en el histograma de niveles de gris. Luego, validan la formulación de los umbrales utilizando un método de inducción matemática y, finalmente, emplean la optimización por enjambre de partículas para obtener los valores de umbral óptimos. Por otro lado, el trabajo de Chakraborty et al. [21] presenta una contribución notable al proponer un enfoque dinámico de PSO para determinar los centroides de un grupo inicial, considerando una función objetivo que incorpora el valor de membresía de cada píxel junto con sus vecinos cercanos. Además, aplican una nueva transformación algebraica a la función objetivo. Como una adición interesante, los autores utilizan un mecanismo de aislamiento basado en gráficos para refinar los resultados de las imágenes segmentadas.

Como se evidencia en estas investigaciones, se ha logrado un avance significativo en el uso del algoritmo PSO. Sin embargo, hasta ahora, los métodos propuestos han sido principalmente mejoras al algoritmo original y han estado complementados con una única técnica de segmentación.

1.6. Organización de la Tesis

A continuación, se describe la organización del resto de la Tesis.

El Capítulo 2 ofrece una visión general de los métodos de segmentación y una posible clasificación, así como el marco teórico de los algoritmos evolutivos (incluyendo el algoritmo genético básico y optimización por cumulo de partículas).

El Capítulo 3 forma parte de la metodología empleada en la tesis, explicando el desarrollo y los elementos que componen los dos algoritmos genéticos creados, así como las diferencias entre ellos y sus contribuciones.

El Capítulo 4 aborda el complemento de los algoritmos genéticos, que es la optimización por cúmulo de partículas, cubriendo todos los procesos aplicados y las diversas combinaciones utilizadas. Se detalla el algoritmo utilizado para resolver problemas de segmentación.

El Capítulo 5 describe en detalle la metodología propuesta para un algoritmo adicional. Aunque no pertenece a los algoritmos evolutivos, ha demostrado un excelente rendimiento. Se explica un método para clasificar los píxeles según sus características.

El Capítulo 6 presenta exhaustivamente los resultados experimentales obtenidos con los diferentes conjuntos de datos utilizados, comparando la efectividad con otras técnicas de segmentación mediante el análisis de las métricas de desempeño.

Finalmente, **El Capítulo 7** resume el documento en su totalidad, destacando las conclusiones más relevantes obtenidas durante la experimentación. Además, se proponen ideas finales para futuros trabajos.

Capítulo 2

Preliminares

2.1. Preprocesamiento de Imágenes

El preprocesamiento constituye una etapa fundamental en el análisis de imágenes digitales, orientada a mejorar la calidad de las imágenes mediante la atenuación de ruido, realce de características y normalización, con el objetivo de optimizar las fases posteriores de procesamiento y análisis computacional.

2.1.1. Filtro Promedio

El filtro promedio o *mean filter* es una técnica lineal de suavizado que reduce el ruido mediante el cálculo de la media aritmética de los valores de píxeles en una vecindad definida por una ventana (por ejemplo, 3×3 , 5×5 , etc.). Este valor se asigna al píxel central. A mayor tamaño de ventana, mayor es el efecto de suavizado, aunque con riesgo de pérdida de detalles finos.

2.1.2. Filtro de Mediana

El filtro de mediana es una técnica no lineal que reemplaza el valor del píxel central por la mediana de los valores en su vecindad. Es particularmente efectivo para eliminar ruido impulsivo, comúnmente conocido como ruido "sal y pimienta". A diferencia del filtro promedio, preserva mejor los bordes de la imagen.

2.1.3. Filtro Gaussiano

Este filtro utiliza una función gaussiana para asignar pesos a los píxeles dentro de la ventana, privilegiando aquellos cercanos al centro. El suavizado gaussiano proporciona una supresión de ruido más natural y mantiene mejor las estructuras relevantes. El núcleo se define por:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

donde σ controla la desviación estándar de la distribución.

2.1.4. Ecualización de Histograma

La ecualización de histograma es una técnica que redistribuye las intensidades de los píxeles para aumentar el contraste global de la imagen. Este proceso resulta en una representación más uniforme de las intensidades, facilitando la detección de características en áreas oscuras o sobreexpuestas.

2.1.5. CLAHE (Contrast Limited Adaptive Histogram Equalization)

CLAHE es una extensión adaptativa de la ecualización de histograma. Divide la imagen en regiones pequeñas y aplica la ecualización localmente, seguida de una interpolación bilineal para evitar discontinuidades entre regiones. Se introduce un parámetro denominado *clip limit* para restringir la amplificación del contraste, evitando así la exageración del ruido en regiones homogéneas. Este método es altamente eficaz para mejorar el contraste en imágenes con variabilidad espacial en la iluminación.

Espacios de color

Un *espacio de color* se define como una representación matemática del espectro cromático, estructurada de manera que permita su procesamiento, análisis y visualización por parte de sistemas digitales de adquisición, visualización e impresión, tales como cámaras, monitores y dispositivos de impresión. Cada espacio de color organiza los colores en función de la combinación de componentes o *canales*, los cuales codifican la intensidad de atributos específicos del color. La elección de un espacio de color está directamente relacionada con el contexto de aplicación, debido a que cada uno presenta ventajas y limitaciones inherentes a su estructura y percepción.

2.1.6. Espacio de Color RGB (Red, Green, Blue)

El modelo RGB es un espacio de color aditivo ampliamente utilizado. En este modelo, el color se genera mediante la suma de intensidades de los componentes rojo (R), verde (G) y azul (B), cada uno cuantificado en un rango de 0 a 255. La combinación completa de estos componentes permite representar hasta $2^{24} = 16,777,216$ colores distintos. Por ejemplo, el color negro se representa como (0, 0, 0), mientras que el blanco idealmente como (255, 255, 255). Este espacio es prevalente en pantallas electrónicas debido a su naturaleza emisiva.

2.1.7. Espacio de Color CMYK (Cyan, Magenta, Yellow, Key/Black)

El espacio CMYK es un modelo sustractivo, utilizado principalmente en impresión. A diferencia de RGB, en CMYK los colores se obtienen mediante la absorción de ciertas longitudes de onda de la luz blanca, aplicando tintas cian (C), magenta (M), amarilla (Y) y negra (K). Cada componente absorbe una parte específica del espectro: C absorbe el rojo, M el verde y Y el azul, mientras que K complementa la generación de tonos oscuros al absorber la mayoría del espectro. Por ejemplo, el negro se representa con $(0, 0, 0, 100)$ indicando una cobertura completa de tinta negra.

- **Cian (C)**: Opuesto al rojo. Absorbe la luz roja y refleja verde y azul.
- **Magenta (M)**: Opuesto al verde. Absorbe verde y refleja rojo y azul.
- **Amarillo (Y)**: Opuesto al azul. Absorbe azul y refleja rojo y verde.
- **Negro (K)**: Se añade para obtener un negro puro y absorber casi toda la luz.

2.1.8. Espacio de Color HSV (Hue, Saturation, Value)

El modelo HSV es una representación perceptual del color, más alineada con la interpretación humana. Se compone de tres parámetros:

- **Hue (Matiz)**: define el tipo de color en una escala circular de 0° a 360° , donde 0° es rojo, 120° es verde, y 240° es azul.
- **Saturation (Saturación)**: mide la pureza del color, de 0 % (gris) a 100 % (color puro).
- **Value (Valor)**: cuantifica el brillo del color, desde 0 % (negro) hasta 100 % (máxima luminosidad).

Este modelo se utiliza frecuentemente en aplicaciones de procesamiento de imágenes por su congruencia con la percepción visual humana.

CIE XYZ

Modelo matemático desarrollado por la CIE. Representa la percepción humana del color en tres dimensiones:

- **X**: Mezcla de rojo y verde.
- **Y**: Brillo o luminosidad.
- **Z**: Mezcla de azul y verde.

Este modelo es fundamental en la conversión entre espacios de color y en estudios de percepción visual. Es independiente del dispositivo (pantalla, impresora, etc.).

2.1.9. Espacio de Color CIE Lab

El modelo CIE Lab es un espacio perceptual diseñado para aproximarse a la visión humana, caracterizado por su uniformidad perceptual. Consta de tres componentes:

- L^* : luminosidad, con valores de 0 (negro) a 100 (blanco).
- a^* : eje verde-rojo, de negativo (verde) a positivo (rojo).
- b^* : eje azul-amarillo, de negativo (azul) a positivo (amarillo).

Este espacio es ampliamente utilizado en la ciencia del color, control de calidad e industria gráfica, dada su independencia del dispositivo y su fidelidad perceptual.

Cada espacio de color está compuesto por canales, los cuales representan dimensiones individuales del modelo de color. La naturaleza de estos canales varía entre espacios: por ejemplo, en RGB cada canal varía entre 0 y 255, mientras que en HSV el canal Hue varía entre 0° y 360° , y los canales Saturation y Value entre 0 % y 100 %. La combinación específica de los valores de los canales determina el color final percibido.

2.2. Métodos de Segmentación

Un sistema de visión está conformado por diversos procesos y cada uno de ellos son indispensables para poder obtener una buena detección de objetos. Estos procesos forman una cadena de procedimientos en la que todos deben ejecutarse de manera precisa. Uno de estos procesos es la segmentación de imágenes, que determina si las imágenes pueden proporcionar características destacadas útiles para alimentar un clasificador [20]. La Figura 2.1 ilustra de manera visual esta estructura operativa.

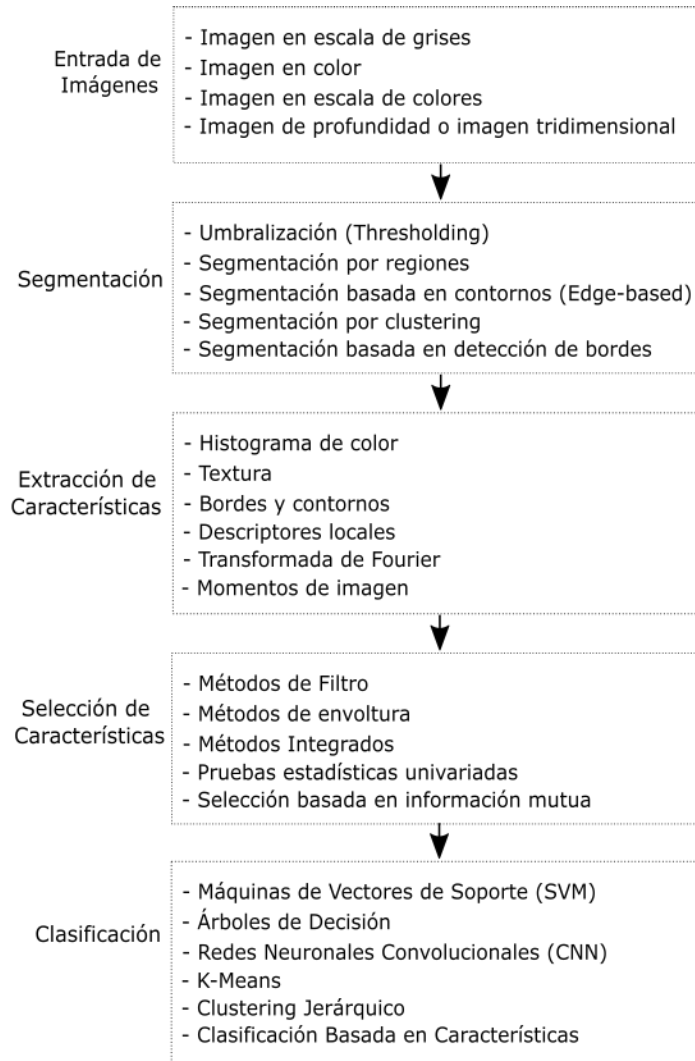


Figura 2.1: Procesos de un sistema de visión

La segmentación de imágenes desempeña un papel crucial en el procesamiento visual al ayudarnos a identificar las áreas de interés mientras ignoramos el fondo o las regiones que no son relevantes. En su esencia, la segmentación implica dividir una imagen en regiones distintas, donde las diferencias entre estas regiones se basan en las características individuales de cada píxel que compone la imagen. Estas características pueden incluir bordes, color, textura, variaciones de intensidad y forma, así como propiedades a nivel de objeto, como su forma, tamaño, orientación y ubicación con respecto a otros objetos [98] [52] [79] [75].

Las características basadas en los píxeles pueden extraerse mediante técnicas simples de procesamiento de imágenes. Por ejemplo, la textura cuantifica la apariencia física percibida de una superficie [40] [55] [25]. Las técnicas de segmentación textural generalmente se clasifican en técnicas de filtrado estadístico, espacial y espectral, donde intentan simular el sistema visual humano mediante el análisis de frecuencia espectral local [30]. Los bordes también representan una característica a nivel de píxel y pueden derivarse mediante el uso de un operador de gradiente en la imagen [78].

La segmentación considera algunos principios fundamentales de la imagen bajo examen:

- **Discontinuidad:** Se observa cuando las regiones que componen la imagen se distinguen y destacan entre sí, presentando bordes bien definidos.
- **Similitud:** Este principio aplica a imágenes en color o escala de grises, donde los píxeles tienen valores similares o coinciden con respecto a una medida establecida.
- **Conectividad:** Este principio establece que varios píxeles que pertenecen a un mismo objeto o región deben ser homogéneos y contiguos para formar parte de esa región.

La segmentación, una técnica fundamental en el procesamiento de imágenes, se diversifica en cuatro enfoques clave que abordan distintos niveles de granularidad y comprensión de la escena visual: Segmentación binaria, segmentación por instancias, segmentación semántica y segmentación panóptica.

Cada uno de estos enfoques tiene sus propias aplicaciones y desafíos específicos, y la elección entre ellos depende en gran medida de la naturaleza y los requisitos de la tarea de procesamiento de imágenes en cuestión.

Segmentación semántica: En esta subclasificación, la segmentación resultante agrupa objetos con propiedades similares en un solo color o grupo. Esto permite identificar y distinguir clases específicas de objetos en una imagen. Ver Figura 2.2 para una representación visual de este proceso.

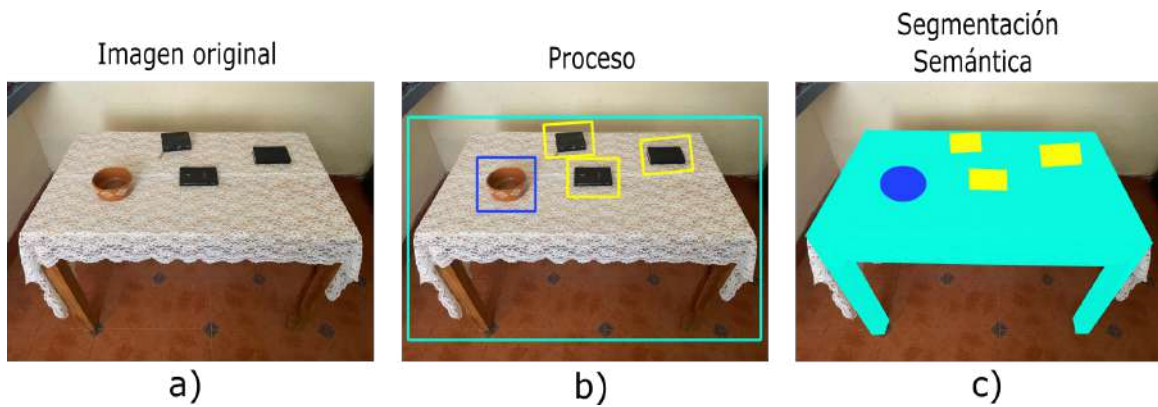


Figura 2.2: Segmentación Semántica. a) La imagen original contiene una mesa y ciertos objetos sobre de ella, tres libros y un tazón. b) En el proceso son detectados todos los objetos y se asocian aquellos que pertenecen a una misma clase. c) Se segmentan de un mismo color los objetos pertenecientes a una clase.

Segmentación por instancias: En este enfoque, se asignan múltiples colores o grupos para diferenciar cada objeto individual de interés en la imagen. Esto permite una identificación clara y distinta de cada instancia u objeto en particular. Ver Figura [2.3](#) para una representación visual.

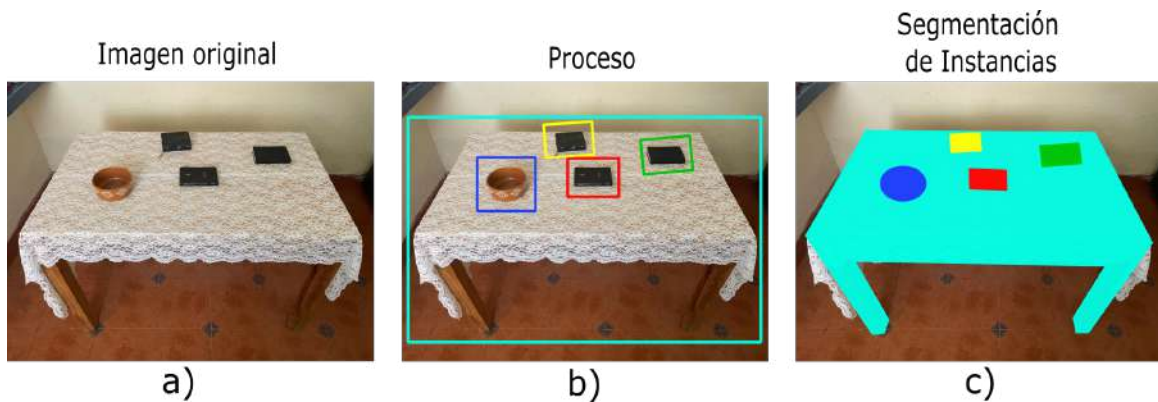


Figura 2.3: Segmentación por instancias. a) La imagen original contiene una mesa y ciertos objetos sobre de ella, tres libros y un tazón. b) En el proceso son detectados todos los objetos en la imagen. c) Se segmentan con distintos colores todos los objetos.

Segmentación binaria: En este tipo de segmentación, se dividen los elementos de la imagen en dos categorías distintas: los puntos de interés (objeto)

y los puntos que no son de interés (fondo). Como resultado, se obtiene una imagen con solo dos colores, representando claramente la distinción entre estas dos regiones. La Figura 2.4 muestra un ejemplo visual de este proceso.

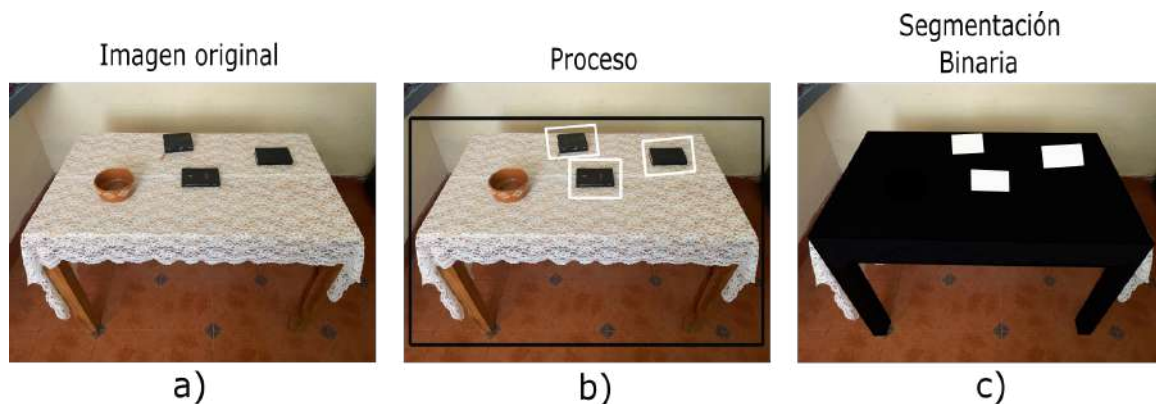


Figura 2.4: Segmentación binaria. a) La imagen original contiene una mesa y ciertos objetos sobre de ella, tres libros y un tazón. b) En el proceso son detectados los objetos de interes y al mismo tiempo el fondo. c) Se segmentan con un solo color todos los objetos de interes que se distinguen del fondo.

Segmentación panóptica: En este subclase, la segmentación panóptica es el resultado de fusionar las características y ventajas tanto de la segmentación por instancias como de la segmentación semántica.

Una segmentación adecuada se logra cuando las fronteras de los objetos se describen con claridad, permitiendo distinguirlos del fondo, y todo ello en un tiempo computacional razonable.

En términos generales, en las investigaciones más destacadas, los métodos de segmentación pueden dividirse en dos clases: aquellos basados en píxeles y aquellos basados en texels, siendo los primeros los más relevantes y los que sustentan este trabajo. En los últimos años, se han propuesto diversos métodos de segmentación, como se detalla en la Figura 2.5

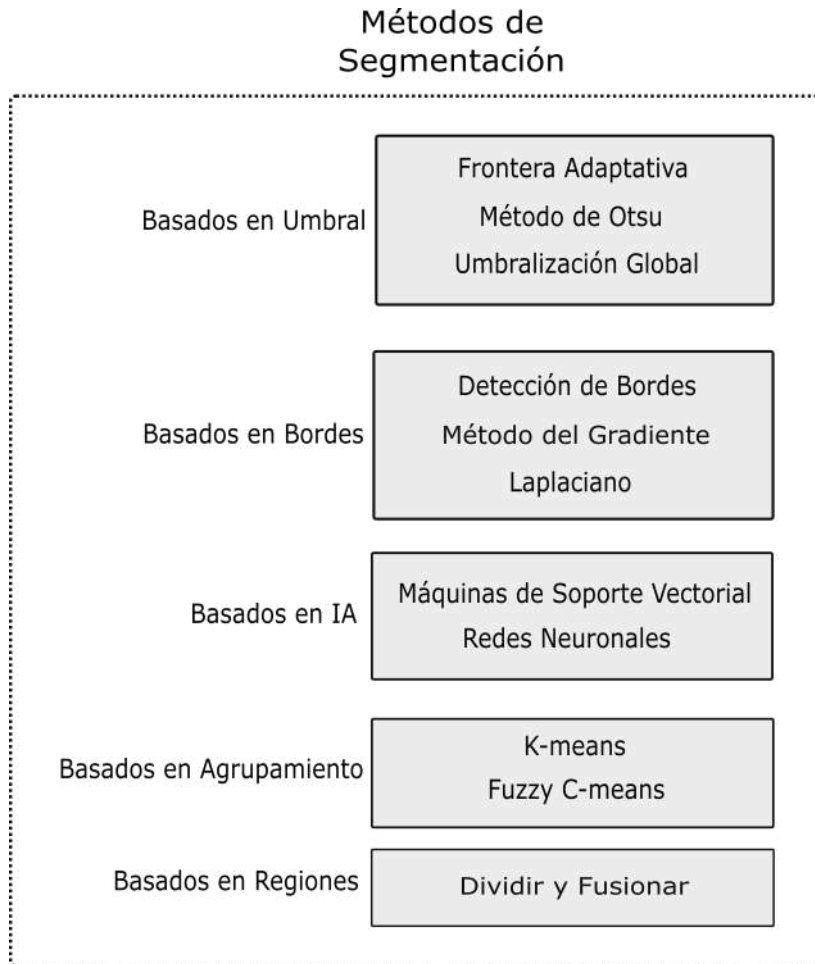


Figura 2.5: Clasificación de los diferentes tipos de Segmentación

2.2.1. Métodos de Segmentación Basados en Umbral

Entre los métodos de segmentación basados en umbrales, se destacan los más conocidos y efectivos, tales como la Segmentación por Umbral Global, la Frontera Adaptativa y el Método de Otsu.

En la Segmentación por Umbral Global, se emplea un bajo coste computacional. El proceso consiste en convertir una imagen de color en una imagen de niveles de grises y, posteriormente, en una imagen binaria. En esta imagen binaria, se asigna un valor de uno si el píxel supera el umbral establecido y un valor de cero si no lo hace [98] [20] [75] [95] [4] [27] [23]. El umbral se calcula de dos formas distintas:

$$th = \frac{\max(I) - \min(I)}{2} \quad (2.1)$$

En esta primera forma de encontrar el umbral (th) [2.1] se hace una diferencia del pixel con la intensidad mas alta y el pixel opuesto con la intensidad mas baja y finalmente se divide entre dos. Otra técnica consiste en la normalización del histograma de una imagen, que en términos más simples, implica calcular la suma de todas las probabilidades de la presencia de cada píxel.

En el método de frontera adaptativa, también se calcula un umbral, pero con la particularidad de que esta frontera varía, ya que el umbral depende de las propiedades de intensidad de una zona o vecindario específico de la imagen.

Finalizando esta subsección, tenemos el método de segmentación Otsu, desarrollado por Nobuyuki Otsu [68], quien emplea técnicas estadísticas para abordar problemas de segmentación. Este método se fundamenta en la varianza como métrica clave, la cual indica qué tan dispersos están los datos, en este caso, se refiere a los valores de gris.

La tarea principal de este método consiste en determinar el valor del umbral de manera que la varianza dentro de cada clase sea mínima, pero al mismo tiempo, la varianza entre clases distintas sea máxima. Aunque actualmente existen muchas técnicas diseñadas para encontrar un umbral adecuado, la mayoría de ellas no ofrecen resultados satisfactorios al procesar imágenes del mundo real, debido a problemas como histogramas planos, iluminación inadecuada o presencia de ruido. En este contexto, el método de Otsu se destaca entre estas técnicas de selección de umbral, ya que realiza una búsqueda exhaustiva y selecciona el umbral óptimo que maximiza la varianza entre clases.

La importancia del método de Otsu radica en su capacidad de automatización, lo que significa que no requiere la intervención de una persona ni información previa sobre las imágenes antes de su procesamiento. La clave para lograr una segmentación eficiente radica en la variación entre dos clases distintas de intensidades, y ajustar esta variación en cada iteración del proceso. Sin embargo, vale la pena señalar una desventaja: a medida que aumenta el número de clases en una imagen, también aumenta el tiempo necesario para seleccionar un umbral óptimo [20].

2.2.2. Métodos de Segmentación Basados en Bordes

En estos métodos, el objetivo principal es identificar las fronteras o bordes de los objetos en una imagen. Estas fronteras se definen como las transiciones entre dos regiones y suelen manifestarse como cambios abruptos de intensidad o color en la imagen. Una de las ventajas más significativas de este enfoque es la capacidad de reducir la cantidad de datos a procesar sin perder las características estructurales de la imagen [88].

Los métodos que se describirán a continuación incluyen detectores de bordes (operadores), el método del gradiente y el método de Laplaciano [89].

Los detectores de bordes son fundamentales en el procesamiento de imágenes y logran resultados satisfactorios a través de una serie de pasos clave:

- Detectar bordes: Este primer paso implica identificar las transiciones significativas en la intensidad de los píxeles de una imagen.
- Eliminar bordes irrelevantes: Una vez que se han detectado los bordes, es importante eliminar los detalles no relevantes que puedan haber sido detectados accidentalmente o debido al ruido en la imagen.
- Conectar o agrupar: Este último paso implica la unión de segmentos de bordes cercanos y relacionados para formar contornos continuos y más significativos.

Los detectores de bordes se pueden clasificar en los siguientes operadores:

- Operador de Sobel.
- Operador de Prewitt.
- Operador de Canny.

El operador de Sobel se clasifica como una técnica de primer orden y se basa en el cálculo de la primera derivada de la imagen para explorar y localizar máximos y mínimos, lo que permite detectar los bordes de la imagen. Específicamente, el operador de Sobel se utiliza para aproximar la magnitud total del gradiente de la intensidad de cada píxel mediante matrices de convolución o un par de filtros de 3x3. Estos filtros proporcionan una estimación de la dirección del gradiente en diagonal. Una ventaja destacada de este operador es su menor sensibilidad al ruido, atribuible al uso de núcleos de convolución más grandes que permiten suavizar la imagen en mayor medida que otras técnicas [9].

El operador de Prewitt funciona de manera similar al operador de Sobel, pero con la distinción de que emplea filtros más pequeños. Utiliza un kernel diferente que proporciona estimaciones de direcciones de gradientes, ya sea verticales u horizontales. Como el operador de Sobel, el operador de Prewitt es una técnica de primer orden que se basa en el cálculo de la primera derivada de la imagen para detectar bordes. Sin embargo, debido al uso de filtros más pequeños, el operador de Prewitt puede ser más rápido computacionalmente en comparación con el operador de Sobel. A pesar de esta diferencia en el tamaño del kernel, ambos operadores comparten la ventaja de ser menos sensibles al ruido en comparación con otras técnicas de detección de bordes, lo que los hace útiles en entornos donde la calidad de la imagen puede ser comprometida por factores como el ruido o la iluminación inconsistente.

El operador de Canny, concebido por John F. Canny, es un algoritmo multietapa utilizado para detectar bordes en imágenes. Este operador se compone de cinco etapas secuenciales, cada una utilizando los resultados de la etapa anterior. Comienza con la conversión de la imagen a escala de grises, seguida de la reducción de ruido, el cálculo del gradiente de intensidad, la supresión de

no máximos, la aplicación de un doble umbral y concluye con el proceso de umbralización por histéresis [11].

El método del gradiente, similar a los operadores mencionados previamente, se basa en el cálculo de la primera derivada para detectar las variaciones locales en la imagen. Sin embargo, a pesar de su utilidad, este método a menudo produce resultados subóptimos al detectar elementos no deseados o pasar por alto características importantes. Para abordar esta limitación, se pueden implementar estrategias de optimización adicionales, aunque esto conlleva un aumento en el costo computacional del proceso.

Para completar esta subsección, se destaca el método Laplaciano, que se distingue de los enfoques anteriores por utilizar una segunda derivada para mejorar la localización de las fronteras. El cálculo del Laplaciano de una imagen bidimensional $I(x, y)$ se puede realizar mediante la ecuación 2.2

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (2.2)$$

El Laplaciano actúa como un operador de filtrado que realza las discontinuidades en la imagen, lo que permite identificar con mayor precisión los bordes. Sin embargo, debido a su susceptibilidad al ruido, el Laplaciano puede generar respuestas no deseadas en áreas de alta frecuencia.

Para mitigar este efecto y obtener resultados más robustos, es común aplicar técnicas de suavizado antes de calcular el Laplaciano. Estas técnicas incluyen el filtrado gaussiano, que ayuda a reducir el ruido mientras preserva las características importantes de la imagen.

A pesar de sus ventajas, el método Laplaciano puede ser computacionalmente costoso debido a la naturaleza de la segunda derivada y a la necesidad de realizar operaciones de convolución en toda la imagen.

A continuación se presentan algunos ejemplos de filtros de Laplaciano comúnmente utilizados en el procesamiento digital de imágenes:

1. Filtro de Laplaciano estándar: Este filtro utiliza una máscara Laplaciana simple para calcular la segunda derivada de la imagen. Por ejemplo:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

2. Filtro de Laplaciano Gaussiano: Este filtro combina el filtro de Laplaciano con un suavizado Gaussiano para reducir el ruido antes de aplicar la detección de bordes. Por ejemplo:

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

3. Filtro de Laplaciano del Gaussiano (LoG): Este filtro primero aplica un filtro Gaussiano para suavizar la imagen y luego aplica el filtro Laplaciano. Por ejemplo, la versión discreta del filtro LoG es:

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Como el Laplaciano es un operador de derivación, el coeficiente que se encuentra localizado en el píxel central debe ser positivo, mientras que los coeficientes faltantes deben de ser negativos o ceros.

En esta situación, la expresión para determinar el Laplaciano es la formula

2.3

$$4z_5 - (z_2 + z_4 + z_6 + z_8) \quad (2.3)$$

2.2.3. Métodos de Segmentación Basados en Agrupamiento

Los métodos aquí descritos se basan en el principio de similitud y tienen como objetivo encontrar regiones o conjuntos de subregiones con características comunes. Estos métodos tienen en cuenta las relaciones espaciales entre los vecinos de cada píxel y ciertos criterios de similitud. A continuación, se describen algunos de estos criterios [20]:

- Se establece un rango definido para la diferencia de intensidad absoluta entre un píxel aspirante y el píxel inicial.
- Se define un rango específico para la diferencia de intensidad absoluta entre un píxel aspirante y la intensidad media de la región circundante.
- Se aplica un criterio básico de suavidad o rugosidad, donde se fija un umbral para la diferencia entre la desviación estándar de la intensidad en una vecindad local particular del píxel aspirante y la desviación estándar de la vecindad local del píxel inicial (puede o no exceder el umbral).

Los métodos analizados en esta tesis son el K-means y el Fuzzy C-means.

K-means es un algoritmo versátil que se utiliza tanto para clasificación como para segmentación. En el contexto de la segmentación de imágenes, K-means agrupa los píxeles en k conjuntos basados en sus características, lo que efectivamente segmenta la imagen en regiones con propiedades similares y específicas, que luego se pueden interpretar como objetos en la imagen procesada [63].

El proceso de agrupamiento implica minimizar la suma de distancias entre cada píxel y el centroide de su grupo, también conocido como cluster [73]. La

distancia utilizada suele ser la distancia euclidiana o la distancia cuadrática. El proceso de K-means se resume en el algoritmo mostrado en el Cuadro 2.1

Entrada:	Número de grupos o k centroides
Salida:	Segmentación de diferentes objetos
1:	Se eligen k puntos como centroides.
2:	WHILE no exista cambios en los centroides Do
3:	Se calcula la distancia de todos los píxeles con respecto a los centroides.
4:	Se calcula la media por cada punto.
5:	se actualiza la posición de cada centroide.
6:	END

Tabla 2.1: Algoritmo K-means

El algoritmo Fuzzy C-means es una variante del método de segmentación por agrupamiento, similar al algoritmo K-means, pero con la mejora de introducir la lógica difusa para manejar situaciones más complejas. A diferencia de K-means, donde un píxel se asigna estrictamente a un solo grupo, en Fuzzy C-means, un píxel puede tener una pertenencia parcial a múltiples grupos.

En el proceso de segmentación por K-means, puede ocurrir que un píxel esté cerca de los centroides de múltiples grupos, lo que dificulta su clasificación en un grupo específico. Fuzzy C-means resuelve este problema representando la similitud entre un píxel y un grupo mediante una función de pertenencia, que toma valores en el rango de $[0,1]$. Valores cercanos a uno indican una alta similitud, mientras que valores cercanos a cero indican una baja similitud. Esta flexibilidad permite una asignación más suave y probabilística de los píxeles a los grupos, lo que resulta en una segmentación más precisa y adaptable a diferentes contextos.

2.2.4. Métodos de Segmentación Basados en Regiones

Los métodos basados en regiones generalmente inician etiquetando un píxel y luego agregan píxeles adicionales según un criterio de similitud. Este proceso se repite iterativamente hasta que se completa una región. Sin embargo, estos métodos pueden presentar desventajas, especialmente en presencia de ruido en la imagen. El rendimiento puede ser deficiente y el costo computacional alto debido a la necesidad de analizar múltiples regiones y evaluar la similitud entre píxeles adyacentes [29] [94].

El método de dividir y fusionar describe su proceso en su nombre. En primer lugar, la imagen de entrada se divide en una estructura de datos de un árbol cuádruple. La imagen se divide inicialmente en cuatro cuadrantes, y este proceso se repite recursivamente, dividiendo cada cuadrante en cuatro subregiones más, hasta que se alcanza el objetivo de agrupar píxeles del mismo color en una única región.

El paso final es la fusión, donde se unen todas las regiones adyacentes que cumplen con un criterio de similitud establecido [67] [87].

2.2.5. Métodos de Segmentación Basados en IA

Los métodos basados en Inteligencia Artificial (IA) son procesos que buscan emular las características, propiedades, cualidades o capacidades del intelecto humano para maximizar el rendimiento y obtener resultados óptimos al aplicarlos a problemas específicos. Entre estos métodos, las redes neuronales artificiales destacan por su capacidad para imitar el comportamiento de las redes neuronales biológicas, permitiendo que las computadoras procesen datos de manera inteligente.

El objetivo de las redes neuronales es resolver problemas mediante el flujo de información, ajustando los valores en las conexiones entre las neuronas para formar capas, de forma similar al cerebro humano. Estas redes generalmente se componen de capas de nodos, que incluyen una capa de entrada, una o varias capas ocultas y una capa de salida. Cada neurona está conectada a otra con un peso o umbral asociado; si la salida de una neurona supera el umbral, se activa, de lo contrario, no se envían datos a la siguiente capa. Una ventaja clave de este modelo es su alto grado de paralelismo, que reduce los tiempos computacionales y su capacidad para aprender y corregir errores mediante un proceso llamado propagación hacia atrás [6].

Las Máquinas de Soporte Vectorial (SVM), conocidas por sus siglas en inglés Support Vector Machines, son un poderoso método de aprendizaje supervisado utilizado en el campo del procesamiento de imágenes para la segmentación y clasificación de elementos. Aunque su aplicación es versátil y puede emplearse en problemas de clasificación multiclase, en muchos escenarios prácticos se utilizan como clasificadores binarios, especialmente para segmentar imágenes en dos clases distintas, como los píxeles de fondo y los píxeles de objetos de interés [16] [58].

La principal característica distintiva de las SVM radica en su capacidad para encontrar un hiperplano óptimo que permita separar de manera óptima las dos clases. Este hiperplano se sitúa en un espacio de alta dimensionalidad definido por las características de las imágenes, y su objetivo es maximizar el margen de separación entre las clases, es decir, la distancia entre el hiperplano y los puntos de datos más cercanos de cada clase. Esto garantiza una segmentación precisa y robusta de la imagen [59] [19] [15].

Además de la capacidad para trabajar con datos de alta dimensionalidad, las SVM tienen la ventaja de ser eficientes en la gestión de conjuntos de datos grandes y complejos. Esto se debe a que, en lugar de depender exclusivamente de los datos de entrenamiento, las SVM se basan en un subconjunto de puntos de datos llamados vectores de soporte, que son los que definen el hiperplano óptimo. Esto reduce significativamente la complejidad computacional y evita el sobreajuste a los datos de entrenamiento [46].

Otra ventaja importante de las SVM es su capacidad para manejar datos no lineales mediante el uso de funciones de kernel, que permiten mapear los datos

de entrada a un espacio de características de mayor dimensionalidad donde las clases son linealmente separables. Esto amplía enormemente la aplicabilidad de las SVM a una amplia gama de problemas de segmentación de imágenes, incluso cuando las relaciones entre las características no son lineales [14].

2.3. Algoritmos Evolutivos

En términos generales, cualquier tarea por realizar puede ser concebida como un problema que requiere una solución utilizando las herramientas disponibles. Bajo esta perspectiva, también puede entenderse como un proceso de exploración a través de un espacio de posibles soluciones.

Los problemas de optimización han sido objeto de un profundo estudio debido a su importancia e impacto en la práctica. Algunos, como la optimización lineal, han sido completamente resueltos mediante métodos matemáticos. Sin embargo, el desafío común de la optimización global no lineal sigue sin resolver, ya que no existe ninguna técnica determinista que garantice siempre la convergencia a una solución óptima de una función objetivo arbitraria con restricciones también arbitrarias [74]. Ante este desafío, los métodos heurísticos han experimentado un gran avance, destacando entre ellos los algoritmos evolutivos.

Los algoritmos evolutivos son estrategias de búsqueda y optimización. En un vasto espacio de posibles soluciones, estos algoritmos encuentran la mejor opción en el menor tiempo posible, lo que justifica su denominación como tácticas de búsqueda y optimización. El término 'evolutivos' hace referencia a la teoría darwiniana, que postula que las especies evolucionan a lo largo del tiempo mediante la selección natural, dando lugar a nuevas especies. La evolución no se limita únicamente al ámbito biológico; de manera similar, está presente en sistemas económicos, artísticos y sociales, y puede ser implementada y observada en sistemas computacionales [72] [84] [57].

La premisa fundamental de los algoritmos evolutivos consiste en mantener un conjunto de individuos, comúnmente denominado "generación", que representen un conjunto de posibles soluciones para el problema planteado. Durante cada iteración, estos individuos se someten a un proceso de cruce y mutación, compitiendo entre sí mediante una presión selectiva estocástica para evitar un crecimiento exponencial del algoritmo.

El cruce implica la combinación de los padres (dos individuos o dos soluciones) para generar un nuevo candidato, mientras que la mutación afecta directamente a un único individuo, produciendo otro candidato. Estas operaciones dan lugar a una nueva generación de candidatos con diferentes características, listos para ser evaluados. Este ciclo de operaciones se repite a lo largo de varias iteraciones hasta que el algoritmo converge hacia un candidato lo suficientemente óptimo.

En nuestro mundo, los recursos son finitos, lo que implica que solo aquellos capaces de adquirir y utilizar eficientemente estos recursos podrán prosperar. Este proceso se asemeja a la selección natural, donde solo los individuos mejor adaptados a su entorno logran sobrevivir y reproducirse con el tiempo. De

manera similar, en el ámbito de las soluciones, aquellas que mejor se adaptan y funcionan de manera eficiente tienen más probabilidades de perdurar. Es una evolución hacia soluciones cada vez más óptimas [1].

En términos simples, los algoritmos evolutivos buscan determinar una secuencia de valores, desde a_1 hasta a_n , donde cada uno de estos valores agrupados conforma un cromosoma artificial, donde los genes a_i representan parámetros del problema y estos son análogos a las bases del ADN biológico. La expresión matemática (2.4) formaliza esta representación:

$$C = (a_1, a_2, a_3, \dots, a_n) \quad (2.4)$$

El objetivo es minimizar una función determinada $f(C)$ o, de manera equivalente, $f(a_1, a_2, a_3, \dots, a_n)$. Durante este proceso, se aplican operadores genéticos como mutación (perturbaciones locales) y cruza (combinación de soluciones) y se ejerce una presión selectiva (preservando individuos con menor $f(C)$) para preservar las características que hacen a un individuo más apto para sobrevivir y resolver el problema planteado. Como consecuencia, los individuos menos aptos son eliminados progresivamente y se garantiza que las soluciones evolucionen hacia regiones óptimas del espacio de búsqueda.

En términos generales, los motores de la evolución pueden clasificarse en tres procesos fundamentales: la reproducción (que genera un conjunto inicial de posibles soluciones de forma aleatoria), las mutaciones (que introducen variabilidad en el conjunto), y la presión selectiva (que aumenta la complejidad del sistema), estos procesos son los que determinan que la evolución pueda considerarse un algoritmo que emerge cuando existe reproducción y suficiente complejidad.

En la Figura 2.6 se presenta una clasificación de métodos de búsqueda y optimización, destacando los métodos que son objeto de estudio en este trabajo.

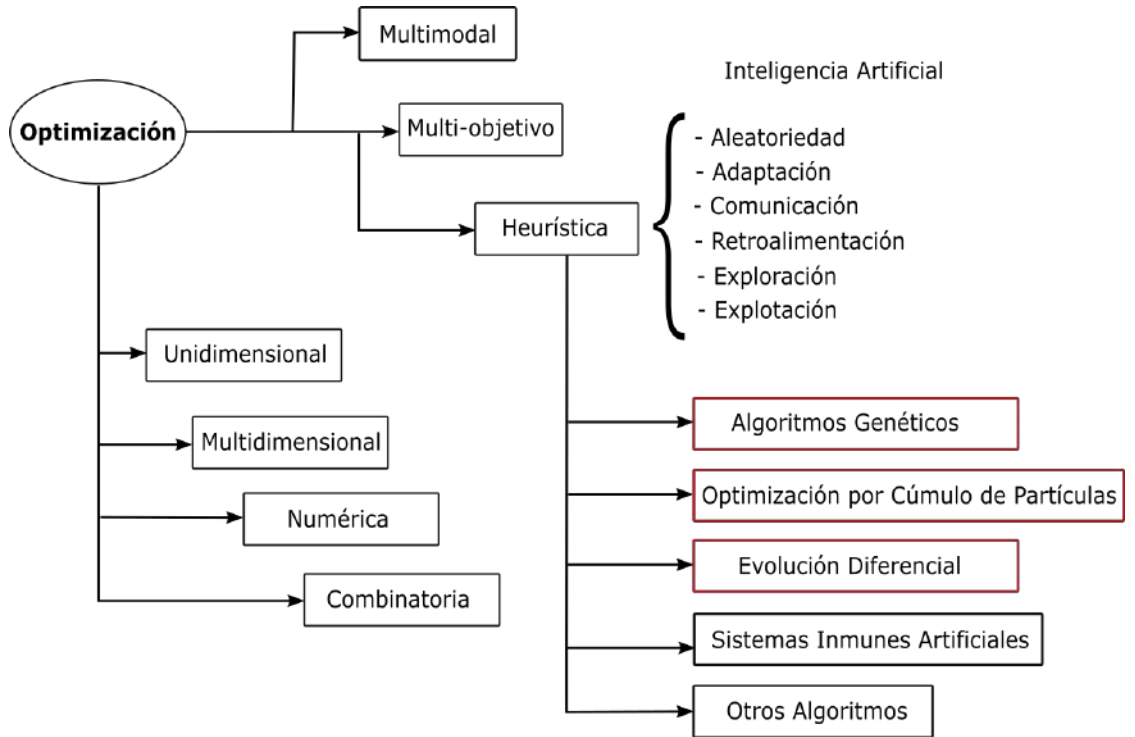


Figura 2.6: Métodos de Optimización

Los algoritmos evolutivos forman parte de las técnicas heurísticas de optimización, las cuales buscan encontrar soluciones satisfactorias para un problema basándose en la imitación del comportamiento de diversas poblaciones en el mundo animal.

Estos algoritmos se caracterizan por explorar iterativamente un espacio de soluciones, con la esperanza de que cada iteración (también conocida como generación) produzca un resultado mejor que la iteración anterior. Si bien existen numerosas variantes de estos algoritmos, así como combinaciones de estas variantes documentadas en la literatura, todos comparten la simulación de la evolución de individuos como característica fundamental. Se diferencian entre sí por la forma en que implementan sus componentes.

En el marco de esta tesis, se han aplicado tres algoritmos evolutivos: los algoritmos genéticos, la evolución diferencial y la optimización por cúmulo de partículas (PSO).

Para terminar esta subsección se agregan las principales ventajas que presenta la implementación de algoritmos evolutivos en problemas de optimización [37](#).

2.3.1. Ventajas de los Algoritmos Evolutivos

Las ventajas de los algoritmos evolutivos son abundantes y fundamentales para su aplicación exitosa en una amplia gama de problemas de optimización y búsqueda. Algunas de estas ventajas incluyen:

- Exploración global: Los algoritmos evolutivos operan sobre una población completa de soluciones, lo que les permite explorar de manera exhaustiva el espacio de búsqueda y reducir significativamente el riesgo de quedar atrapados en óptimos locales. Esta capacidad de búsqueda global es esencial para encontrar soluciones óptimas en problemas complejos con múltiples óptimos locales.
- No requiere información previa del problema: A diferencia de algunos métodos de optimización que dependen de información específica sobre el problema, los algoritmos evolutivos pueden aplicarse sin necesidad de conocimiento previo detallado del problema. Esto los hace extremadamente versátiles y adecuados para una amplia variedad de aplicaciones y escenarios.
- Flexibilidad en la implementación: Los algoritmos evolutivos pueden utilizarse de forma independiente o en combinación con otros métodos de búsqueda para mejorar los resultados. Esta capacidad de integración les permite adaptarse a las necesidades específicas de cada problema y aprovechar sinergias entre diferentes enfoques de optimización.
- Paralelización sencilla: Los algoritmos evolutivos se prestan fácilmente a la paralelización, lo que significa que pueden ejecutarse en múltiples procesadores o núcleos de manera simultánea para acelerar el proceso de búsqueda y obtener resultados más rápidos en problemas de gran escala.
- Implementación y manejo sencillos: Aunque son poderosos en términos de capacidad de búsqueda, los algoritmos evolutivos suelen ser relativamente sencillos de implementar y manejar. Su estructura y operación intuitivas los hacen accesibles incluso para aquellos sin experiencia previa en optimización evolutiva.

2.4. Algoritmos Genéticos

Los algoritmos genéticos tienen como base los principios de la evolución y el proceso genético de los organismos vivos, donde la supervivencia del individuo más fuerte es fundamental. Los individuos más aptos se adaptan al entorno en el que se encuentran para asegurar su supervivencia [69] [20] [17] [47].

A diferencia de otras técnicas de búsqueda convencionales, los algoritmos genéticos comienzan con una población inicial de posibles soluciones generada aleatoriamente, donde cada individuo se representa como un cromosoma. A lo largo de las generaciones, esta población evoluciona y se fortalece, sometiéndose

a operadores como la mutación y el intercambio de material genético entre los descendientes [64] [18].

La evaluación de cada cromosoma se realiza mediante una medida de aptitud calculada mediante una función específica de evaluación para cada problema. Por lo tanto, el valor de aptitud indica la calidad de la solución, con valores más altos para problemas de maximización y valores más bajos para problemas de minimización. El cruzamiento genético puede llevarse a cabo de diversas maneras, como el cruzamiento a un punto o a dos puntos, donde se intercambia información genética entre los padres. Además, se realiza una mutación en un elemento de la cadena cromosómica.

Estos algoritmos se componen típicamente de cinco etapas principales: la inicialización de la población, la evaluación de la aptitud de los individuos, la selección de individuos para la reproducción, el cruzamiento genético y la mutación. Este ciclo se repite iterativamente hasta alcanzar un criterio de parada [93]. A continuación, se muestra un ejemplo de un algoritmo genético básico en el Cuadro 2.2.

Entrada:	Variables, espacio de búsqueda, tamaño de cromosoma, etc
Salida:	Optima solución
1:	Generar aleatoriamente una población inicial
2:	Evaluar la aptitud de la población creada
3:	WHILE condición de paro no se cumple Do
4:	Selección de los padres para la reproducción
5:	Cruza de los padres seleccionados
6:	Mutación de individuos
7:	Obtener la nueva generación END

Tabla 2.2: Algoritmo Genético Básico.

Al igual que cada individuo tiene una composición genética única, compuesta por varios elementos, los individuos dentro del algoritmo genético también contienen características distintivas. En la Figura 2.7 se presenta una representación gráfica de estos elementos.

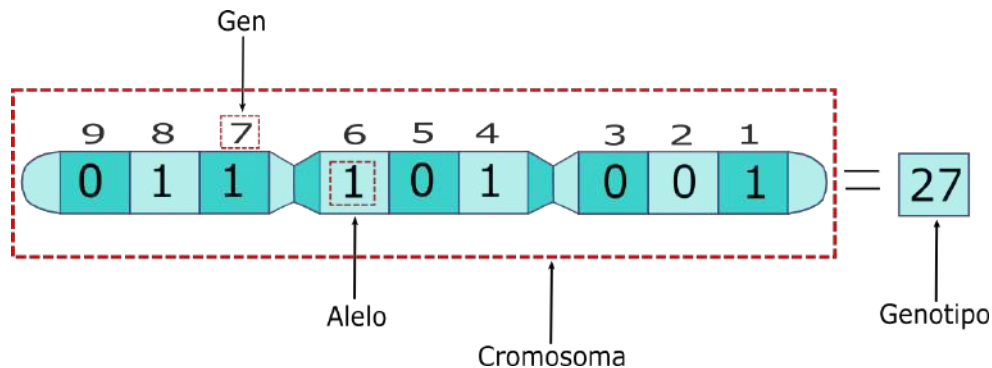


Figura 2.7: Representación gráfica de las expresiones que se usa en la genética

Cada individuo en el algoritmo está representado por una cadena binaria conocida como cromosoma. El término 'gen' se utiliza para referirse a la codificación de una cualidad específica del individuo, donde cada gen puede tomar varios valores representados por los alelos. El conjunto de todos estos genes constituye una cadena binaria que, una vez decodificada, proporciona el genotipo del individuo.

El desarrollo de un algoritmo genético básico implica tomar las decisiones más simples posibles. Comienza con la generación aleatoria de individuos, seguida de una selección en la que solo los más adaptados sobreviven. Los individuos para el cruzamiento se eligen de manera secuencial, ordenados por su eficiencia. La mutación se realiza de forma aleatoria y se calcula la aptitud de cada individuo mediante una función de adaptación. Estos pasos se repiten en un ciclo hasta que se cumplan los criterios de parada.

2.4.1. Población y Selección de Individuos

Como se mencionó anteriormente, la población se va creando de manera aleatoria, con cada individuo siendo generado con la misma probabilidad. Esto implica que los cromosomas se van formando a partir de sus genes, donde cada gen es asignado mediante una función que produce un valor binario (0 o 1) con igual probabilidad (50 % para cada opción) [3]. Es crucial que el algoritmo tenga una amplia variedad de cromosomas para poder explorar todas las regiones del espacio de búsqueda.

Existen diversas técnicas de selección en algoritmos genéticos, que pueden clasificarse en tres grupos principales:

- Selección proporcional: También conocida como selección basada en ruleta, este método asigna una probabilidad de selección a cada individuo de la población proporcional a su aptitud relativa.
- Selección mediante torneo: En este enfoque, se seleccionan aleatoriamente un número fijo de individuos de la población (llamado tamaño del torneo)

y se elige al mejor de ellos como padre para la reproducción. Este proceso se repite varias veces para formar la población de descendencia. La ventaja de esta técnica es que no requiere conocer la aptitud relativa de todos los individuos en la población, lo que la hace eficiente computacionalmente.

- Selección de estado uniforme: En este método, todos los individuos tienen la misma probabilidad de ser seleccionados para la reproducción, independientemente de su aptitud. Esto asegura una exploración más diversa del espacio de búsqueda y puede ser útil cuando se busca evitar la convergencia prematura hacia óptimos locales.

En este trabajo, se analizará exclusivamente una técnica de selección conocida como **Ruleta**, la cual pertenece al grupo de "selección proporcional". Esta técnica se destaca y se utiliza en algoritmos genéticos debido a que la probabilidad de que un individuo sea seleccionado es directamente proporcional a su función de aptitud. Es decir, el proceso de selección favorece la reproducción de aquellos individuos que mejor se adaptan al entorno.

La técnica de la ruleta es ampliamente empleada desde los primeros días de los algoritmos genéticos, principalmente debido a su simplicidad de implementación. Sin embargo, presenta un inconveniente notable: existe la posibilidad de que los individuos con menor aptitud sean seleccionados una o más veces durante el proceso. El algoritmo de la ruleta se detalla en el Cuadro 2.3

Entrada:	Valor de T , Tamaño de la población
Salida:	Individuos Seleccionados
1:	Calcular la suma de valores esperados T
2:	Repetir N veces (N es el tamaño de la población)
3:	Generar un número aleatorio r entre 0,0 y T
4:	Ciclar a través de los individuos de la población sumando los valores esperados hasta que la suma sea mayor o igual a r .
5:	El individuo que exceda este resultado será seleccionado.

Tabla 2.3: Algoritmo de la Técnica de Selección: Ruleta.

2.4.2. Cruza

Durante este subproceso, se lleva a cabo la fusión de la información genética de dos cromosomas. El cruzamiento implica la participación de los padres con el fin de transmitir características importantes a su descendencia.

Una de las formas más simples de llevar a cabo este subproceso es mediante el intercambio de información en un solo punto. Esto implica seleccionar un punto de corte entre dos genes de la cadena cromosómica de ambos padres y cambiar las secciones divididas por esta posición entre sí. La Figura 2.8 muestra una representación gráfica de este proceso

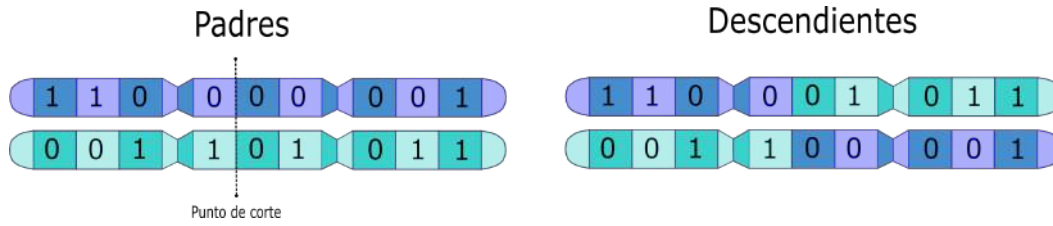


Figura 2.8: Representación gráfica de cruce a un punto

2.4.3. Mutación

El operador de mutación es un subproceso en el cual se altera el material genético. En este caso, uno de los genes en la cadena cromosómica sufre una variación de manera aleatoria. En la genética humana, este fenómeno puede ocurrir debido a errores en la reproducción o deformaciones genéticas, aunque su probabilidad de ocurrencia es baja. En los algoritmos genéticos, la mutación tiene una probabilidad mayor de ocurrir y no debe ser considerada perjudicial, sino más bien beneficiosa, ya que ayuda a evitar la convergencia hacia mínimos locales [50].

La forma más simple de este proceso implica modificar el valor de una posición en la cadena de manera aleatoria. Si el valor original es uno, la mutación cambiará el valor a cero; si el valor original es cero, la mutación lo cambiará a uno. Esto se ilustra en la Figura 2.9.

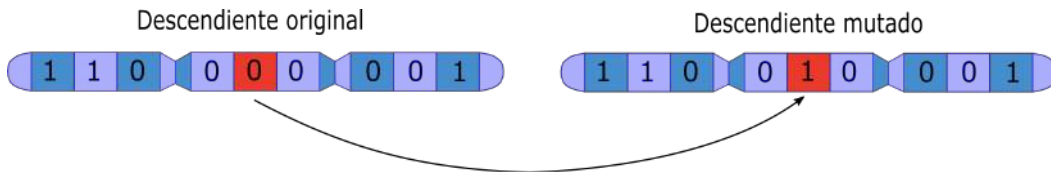


Figura 2.9: Operador de Mutación

2.4.4. Condición de Paro

Hay varias condiciones de terminación que pueden detener el funcionamiento del algoritmo y llevarlo a una solución. Las más comunes incluyen alcanzar el número máximo de iteraciones establecido y detectar que la mayoría de los individuos en la población convergen hacia una solución similar, lo que indica una pérdida de diversidad. En este punto, continuar con la evolución puede ser redundante y, por lo tanto, innecesario.

2.5. Optimización por Cúmulos de Partículas

La Optimización por Cúmulo de Partículas, también conocida como PSO (por sus siglas en inglés, Particle Swarm Optimization), es un algoritmo metaheurístico inspirado en la naturaleza, específicamente en el comportamiento de enjambres de peces y aves. Su objetivo es encontrar de manera eficiente las soluciones globales para problemas de optimización, ya sea maximizando o minimizando una función dentro de una región específica. Este método se ha aplicado con éxito en diversos desafíos en áreas como aprendizaje automático, procesamiento de mallas e investigaciones operativas [38].

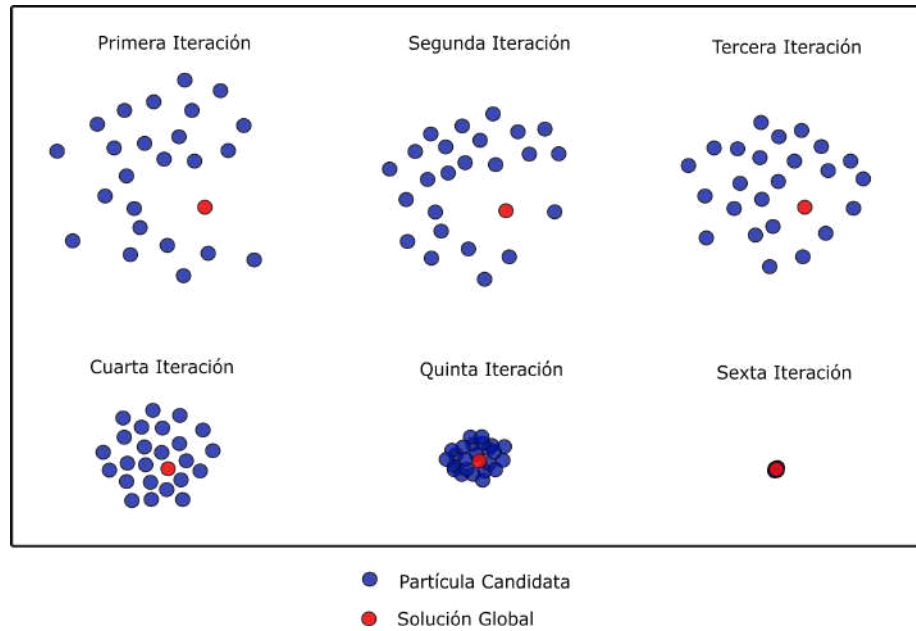


Figura 2.10: Proceso General del Algoritmo PSO

Una de las principales dificultades en la minimización de una función radica en la complejidad de los problemas prácticos, que a menudo implican cálculos de derivadas complicados y pueden converger a mínimos locales, lo que a veces resulta imposible de verificar. Por esta razón, los algoritmos metaheurísticos son herramientas de apoyo valiosas. Estos algoritmos están diseñados para funcionar mediante el método de prueba y error, destacando por su capacidad de cooperación, donde cada partícula en el sistema se comunica y comparte información sobre el área de trabajo.

Esta estructura de cooperación es una ventaja significativa, ya que permite que un conjunto grande de partículas aprenda unas de otras y converjan hacia la solución óptima en cada iteración. En este contexto, cada partícula en el

algoritmo representa una solución factible, con las soluciones menos eficientes descartadas y la mejor solución conservada.

Una representación visual de este proceso se muestra en la Figura [2.10](#)

El algoritmo comienza con un conjunto de puntos seleccionados al azar en la región donde se busca el mínimo de una función. Este conjunto de puntos se denomina enjambre y a cada punto se le llama partícula. En cada iteración del algoritmo, estas partículas se mueven de manera semialeatoria, siendo atraídas hacia la partícula que tiene el valor mínimo global en la iteración actual. A medida que avanza el proceso, se llega a un punto donde la función no puede minimizarse más o se alcanza el número máximo de iteraciones permitidas. En ese momento, el algoritmo finaliza y se selecciona como solución la partícula que tiene el valor mínimo global en la última iteración.

Matemáticamente, se calculan las posiciones de cada partícula en el enjambre y se guarda el valor de la partícula mejor posicionada. En cada iteración, se calculan las nuevas posiciones y velocidades de las partículas, comparando la mejor posición de la iteración actual con la de la anterior y conservando la mejor de ambas.

La fórmula para calcular la nueva posición de la partícula es la siguiente:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2.5)$$

donde:

- x_i^{t+1} Es la nueva posición de la partícula i
- x_i^t Es la posición de la partícula i en la iteración t
- v_i^{t+1} Es la nueva velocidad de la partícula i

El valor de la nueva velocidad no es un dato proporcionado, por lo tanto, se calcula con la siguiente ecuación:

$$v_i^{t+1} = v_i^t + \alpha \varepsilon_1 \cdot (x_i^* - x_i^t) + \beta \varepsilon_2 \cdot (g^* - x_i^t) \quad (2.6)$$

donde:

- v_i^t Es la velocidad de la partícula i en la iteración t
- α/β Son constantes de aceleración
- $\varepsilon_1/\varepsilon_2$ Son vectores aleatorios cuyas componentes están uniformemente distribuidas entre $[0, 1]$
- x_i^* Es la mejor posición de la partícula i en la iteración t
- g^* Es la mejor posición global en la iteración t

El algoritmo de PSO se puede observar en la Tabla [2.4](#)

En términos simples, el algoritmo de Optimización por Cúmulo de Partículas considera a cada individuo como una partícula que se desplaza por un espacio de

Entrada:	Función objetivo, constantes de aceleración.
Salida:	Mejor posición de la partícula y mejor posición global
1:	Inicialización del enjambre
2:	Se definen de manera aleatoria las posiciones de las partículas
3:	Se definen las velocidades iniciales de cada partícula
4:	WHILE condición de paro no se cumple Do
5:	Se encuentran las nuevas posiciones de las partículas
6:	Se calculan las nuevas velocidades de las partículas
7:	Se guarda la mejor posición de la iteración
8:	Se compara la mejor posición de la iteración con respecto a la mejor posición global
9:	END

Tabla 2.4: Algoritmo Optimización por Cúmulos de Partículas.

soluciones de un problema específico. Cada partícula tiene una posición x^ti que representa una posible solución al problema, y una velocidad v^ti que determina su movimiento en el espacio. La calidad de la posición de cada individuo se denomina aptitud, y para explorar ampliamente el espacio de búsqueda, las posiciones y velocidades iniciales de las partículas se generan aleatoriamente.

A medida que avanzan las iteraciones, las velocidades y posiciones de las partículas cambian en función de la interacción social entre ellas. Cada partícula tiende a emular el éxito de otras partículas que han encontrado una mejor solución. La actualización de la posición de una partícula depende de dos factores principales: la mejor posición visitada por la propia partícula x_i^* y la mejor posición visitada por cualquier otra partícula en el enjambre g^* . Con el tiempo, el algoritmo concentra las partículas en la zona donde se encuentra la mejor solución global g^* .

Al finalizar, este algoritmo de optimización proporciona la mejor solución encontrada por alguna de las partículas del enjambre [48].

Capítulo 3

Segmentación Basada en Algoritmos Genéticos

Este capítulo detalla los procedimientos empleados en los métodos propuestos de algoritmos genéticos, incluyendo sus características y estructuras. El objetivo primordial de esta investigación consiste en optimizar la segmentación en cualquier conjunto de datos. Para alcanzar este propósito, se han explorado y aplicado dos algoritmos genéticos en diversas bases de datos. En primer lugar, se examinan las similitudes entre ambos algoritmos, seguido por un análisis detallado de sus particularidades.

3.1. Parámetros optimizados

Cada método de segmentación está compuesto por una serie de procesos que, a su vez, están definidos por distintos parámetros. La elección adecuada de estos parámetros es crucial para lograr una segmentación efectiva. Los algoritmos propuestos consideran cada uno de estos parámetros como un individuo, y mediante un proceso de selección, identifican al más apto entre una vasta población de posibles candidatos. En la Tabla 3.1 se presentan los parámetros utilizados para cada uno de los procesos en los algoritmos propuestos.

3.1.1. CLAHE

La técnica CLAHE (Ecuilización de Histogramas Adaptativa Limitada por Contraste, por sus siglas en inglés Contrast Limited Adaptive Histogram Equalization) representa una evolución de la tradicional ecualización de histograma, diseñada para superar las limitaciones asociadas con la distorsión del contraste en imágenes con iluminación desigual. Es particularmente útil en aplicaciones de visión por computadora y procesamiento de imágenes, donde se busca mejorar el contraste en regiones específicas de una imagen que presentan una alta

Método	Parámetro	Intervalo
CLAHE	Distribución	Uniforme, Rayleigh
	Alfa (Rayleigh y exponencial)	Exponencial [0.2, 0.5]
	Losas	8, 16, 32, 64
	Límite de recorte	[0.02, 0.05]
	Histograma de contenedores	128, 256
Filtro		3, 5, 7
Gaussiano	(Tamaño y σ)	$\sigma \in [1, 4]$ y $[0, 1]$
Filtro Promedio	Tamaño	3, 5, 7 y 9
Segmentación	Algoritmo	Otsu, Adaptativo K-means, PCA
Espacio de Color	RGB, NTSC, HSV, XYZ, YCBCR	1, 2, 3, 4, 5
Canal	Número de canal	1, 2, 3, 4, 5 o todos los canales
Operadores	Erosión, Dilatación	1, 2, 3, 4
Morfológicos	Abierto, Cerrado	Orden

Tabla 3.1: Parámetros optimizados con el algoritmo genético propuesto.

variabilidad de intensidad lumínica.

La retinopatía diabética representa uno de los mayores desafíos en el campo médico, ya que la detección temprana de estas anomalías en la retina es crucial para prevenir la ceguera. La observación de la red de vasos sanguíneos en el fondo del ojo proporciona información vital para el análisis de enfermedades. Estos vasos sanguíneos se pueden visualizar de manera clara en imágenes segmentadas del fondo del ojo.

Para abordar esta problemática médica, existen diversas técnicas de segmentación automática de imágenes retinianas, que se benefician de la aplicación del algoritmo CLAHE. El principal objetivo de estos algoritmos evolutivos es mejorar la segmentación de los vasos sanguíneos, aprovechando la asistencia proporcionada por el algoritmo CLAHE.

3.1.2. Espacios de Color

A veces es necesario implementar otros espacios de color para realizar una segmentación más adecuada, basada en la información específica de la imagen. En la Figura 3.1, se presentan cuatro ejemplos de imágenes. En la primera columna se muestran las imágenes originales en el espacio de color RGB. La segunda columna muestra las mismas imágenes segmentadas utilizando el método de Otsu. A pesar de que este método optimiza el umbral de segmentación, se observa una calidad de segmentación deficiente en las imágenes.

La tercera columna muestra nuevamente las imágenes originales, pero esta

vez en el espacio de color Lab. En la cuarta columna se presentan las mismas imágenes, pero segmentadas utilizando el método de Otsu. Como se puede apreciar, el cambio al espacio de color Lab mejora la calidad de la segmentación en algunos casos.

En resumen, la elección del espacio de color adecuado puede mejorar significativamente la calidad de la segmentación. Sin embargo, esta elección debe basarse en las características específicas del conjunto de datos en cuestión.

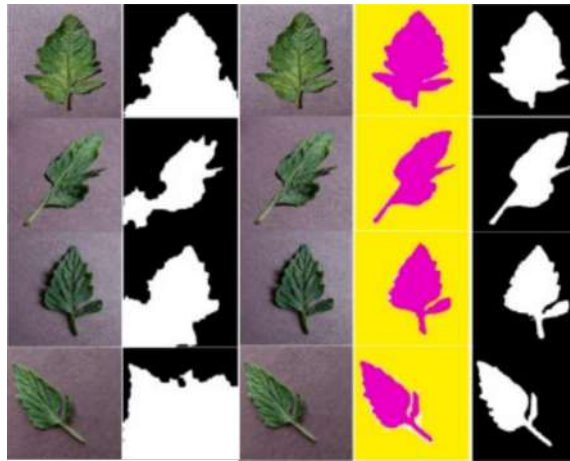


Figura 3.1: Segmentación Usando Diferentes Espacios de Color

3.1.3. Canal G

Una imagen digital en color se compone de millones de puntos dispuestos en filas y columnas conocidos como píxeles. Cada píxel puede representar diferentes combinaciones de los colores primarios. Estos colores primarios tienen escalas de tonalidad, comúnmente conocidas como canales. En el caso de una imagen RGB (rojo, verde, azul), se utilizan tres canales para representar los tres colores primarios.

En investigaciones recientes que emplean el algoritmo CLAHE, se ha observado que procesar imágenes utilizando únicamente el canal verde puede mejorar el contraste en el fondo del ojo [61]. Sin embargo, es importante destacar que al utilizar solo el canal verde, se reduce el tamaño de la imagen y puede haber pérdida de información al excluir los otros dos canales. A pesar de ello, la mayoría de los investigadores concuerdan en que el canal verde ofrece el mayor contraste entre los vasos sanguíneos y el fondo en estas imágenes.

En la metodología propuesta para el algoritmo genético 1, se ha optado por utilizar exclusivamente el canal verde (Canal G).

3.1.4. Subconjunto de Datos Seleccionados

Es necesario elegir un subconjunto representativo de imágenes para trabajar con ellas. Estas imágenes seleccionadas deben abarcar las diversas características presentes en todo el conjunto de datos, como diferentes niveles de iluminación (alta y baja), presencia de sombras en las imágenes y variedad de formas. Este subconjunto de datos se segmenta manualmente y la segmentación resultante se considera como la óptima. Esta segmentación óptima servirá como guía para el algoritmo genético.

3.1.5. Filtro Gaussiano

Los filtros digitales comprenden un conjunto de técnicas cuyo objetivo primordial es mejorar diversas características de una imagen, tales como suavizar, eliminar ruido, realzar bordes y detectarlos. En esencia, los filtros son operaciones aplicadas a los píxeles de una imagen con el fin de lograr efectos específicos en ella [12].

Existen varios tipos de filtros que se utilizan principalmente en el preprocesamiento digital de imágenes para mejorar los resultados durante la segmentación. Un ejemplo común es el filtro Gaussiano, donde el valor máximo se encuentra en el píxel central de la imagen y disminuye gradualmente hacia los bordes. La rapidez con la que disminuye este valor está inversamente relacionada con el parámetro de desviación estándar σ [96].

En estos experimentos, se consideró beneficioso aplicar el filtro Gaussiano antes de implementar las técnicas de segmentación. Los parámetros que se optimizaron fueron la desviación estándar σ y el tamaño del filtro S_f .

3.1.6. Filtro Promedio

El filtro promedio es una técnica de procesamiento de imágenes utilizada para suavizar la imagen y reducir el ruido. Funciona calculando el valor promedio de un conjunto de píxeles en una región determinada y asignando este valor promedio al píxel central de esa región. El proceso implica el desplazamiento de una ventana de tamaño específico a lo largo de toda la imagen, sustituyendo los valores de los píxeles por el valor promedio calculado de los píxeles dentro de la ventana.

El tamaño de la ventana afecta el grado de suavizado: cuanto mayor sea la ventana, más pronunciado será el suavizado, mientras que ventanas más pequeñas pueden preservar detalles más finos pero pueden no eliminar tanto el ruido.

El uso de este filtro puede mejorar la calidad visual de la imagen al suavizar transiciones abruptas y reducir el ruido, lo que facilita la detección de características importantes en la imagen [12].

3.1.7. PCA

El Análisis de Componentes Principales (PCA por sus siglas en inglés, Principal Component Analysis) es una técnica ampliamente utilizada en el análisis estadístico, visión por computadora, reconocimiento de patrones y diversas áreas donde se manejan grandes cantidades de variables. La función principal de PCA es reducir la dimensionalidad de un conjunto de datos para identificar y representar una estructura mejorada de variables, lo que simplifica el análisis y mejora la interpretación de los datos.

Los componentes principales, obtenidos a partir de las variables originales, están ordenados en función de la cantidad de varianza que explican en los datos originales.

3.1.8. Métodos de Segmentación

En el procesamiento digital de imágenes, la segmentación se refiere a la tarea de identificar y separar la región de interés de una imagen del fondo circundante. Este proceso es fundamental para muchos análisis y aplicaciones de visión por computadora. La técnica más comúnmente utilizada para lograr esto es la binarización, que implica encontrar un umbral que distinga las intensidades de la región de interés (generalmente representada en blanco) de las del fondo (representado en negro) [34]. En este trabajo, se exploran tres métodos de segmentación: Umbral Adaptativo, Método de Otsu y Método de K-Means.

El Método de Otsu determina un umbral óptimo para la separación de las intensidades de píxeles en dos clases: la región de interés y el fondo. Este umbral se selecciona para minimizar la varianza intraclase y maximizar la varianza interclase en relación con el umbral [68].

El método de umbral adaptativo, también conocido como método de umbral variable, es una de las técnicas más comúnmente empleadas debido a su bajo costo computacional y su capacidad para adaptarse a condiciones variables de iluminación y ruido en una imagen. Este enfoque es particularmente útil cuando una o varias umbrales globales no son adecuados para segmentar la imagen de manera efectiva. En esencia, este método calcula un valor de umbral T para cada píxel (x, y) de la imagen, utilizando información local de las regiones circundantes Z_{xy} [77] [98]. En otras palabras, el umbral se calcula en función de las propiedades locales de la imagen, lo que permite una adaptación más precisa a las características de contraste en diferentes partes de la imagen.

La fórmula (3.1) muestra la manera en que se calcula el umbral T_{xy} en función de las características locales de la imagen.

$$T_{xy} = a\sigma_{xy} + bm_{xy} \quad (3.1)$$

Donde:

- a y b son constantes no negativas
- σ_{xy} y $m_{xy} \in Z_{xy}$

Posteriormente se aplican las formulas (3.2) y (3.3) para obtener la imagen segmentada.

$$g(x, y) = 1 \quad si \quad f(x, y) > T_{xy} \quad (3.2)$$

$$g(x, y) = 0 \quad si \quad f(x, y) \leq T_{xy} \quad (3.3)$$

Donde:

- $g(x, y)$ Es el resultado de nuestra segmentación, asignando el valor de 1 a los pixeles que conforman la zona de interés y el valor de 0 a los pixeles que son el fondo.
- $f(x, y)$ Imagen binarizada en escala de grises.

Por último, el método de k-means es un algoritmo de agrupamiento no supervisado que clasifica un conjunto de n elementos en k grupos según sus características, asignando cada elemento al grupo más cercano [54]. El algoritmo consta de los siguientes pasos:

1. Inicialización de Centroides: Una vez determinado el número de grupos k , se seleccionan aleatoriamente k puntos como centroides en el espacio de todas las piezas. Estos centroides representarán el centro de cada grupo inicial.
2. Asignación de Puntos a Centroides: Se calculan las distancias entre cada pieza y todos los centroides. Luego, cada pieza se asigna al centroide más cercano en función de estas distancias, formando así k grupos iniciales.
3. Actualización de Centroides: Se recalcula la posición de cada centroide moviéndolo hacia el centro geométrico de las piezas asignadas a él. Este nuevo centroide es esencialmente la media de todas las piezas pertenecientes al grupo.
4. Convergencia: Los pasos 3 y 4 se repiten iterativamente hasta que no haya cambios significativos en las posiciones de los centroides entre iteraciones. Esto se logra cuando los centroides ya no se mueven o la variación en sus posiciones es mínima.

Este proceso iterativo garantiza que los centroides converjan hacia posiciones estables que representen de manera óptima los grupos de datos en el espacio multidimensional

3.1.9. Operadores Morfológicos

En el ámbito del procesamiento digital de imágenes, los operadores morfológicos han sido ampliamente utilizados para el procesamiento de imágenes en escala de grises y binarias. Estos operadores desempeñan un papel crucial en una variedad de aplicaciones, incluida la reducción de ruido, la reconstrucción de formas de objetos, la detección de características y, en general, la mejora

de la calidad de la imagen. Los operadores morfológicos se basan en conceptos matemáticos que se inspiran en la morfología matemática, una rama de las matemáticas que se ocupa del estudio de las formas y las estructuras.

Existen cuatro operadores morfológicos principales: erosión, dilatación, apertura y cierre. Estos operadores se utilizan tanto en imágenes en escala de grises como en binarias. La erosión es un proceso que reduce el tamaño de los objetos en una imagen, mientras que la dilatación aumenta el tamaño de los objetos. La apertura es una operación que combina la erosión seguida de la dilatación, y se utiliza principalmente para eliminar pequeños objetos y ruido en los bordes de los objetos más grandes. Por otro lado, el cierre es una operación que combina la dilatación seguida de la erosión, y se utiliza para cerrar pequeños huecos en los objetos y suavizar los contornos [26].

En este trabajo, se aplicaron los cuatro operadores morfológicos como técnicas de postprocesamiento para mejorar la calidad de la imagen segmentada. Estas técnicas jugaron un papel fundamental en la eliminación de artefactos no deseados y en la mejora de la precisión de la segmentación obtenida.

3.2. Algoritmo Genético

Dada la complejidad e información inherente a los distintos escenarios de imágenes, resulta crucial integrar un algoritmo genético que busque sistemáticamente la combinación óptima de parámetros y técnicas para lograr los mejores resultados de segmentación.

El algoritmo genético se encarga de identificar las técnicas que generan los mejores resultados de segmentación. En la Figura 3.2 se pueden apreciar las diversas técnicas disponibles. Cada combinación de estas técnicas, junto con sus respectivos parámetros, se codifica para formar un cromosoma que representa a un individuo en el proceso evolutivo. Posteriormente, se evalúa a cada individuo, asignándole un valor numérico que representa su aptitud, al tiempo que se evalúa la calidad de la segmentación resultante.

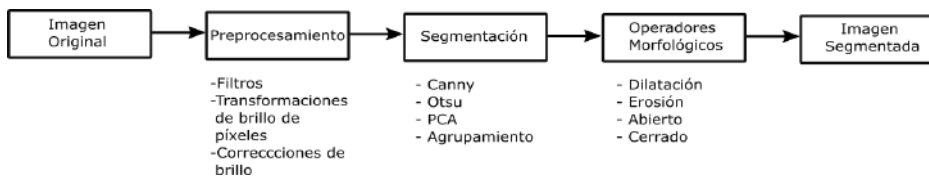


Figura 3.2: Técnicas del AG

Las soluciones obtenidas en esta primera etapa pasan a ser refinadas mediante la recombinación y la mutación, dando lugar a nuevos individuos con una mayor aptitud en cada generación subsiguiente. Los individuos con una aptitud más alta tendrán una probabilidad más alta de persistir en cada iteración, lo que aumenta las posibilidades de que sus genes se transmitan a la siguiente

generación. Todo este proceso se lleva a cabo de forma iterativa durante un número específico de generaciones (según la condición de parada), o hasta que se alcance la convergencia hacia una solución óptima. El proceso detallado de este algoritmo genético se muestra en la Figura 3.3.

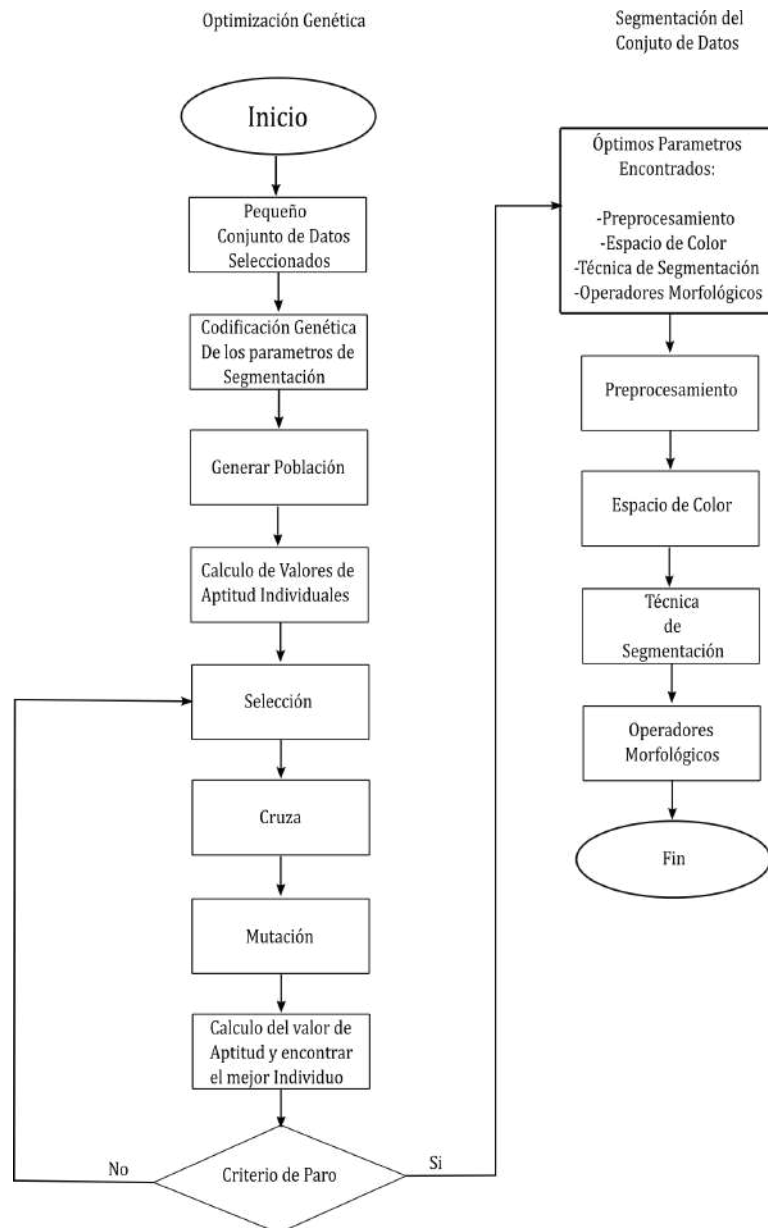


Figura 3.3: Método Propuesto AG

1. Población Inicial. La población inicial es generada por una forma aleatoria, es decir, se crea un conjunto de individuos artificialmente. Una población se puede definir por $S^g = \{S_1^g, S_2^g, \dots, S_n^g\}$, donde g representa la generación actual donde se está iterando y n simboliza el número de individuos que constituye la población. En el algoritmo y experimentos realizados, se trabaja con una generación de 120 individuos, quedando la fórmula de la siguiente forma:

$$S^g = \{S_1^g, S_2^g, \dots, S_{120}^g\}. \quad (3.4)$$

El vector que representa a cada individuo esta dado por:

$$s_i^g = \{s_{i,1}^g, s_{i,2}^g, \dots, s_{i,d}^g\} \quad (3.5)$$

donde i es el individuo con el que se esta trabajando y d es la dimensión del vector, en esta tesis se trabajó con once variables, quedando finalmente la ecuación como:

$$s_i^g = \{s_{i,1}^g, s_{i,2}^g, \dots, s_{i,11}^g\} \quad (3.6)$$

Los 120 vectores se inicializan de manera aleatoria en el espacio de búsqueda. Luego, se evalúa a cada individuo dentro de la población para determinar su aptitud, la cual es esencialmente una medida de la calidad de la segmentación.

2. Evaluación de los individuos. Individualmente cada cromosoma contiene en sus alelos la información necesaria (variables) para segmentar una imagen. Las variables definen el procesamiento, la técnica de segmentación empleada, los operadores morfológicos aplicados, el espacio de color seleccionado, así como su canal de trabajo. Cada combinación de estas variables da como resultado una distinta imagen segmentada con una calidad de segmentación en particular. La evaluación de esta calidad de segmentación se obtiene por medio de una comparación con la segmentación objetivo o la segmentación manual.
3. Selección. El proceso de selección es mediante un método de azar llamado “La ruleta” donde no todos los individuos tienen la misma probabilidad de ser elegidos, la probabilidad se incrementa proporcionalmente a la aptitud del individuo, así que los individuos que tienen más oportunidad de trascender a la siguiente generación serán los más aptos.

La probabilidad de que cada individuo sea seleccionado se calcula de la siguiente manera:

$$P_i = \frac{f(S_i)}{\sum_{j=1}^n f(S_j)} \quad (3.7)$$

donde $f(S_i)$ es el valor de aptitud del individuo y las variables i , y n representa el número de individuo y la población respectivamente.

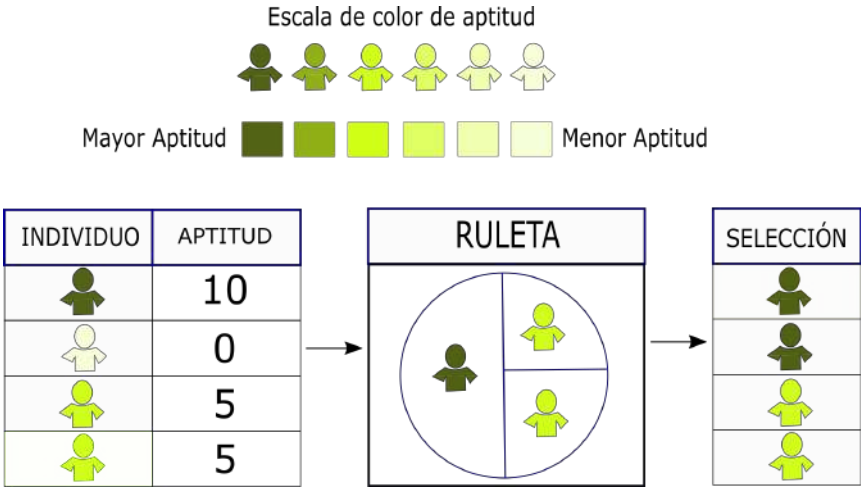


Figura 3.4: Proceso de Selección por el método de Ruleta

4. Cruza. Este mecanismo de evolución es esencial en los algoritmos genéticos y se caracteriza por la combinación de material genético entre padres y como se transmite esa información genética a los hijos, en este trabajo se realizó un cruce a dos puntos.

Entrada:	Aptitud de cada individuo f_i
Salida:	Individuos seleccionados
1 :	Calcular la suma total de aptitudes de los n individuos $\sum_{i=1}^n f_i$
2 :	Generar un número aleatorio $r \in [0, 1]$
3 :	Obtener el valor de $s = r * \sum_{i=1}^n f_i$
4 :	Si $s \leq \sum_{i=1}^n f_i$ entonces se selecciona el kth individuo oara la siguiente generación
5 :	Repetir pasos del 2 al 4 hasta que el número de individuos seleccionados sea igual a n

Tabla 3.2: Algoritmo de Selección Ruleta.

La cruza se define como la combinación de dos individuos para obtener información genética de ambos cromosomas, siendo un proceso fundamental para la transmisión de información genética de padres a hijos. Aunque existen varios tipos de cruces, en estos experimentos se implementó el cruce a dos puntos. En la Figura 3.5 se detalla el proceso de este tipo de cruce.

De dos padres seleccionados $P1$ y $P2$, se seleccionan aleatoriamente dos puntos de cruce en cada cadena cromosómica del individuo, de tal manera que los padres $P1$ y $P2$ se dividen en tres secciones, es decir, $P1 = a_1b_1c_1$ y $P2 = a_2b_2c_2$. Al combinar la información de los padres, generaran dos descendientes intercambiando las ubicaciones de las secciones b_1 y b_2 , finalizando con las siguientes secciones de la descendencia $H1 = a_1b_2c_1$ y $H2 = a_2b_1c_2$

5. Mutación. El uso de esta herramienta de la herencia permite modificar la cadena cromosómica de forma aleatoria; esto suele suceder normalmente cuando se genera un error en la reproducción, conocido como deformación genética. La probabilidad de mutación en la evolución natural no es tan alta como en los algoritmos genéticos. Los dos principales objetivos de la mutación son los siguientes:

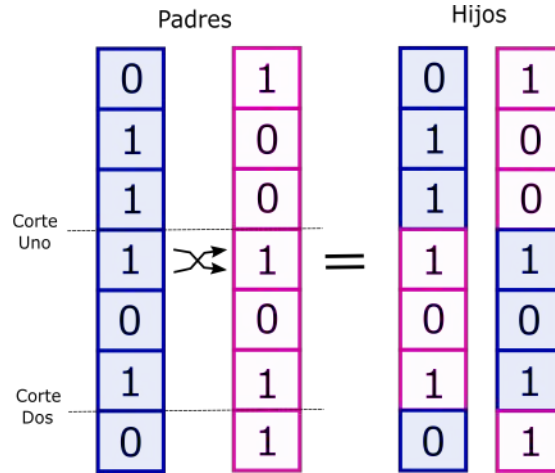


Figura 3.5: Operador de cruce de dos puntos en un Algoritmo genético

- a) La mutación desempeña un papel crucial al introducir variabilidad en la población de cromosomas. Este proceso implica que un cromosoma altere aleatoriamente su información genética, lo que le permite explorar nuevas áreas del espacio de búsqueda y potencialmente descubrir soluciones más óptimas. Es como si el algoritmo genético permitiera a los cromosomas dar saltos hacia lo desconocido, abriendo la puerta a la diversidad genética y facilitando la búsqueda de soluciones más efectivas.
- b) Otro aspecto importante de la mutación es su capacidad para evitar que el algoritmo genético se quede atrapado en mínimos locales. Al introducir cambios aleatorios en los cromosomas, la mutación puede sacar al algoritmo de estos mínimos locales al proporcionar una exploración más amplia del espacio de búsqueda. De esta manera, la

mutación actúa como un mecanismo de escape que promueve la diversidad genética y aumenta las posibilidades de encontrar la solución óptima global.

En los experimentos de este algoritmo genético, se utilizó una probabilidad de mutación = 0,08

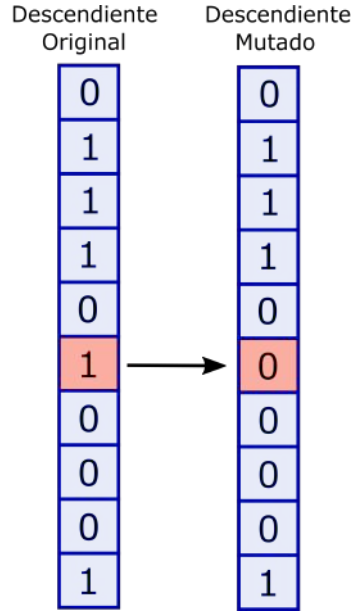


Figura 3.6: Operador de Mutación en un algoritmo genético

La mutación consiste en seleccionar aleatoriamente un individuo (i) de la población y elegir un valor dentro de sus alelos (v) para cambiar su valor. Si $S(i, v) = 1$ entonces se cambia a 0 ($S(i, v) \leftarrow 0$). Por el contrario, Si $S(i, v) = 0$ entonces se cambia 1 ($S(i, v) \leftarrow 1$).

6. **Elitismo.** El concepto de elitismo se resume en preservar a los mejores individuos de una generación para las siguientes iteraciones, asegurando que las soluciones más óptimas no se pierdan a medida que avanza la búsqueda. En el algoritmo genético 2, se implementa un enfoque de elitismo modificado. En este caso, solo se replica un cromosoma en la siguiente generación: aquel con la mejor aptitud de toda la población. Este enfoque no solo garantiza la conservación de las mejores soluciones, sino que también contribuye a evitar que el algoritmo quede atrapado en óptimos locales.

Es importante destacar que el algoritmo finaliza después de diez iteraciones si no se observa una mejora en la evolución de los individuos.

La fórmula utilizada para calcular el elitismo se expresa como sigue:

$$f(S_{best}^{g+1}) = \max(f(S_i^g)) \quad (3.8)$$

3.2.1. Codificación Genética

La codificación genética es la parte central de la metodología propuesta. En el algoritmo genético se trabaja con individuos o cromosomas que a su vez están conformados por los alelos donde cada alelo representa cada variable del algoritmo. La Figura 3.7 muestra las variables representadas en una cadena cromosómica.

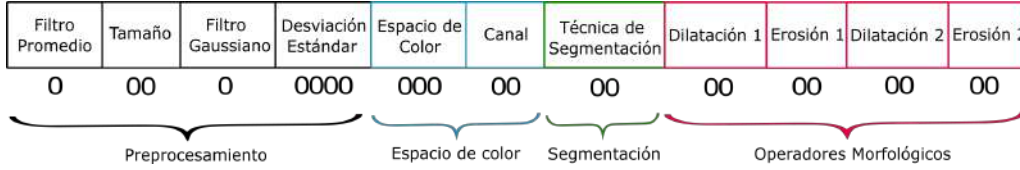


Figura 3.7: Codificado de las variables utilizadas del algoritmo propuesto

Donde:

- Filtro promedio: Es una variable discreta que adquiere el valor de (1) si se utiliza un filtro promedio o un (0) si no se utiliza.
- Tamaño del filtro: Es una variable discreta y puede adquirir los valores de 3, 5, 7 y 9.
- Filtro Gaussiano: Es una variable discreta que se representa con el valor de 1 si se utiliza un filtro gaussiano o un (0) si no se aplica un filtro gaussiano.
- Desviación Estándar: Es la única variable continua que puede tomar valores de 0 a 1. La desviación estándar pertenece al filtro Gaussiano.
- Segmentación: Es una variable discreta y puede representarse por cuatro valores que son los cuatro tipos de segmentación a utilizar (agrupación, PCA, umbral adaptativo y método Otsu).
- Dilatación: Esta variable también es discreta y puede tomar los valores 1, 2, 3 y 4. Los cuatro valores son los tamaños de los círculos a dilatar.
- Erosión: La erosión tiene las mismas características que la dilatación en cuanto al tipo y número de variables.
- Espacio de color: Es una variable discreta y puede ser representada por los cinco valores de espacios de color aplicado (RGB, NTSC, HSV, XYZ, YCBCR).

- Canal: Variable discreta que puede ser representada por cuatro configuraciones según el canal utilizado (Primer canal, Segundo canal, Tercer canal o todos los canales).

Los últimos dos alelos del cromosoma son llenados nuevamente por las variables de dilatación y erosión, tomando aleatoriamente nuevos valores para mejorar la calidad de la segmentación. El tiempo de dilatación y erosión empleados son diferentes, y según la combinación utilizada genera una segmentación final diferente. Cada variable es decodificada por la Ecuación 3.9 para obtener su valor. Como se ilustra en la Figura 3.7, casi todas las variables son discretas; a excepción de una; esta es la desviación estándar utilizada para el filtro gaussiano. En el caso de la desviación estándar, se normalizó en $[0,1]$ con pasos de 0.1 y para estos posibles valores solo se necesitan cuatro dígitos para decodificarla.

$$x_i = x_i^l + \left(\frac{x_i^u - x_i^l}{2^\beta - 1} \right) \gamma \quad (3.9)$$

Donde:

- β : Número de dígitos del parámetro
- x_i^l : Límite inferior
- x_i^u : Límite superior
- γ : Valor del parámetro a decodificar

Capítulo 4

Segmentación Basada en PSO

En este capítulo se expone de manera detallada el enfoque y los procedimientos empleados en el método de optimización por cúmulo de partículas (Particle Swarm Optimization -PSO-), haciendo énfasis en sus propiedades y componentes de la fórmula que la conforman. El propósito primordial de esta metodología es mejorar la segmentación en diversos conjuntos de datos. Con este fin, se ha aplicado este algoritmo evolutivo en diferentes bases de datos para obtener resultados óptimos en el proceso de segmentación.

4.1. Componentes del Algoritmo propuesto

La metodología de PSO se basa en estudios sobre el comportamiento de movimiento de aves y peces. Cuando un pez detecta alimento, transmite información a los demás sobre la ubicación del mismo y cada pez va actualizando su posición con respecto a información del cúmulo y de él mismo. De manera similar, el algoritmo PSO permite encontrar la solución óptima aprovechando la colaboración entre todos los individuos del enjambre.

En esta sección se presenta el algoritmo de segmentación desarrollado, el cual está basado en Optimización por Cumulos de Partículas. El método propuesto busca mejorar la segmentación de imágenes mediante la optimización de parámetros relevantes en el proceso de segmentación.

La Figura 4.1 ilustra la metodología seguida para llevar a cabo la segmentación óptima. El proceso inicia con la imagen de entrada, que es transformada a un espacio de color adecuado para facilitar la identificación de las regiones de interés. Posteriormente, se selecciona un canal específico de la imagen para su procesamiento, el cual es sometido a una etapa de preprocesamiento con el fin de mejorar la calidad de la imagen y reducir el ruido.

Después del preprocesamiento, se lleva a cabo la segmentación inicial, seguida por la aplicación de operadores morfológicos que refinan los contornos y

estructuras de los objetos segmentados. Paralelamente, se cuenta con una segmentación de referencia, la cual sirve como criterio de comparación para evaluar la calidad de los resultados obtenidos.

En la etapa de optimización, se emplea el algoritmo PSO para ajustar los parámetros de segmentación con el objetivo de obtener una segmentación óptima. El PSO recibe como entrada tanto la segmentación obtenida tras el preprocesamiento y los operadores morfológicos como la segmentación de referencia (Ground Truth), y a través de su proceso iterativo, optimiza los valores de los parámetros hasta alcanzar la mejor correspondencia posible con la referencia.

Finalmente, el resultado de este procedimiento es una segmentación óptima que mejora la precisión y la calidad de la identificación de regiones en la imagen, logrando así un método más robusto y eficiente para la segmentación basada en PSO.

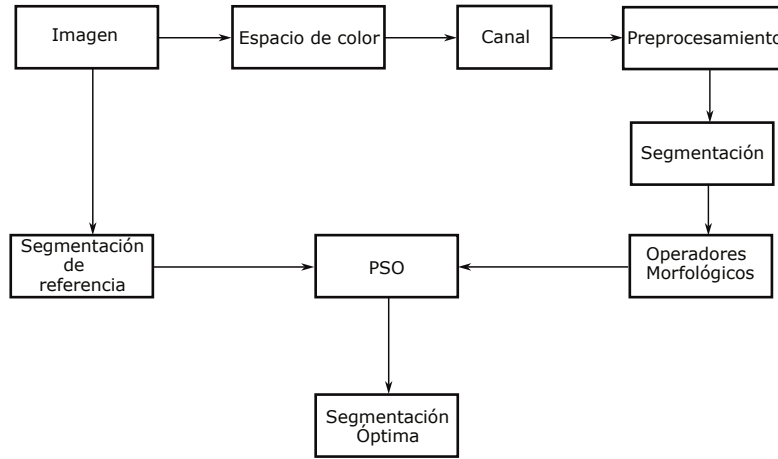


Figura 4.1: Segmentación usando PSO

En las siguientes subsecciones se muestra el algoritmo en detalle y los pasos para optimizar la segmentación. Las siguientes subsecciones muestran los parámetros a optimizar por el PSO.

4.1.1. Espacios de color

Uno de los parámetros a optimizar por el PSO es el espacio de color. El espacio de color utilizado influye significativamente en la segmentación de imágenes, esto debido a que el espacio de color determina como se representan y diferencian los colores en una imagen. Esto influye directamente en la precisión y eficiencia de los métodos de segmentación [102] [10] [43].

Esto es, los espacios de color pueden mejorar la separación entre los objetos de interés y el fondo. Es sabido que en el espacio RGB, los canales están altamente correlacionados, lo que puede dificultar la segmentación, Mientras que,

en espacios como HSV o Lab pueden facilitar la segmentación al representar mejor las diferencias perceptuales entre colores.

Además, el espacio RGB es sensible a cambios en la iluminación, lo que puede afectar la segmentación. Espacios como HSV, YCbCr o Lab pueden reducir este problema, ya que separan la información de luminancia de la crominancia.

Es importante señalar que dependiendo del espacio de color, ciertos algoritmos de segmentación pueden ser más eficientes. Para segmentación basada en clustering (como K-means), el espacio de color Lab puede proporcionar mejores resultados debido a su percepción uniforme del color. En técnicas basadas en redes neuronales o aprendizaje profundo, espacios como YUV pueden mejorar la representación de características.

En conclusión, la elección del espacio de color es un factor crucial en la segmentación de imágenes, ya que puede mejorar la precisión, reducir el ruido y hacer que los métodos sean más robustos ante variaciones de iluminación y textura. Con el objetivo de hacer uso de las ventajas que suponen los espacios de color, el algoritmo basado en PSO utiliza diferentes espacios de color (RGB, NTSC, HSV, XYZ y YCbCr). El uso de diferentes espacios de color puede ayudar a mejorar la calidad de segmentación. En los experimentos llevados a cabo, la variable que define el espacio de color es $C = 0, 1, \dots, 4$ y tiene asociados valores entre 0 y 4.

4.1.2. Canales de color

Por otro lado, Algunos espacios de color permiten aplicar umbrales de forma más efectiva. Esto es muy importante sobre todo cuando se utilizan métodos de segmentación basados en umbrales. En HSV, el canal de matiz (H) puede usarse para detectar colores específicos sin importar su intensidad, mientras que en el espacio de color YCbCr, los canales Cb y Cr pueden ayudar por ejemplo a detectar tonos de piel en aplicaciones de reconocimiento facial. Esto es, en algunas ocasiones solo es necesario un solo canal del espacio de color. El algoritmo PSO presentado en esta sección puede hacer uso de los tres canales o de uno solo. En nuestro caso, la variable que define el canal de color es $Ch = 0, 1, \dots, 3$ y tiene asociados valores entre 0 y 3. Un valor seleccionado de 0 significa que se utilizan todos los canales, mientras que al seleccionar 1, 2 o 3 se selecciona automáticamente ese canal de color.

4.1.3. Pre procesamiento

El preprocesamiento de imágenes es una etapa muy importante para la segmentación de imágenes, ya que mejora la calidad de las imágenes antes de aplicar técnicas de segmentación. Su principal objetivo es reducir el ruido, mejorar el contraste y resaltar características relevantes, lo que permite a los algoritmos de segmentación implementarse con mayor precisión y robustez.

A continuación, se describen tres técnicas de preprocesamiento comunes y su impacto en la segmentación:

Filtro Promedio

Este filtro es una técnica de suavizado que reemplaza cada píxel con el promedio de los valores de los píxeles que se encuentran en una vecindad definida. El principal beneficio en la segmentación es la reducción de ruido aleatorio en la imagen, lo que disminuye la aparición de artefactos no deseados en la segmentación basada en umbrales. Sin embargo, puede provocar pérdida de detalles finos y reducir la nitidez de los bordes, sobre todo cuando aumentamos el tamaño del filtro. Esto podría afectar la precisión en técnicas que dependen de contornos bien definidos. El algoritmo PSO presentado en esta sección puede elegir el tamaño adecuado de filtro y si es adecuado su uso. En nuestro caso, la variable que define el filtro promedio es $P = 0, 1, \dots, 3$ donde 0 no usa el filtro, 1 utiliza un filtro de 3×3 , 2 utiliza un filtro de 5×5 y 3 utiliza un filtro de 7×7 .

Filtro Gaussiano

Este filtro obtiene resultados similares al filtro promedio, esta es una técnica de suavizado que utiliza una distribución gaussiana para asignar mayor peso a los píxeles centrales en una vecindad. Su uso en segmentación mejora la eliminación de ruido sin afectar tanto los bordes en comparación con el filtro promedio. Esto permite obtener una segmentación más precisa en aplicaciones donde los bordes son críticos, como en la detección de objetos en imágenes naturales o imágenes biomédicas. En nuestro caso, la variable que define el filtro Gaussiano es $G = 0, 1, \dots, 3$ donde 0 no usa el filtro, 1 utiliza un filtro de 3×3 , 2 utiliza un filtro de 5×5 , 3 utiliza un filtro de 7×7 . El valor de σ es fijado en 1,2.

CLAHE

La técnica de preprocesamiento Contrast Limited Adaptive Histogram Equalization (CLAHE) es una técnica de mejora de contraste [13]. CLAHE permite ajustar la intensidad de los píxeles en bloques locales de la imagen, evitando la sobreexposición en regiones brillantes y la pérdida de detalles en áreas oscuras. Su aplicación en segmentación es de suma importancia, ya que permite mejorar la visibilidad de estructuras con bajo contraste, facilitando una separación más precisa de las regiones. En particular, resulta útil en imágenes médicas y de baja iluminación donde los bordes y detalles finos son esenciales. En nuestro caso, la variable que define el filtro CLAHE es $CL = 0, 1, \dots, 3$ donde 0 no usa el filtro, los valores de 1 a 3 definen el tamaño de los bloques en los que se dividirá la imagen antes de aplicar la ecualización. 1 para un tamaño pequeño 4×4 , 2 para mediano 8×8 y 3 para un tamaño grande 16×16 . Un tamaño pequeño mejora el contraste local, pero puede producir ruido, mientras que un tamaño grande 16×16 suaviza la imagen.

El uso adecuado de las técnicas de preprocesamiento permite mejorar la segmentación en diferentes escenarios. CLAHE mejora la detección de bordes y estructuras con bajo contraste, el filtro promedio suaviza la imagen y reduce ruido pero puede perder detalles, y el filtro Gaussiano ofrece un equilibrio entre

suavizado y conservación de bordes. La elección de la técnica de preprocesamiento depende del tipo de imagen y del método de segmentación que se piense utilizar.

4.1.4. Segmentación

En estos experimentos se utilizaron tres métodos de segmentación: Otsu, PCA y segmentación adaptativa. La variable que define el tipo de segmentación esta dado por $S = 1, \dots, 3$ donde 1 representa la segmentación por PCA, 2 segmentación por el método Otsu y 3 segmentación adaptativa

4.1.5. Operadores morfológicos

Los operadores morfológicos son técnicas fundamentales en el procesamiento de imágenes. Los operadores de dilatación y erosión son especialmente en la segmentación y análisis de estructuras en imágenes binarias y en escala de grises. Ambos operadores utilizan un elemento estructurante para modificar la geometría de los objetos en la imagen.

La erosión reduce el tamaño de los objetos en una imagen al eliminar píxeles en los bordes. Se usa para eliminar ruido, separar objetos conectados y reducir detalles pequeños. Reduce el tamaño de los objetos segmentados, lo que ayuda a separar regiones que estén conectadas por ruido o puentes delgados. Sin embargo, puede eliminar regiones pequeñas si el elemento estructurante es grande, lo que puede ser perjudicial si las estructuras de interés son delgadas.

Por otro lado, la dilatación aumenta el tamaño de los objetos en la imagen al agregar píxeles en los bordes. Se usa para rellenar huecos, conectar componentes cercanos y suavizar formas. Es especialmente útil cuando la segmentación inicial deja regiones incompletas ya que expande el tamaño de los objetos segmentados, ayudando a recuperar estructuras erosionadas en pasos previos. Puede unir objetos cercanos, lo que es beneficioso si representan una sola región, pero puede ser problemático si deben estar separados. Es decir, su uso adecuado puede mejorar la calidad de los segmentos detectados, pero también puede introducir errores si no se aplican correctamente.

En los experimentos llevados a cabo, se utilizaron dilatación y erosión. La variable que define el tipo de operador morfológico esta dado por $M = 1, \dots, 4$ donde 1 representa el uso de un operador de dilatación, 2 el uso de un operador de erosión, 3 el uso de un operador de cerradura y 4 el uso de un operador de abertura.

4.2. Algoritmo propuesto

El algoritmo propuesto combina el PSO y varios espacios de color, canales de color, algoritmos de pre-procesamiento y segmentación. Este enfoque construye un algoritmo con fuertes capacidades para segmentar al realizar una búsqueda

global de la combinación de las mejores técnicas. Adicionalmente, el algoritmo realiza una búsqueda de la mejor combinación de operadores morfológicos después de la segmentación para mejorar la calidad de la segmentación final.

El algoritmo PSO se basa en la cooperación y el intercambio de información entre múltiples individuos dentro de un espacio de búsqueda. Cada individuo es denominado partícula y representa una posible solución del problema. Las partículas o individuos ajustan su posición iterativamente en función de su propia experiencia y la influencia del resto del enjambre.

El algoritmo consta de n partículas y la posición de la partícula ith ($1 \leq i \leq m$) mide la solución potencial en un espacio d -dimensional. La ith partícula es expresada como:

$$x_i = (x_{i1}, x_{i2}, \dots, x_{id}) \quad (4.1)$$

En nuestro caso, cada partícula está definida por los componentes arriba propuestos. Es decir, la ith partícula es expresada como:

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ x_{i4} \\ x_{i5} \\ x_{i6} \\ x_{i7} \\ x_{i8} \\ x_{i9} \\ x_{i10} \end{bmatrix} = \begin{bmatrix} \text{Espacio de color} \\ \text{Canal} \\ \text{Filtro Promedio} \\ \text{Filtro Gauss} \\ \text{CLAHE} \\ \text{Segmentación} \\ \text{Dilatación} \\ \text{Erosión} \\ \text{Apertura} \\ \text{Cerradura} \end{bmatrix} \quad (4.2)$$

donde cada elemento de las partículas tiene un rango de valores, en este caso sus rangos fluctúan de la siguiente manera:

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ x_{i4} \\ x_{i5} \\ x_{i6} \\ x_{i7} \\ x_{i8} \\ x_{i9} \\ x_{i10} \end{bmatrix}, x_i \in \begin{bmatrix} 0,4 \\ 0,3 \\ 0,3 \\ 0,3 \\ 0,3 \\ 1,3 \\ 0,1 \\ 0,1 \\ 0,1 \\ 0,1 \end{bmatrix} \quad (4.3)$$

donde cada uno de los rangos se define como está descrito en las secciones de [4.1.1](#) a [4.1.5](#).

Tras la primera iteración, cada partícula i evalúa su posición actual mediante una función de aptitud (fitness) que cuantifica la calidad de la segmentación obtenida. Esta métrica en nuestro caso se evalúa mediante la precisión en bordes, para ello utilizamos inicialmente varias métricas que miden la calidad de

la segmentación como PRI, VOI y GCE. Sin embargo al final para este algoritmo nos decidimos por utilizar Delta Jaccard para medir el desempeño en la segmentación comparando con la segmentación de referencia (ground truth).

Las partículas actualizan su estado según tres principios fundamentales:

- Inercia
- Mejor posición (individual) o componente cognitivo que refleja el conocimiento individual de la partícula y su mejor posición encontrada hasta el momento (pbest)
- Mejor posición del cúmulo o componente social que representa el aprendizaje global del enjambre, guiando a la partícula hacia la mejor posición encontrada por cualquier miembro del grupo (gbest).

La velocidad de la partícula i en el tiempo $t + 1$ se puede calcular por medio de la siguiente formula:

$$v_i^{t+1} = w \cdot v_i^t + C_1 \cdot r_1 \cdot (pbest_i - x_i^t) + C_2 \cdot r_2 \cdot (gbest - x_i^t) \quad (4.4)$$

donde:

- w Es uno de los tres principios fundamentales que representa la Inercia
- v_i^t Es la velocidad de la partícula i en la iteración t
- C_1 es el primer coeficiente de aceleración.
- r_1 Es el primer vector aleatorio cuya componente está uniformemente distribuida entre $[0, 1]$
- $(pbest - x_i^t)$ Es el segundo de los tres principios fundamentales que corresponde a la mejor posición individual. El termino $Pbest$ representa la mejor posición del individuo i en la dimensión d en k eventos y x_i^t es la posición de la partícula i en la iteración t .
- C_2 es el segundo coeficiente de aceleración.
- r_2 Es el segundo vector aleatorio cuya componente está uniformemente distribuida entre $[0, 1]$
- $(gbest - x_i^t)$ Es el último de los tres principios fundamentales que corresponde a la mejor posición del cúmulo o también llamada componente social. El termino $gbest$ se interpreta como la mejor posición global y la mejor solución en la iteración t .

La posición de cada partícula se actualiza tomando en cuenta la siguiente formula:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (4.5)$$

donde:

- x_i^{t+1} Es la nueva posición de la partícula i .
- x_i^t Es la posición de la partícula i en la iteración t .
- v_i^{t+1} Es la nueva velocidad de la partícula i .

Una vez alcanzada la segmentación inicial, el algoritmo explora combinaciones de espacios de color, canales, técnicas de preprocesamiento y segmentación con operadores morfológicos (erosión, dilatación, apertura, cerradura) con elementos estructurantes de tamaño adaptativo. La selección óptima se guía por la reducción de ruido o la minimización de píxeles aislados y la preservación de detalles.

El proceso iterativo finaliza cuando se alcanza el máximo de iteraciones fijado de antemano.

A través de la iteración continua, las partículas exploran y explotan el espacio de búsqueda, equilibrando la exploración de nuevas regiones y la explotación de soluciones prometedoras. Este mecanismo permite que el algoritmo converja eficientemente hacia óptimos locales o globales en problemas de optimización complejos.

4.3. Exploración y Explotación

La exploración hace referencia al proceso mediante el cual las partículas se dispersan por gran parte del espacio de búsqueda. El objetivo principal de la exploración es evitar que las partículas queden atrapadas en óptimos locales, lo que permite encontrar la solución óptima en todo el espacio de búsqueda.

Este comportamiento de exploración ocurre cuando la velocidad de las partículas es lo suficientemente alta, lo que las hace más susceptibles a ser influenciadas por la mejor posición global que por su propia mejor posición individual.

Por otro lado, la explotación se refiere al proceso de identificar la mejor opción dentro de una zona específica del espacio de búsqueda que ya ha sido explorada. A diferencia de la exploración, la explotación se favorece cuando las partículas tienen una velocidad baja y están más influenciadas por su mejor posición individual que por la mejor posición global.

4.4. Factor de la inercia

El valor de la inercia se puede fijar como un valor constante entre 0,4 y 0,9, y no variará a lo largo de las iteraciones. Mantener el factor de inercia constante tiene como objetivo evitar que el algoritmo se quede atrapado en óptimos locales, sin que esto haga que la búsqueda sea excesivamente aleatoria. Un valor comúnmente utilizado en este tipo de algoritmos es $w = 0,7$.

Alternativamente, el valor de la inercia también puede ser considerado de manera decreciente, lo que significa que, con cada iteración, el valor de w disminuirá. Este enfoque permite que, al principio, el algoritmo explore de manera

más amplia el espacio de búsqueda, y a medida que avanza, se enfoque más en la explotación de las áreas más prometedoras.

La fórmula para actualizar w en cada iteración es la siguiente:

$$w(t) = w_{max} - \left(\frac{w_{max} - w_{min}}{T_{max}} \right) \cdot t \quad (4.6)$$

donde:

- w_{max} es el valor inicial de w .
- w_{min} es el valor final de w , que podría ser un valor pequeño (un ejemplo sería 0.4).
- T_{max} es el número máximo de iteraciones o generaciones.
- t Es el número actual de la iteración.

Una última forma de obtener el factor de inercia es de manera aleatoria, incorporando un componente estocástico adicional para mejorar la exploración en el espacio de búsqueda.

$$w = w_{min} + (w_{max} - w_{min}) \cdot r \quad (4.7)$$

donde:

- w_{max} es el valor inicial de w .
- w_{min} es el valor final de w .
- r es un número aleatorio en un rango de $[0,1]$.

En conclusión, el valor de w puede mantenerse constante durante todo el algoritmo, decrecer para priorizar la explotación o variar de manera aleatoria para favorecer la exploración. Cualquiera de estas tres opciones es válida y puede generar los resultados esperados. El problema a estudiar será el que determine la forma adecuada de seleccionar w .

4.5. Coeficientes de aceleración

Los coeficientes C_1 y C_2 son parámetros fundamentales para determinar el comportamiento de las partículas en un proceso de optimización. Cada uno de estos coeficientes tiene una función específica: C_1 busca optimizar la posición individual de las partículas, mientras que C_2 se enfoca en la mejor posición social. La forma de seleccionar estos valores es similar a la elección del factor de inercia. A continuación, se detallan las distintas maneras de determinar estos coeficientes. Una opción es asignar un valor predeterminado a los coeficientes C_1 y C_2 , generalmente dentro del rango $[1.5, 2]$. Esto se debe a que estos valores tienden a funcionar de manera efectiva en una variedad de problemas de

optimización. Al igual que el coeficiente de inercia, los coeficientes de aceleración también pueden ser calculados de manera dinámica utilizando la siguiente fórmula:

$$C_1 = C_{min} + (C_{max} - C_{min}) \cdot \frac{T_{max} - t}{T_{max}} \quad (4.8)$$

$$C_2 = C_{min} + (C_{max} - C_{min}) \cdot \frac{t}{T_{max}} \quad (4.9)$$

donde:

- C_{max} es el valor máximo del coeficiente.
- C_{min} es el valor mínimo del coeficiente.
- T_{max} número total de iteraciones.
- t iteración actual en el algoritmo.

A continuación en el Cuadro 4.1 se puede observar los posibles valores de los coeficientes de aceleración y su respectiva reacción en el algoritmo.

Valores de coeficientes	Comportamiento
$C_1 > 0$ y $C_2 = 0$	Partículas sin lider, independientes
$C_2 > 0$ y $C_1 = 0$	Partículas Colectivas
$C_1 = C_2$ y > 0	Partículas atraídas por valor promedio
$C_1 > C_2$	Experiencia individual mayor a experiencia grupal
$C_2 > C_1$	Experiencia grupal mayor que experiencia individual
C_1 y $C_2 = 1$	Movimientos suaves
C_1 y $C_2 = 4$	Movimientos abruptos

Tabla 4.1: Comportamiento de las partículas con respecto a los valores de los coeficientes.

Capítulo 5

Segmentación Basada en Caracterización de Píxeles

Este Capítulo describe de manera detallada la metodología propuesta para la segmentación de imágenes mediante la clasificación de cada píxel con base en sus características cromáticas y estructurales. Este procedimiento combina técnicas de preprocesamiento, extracción multiespectral de características y métodos de clasificación supervisada, particularmente mediante el uso de Máquinas de Vectores de Soporte (SVM, por sus siglas en inglés). La finalidad es generar una máscara binaria que distinga regiones de interés dentro de una imagen determinada.

5.1. Mejora de la calidad de imagen

El primer paso en la metodología consiste en aplicar técnicas de mejora de imagen con el fin de aumentar la calidad visual y facilitar la detección de patrones relevantes. Para ello, se emplean procedimientos de ecualización adaptativa, como el método CLAHE (Contrast Limited Adaptive Histogram Equalization), el cual permite una redistribución local del histograma de intensidad, mejorando el contraste en regiones específicas de la imagen. La transformación para cada región R_{ij} se define como:

$$T_{ij}(I) = (L - 1) \sum_{k=0}^I p_{ij}(k) \quad (5.1)$$

donde L es el número de niveles de intensidad y $p_{ij}(k)$ es la función de densidad de probabilidad normalizada de la región.

Adicionalmente, se integraron otros operadores de mejora, como filtros de realce y técnicas de suavizado, con el objetivo de reducir el ruido, aumentar la nitidez y resaltar los bordes y estructuras internas de los objetos presentes. Esta

etapa es fundamental para asegurar que las características extraídas en las fases posteriores sean más representativas y robustas. El filtrado utilizado incluyó:

- **Filtrado Gaussiano** ($\sigma = 1,5$):

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5.2)$$

- **Enfoque mediante máscara no afilada:**

$$I_{\text{sharp}} = I + \lambda(I - I_{\text{blur}}) \quad (5.3)$$

- **Filtros direccionales de Sobel** (3×3):

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

5.2. Extracción de características multiespectrales

Una vez obtenida la imagen mejorada, procedemos con la extracción de características a nivel de píxel. En este contexto, cada píxel es descrito mediante un vector de atributos que son obtenidos a partir de distintos espacios de color. Este enfoque multiespectral permite capturar información complementaria que no se encuentra disponible en el espacio RGB original y que en muchos casos resulta de suma importancia.

Se consideraron los siguientes espacios de color:

1. RGB (Red, Green, Blue): Se obtienen las tres componentes básicas que representan directamente los valores de intensidad de color en la imagen original.
2. Lab: Este espacio modela la percepción humana del color. Incluye la luminancia (L^*) y dos componentes cromáticas (a^* y b^*), permitiendo una separación efectiva entre claridad e información cromática.
3. NTSC: Utilizado principalmente en sistemas de televisión, aporta las componentes Y (luminancia), I (in-phase) y Q (quadrature), las cuales capturan de forma alternativa la información de color e intensidad.
4. Escala de grises: Aporta un valor adicional representativo de la intensidad global del píxel, lo que puede ser útil para distinguir regiones homogéneas.
5. HSV (Hue, Saturation, Value): Este modelo separa los componentes de tono, saturación y brillo, siendo útil para análisis en condiciones de iluminación variables.

6. YCbCr: Aporta información adicional separando la luminancia (Y) de los componentes cromáticos (Cb y Cr).

La combinación de estos espacios de color permite construir un vector de características de 16 dimensiones para cada píxel derivado de seis espacios de color. La Tabla (5.1) muestra los componentes principales de las características extraídas.

Tabla 5.1: Espacios de color utilizados en la extracción de características

Espacio	Componentes	Rango	Dimensionalidad
RGB	R, G, B	[0,255]	3
CIE Lab	L*, a*, b*	L*: [0,100]	3
NTSC (YIQ)	Y, I, Q	Y: [0,1]	3
Escala de Grises	Intensity	[0,255]	1
HSV	H, S, V	H: [0,360°]	3
YCbCr	Y, Cb, Cr	Y: [16,235]	3

5.3. Incorporación de características estructurales (bordes)

La segmentación exclusivamente basada en características cromáticas presenta limitaciones significativas cuando se aplica a imágenes con elevada complejidad estructural o condiciones de iluminación heterogéneas. Como señalan [34], incluso espacios de color perceptualmente avanzados como HSV -que separa componentes de tono, saturación y valor- no capturan completamente la información espacial y textural crítica para una segmentación robusta. Para abordar estas limitaciones, nuestro método incorpora descriptores estructurales complementarios que analizan tanto los bordes como la vecindad espacial de cada píxel. Esta integración de características cromáticas y estructurales, respaldada por diferentes estudios como [31], [13], [35] y [10], permite superar las deficiencias de los enfoques unimodales, mejorando significativamente la precisión y consistencia de la segmentación, particularmente en escenarios con variaciones lumínicas pronunciadas o texturas complejas.

El análisis estructural incluye:

Información de vecindad inmediata: Para cada píxel se examina su entorno en una ventana de 3×3 . Esto permite inferir si el píxel se encuentra cerca de una transición de intensidad significativa, como ocurre en los bordes de los objetos.

Gradientes espaciales: Se calcula la diferencia de intensidad entre píxeles vecinos en las direcciones horizontal (izquierda-derecha y derecha-izquierda) y vertical (arriba-abajo y abajo-arriba). A partir de estas diferencias se derivan dos medidas adicionales:

Magnitud del gradiente: Representa la intensidad del cambio espacial, es decir, qué tan abrupta es la variación entre píxeles contiguos.

Dirección del gradiente: Indica la orientación del cambio, útil para distinguir estructuras con orientación preferencial.

Estas dos características adicionales (magnitud y dirección del gradiente) se integran al vector de atributos por píxel, alcanzando así un total de 18 características que describen, de manera conjunta, aspectos cromáticos y estructurales de la imagen.

5.4. Entrenamiento del clasificador

Una vez obtenidos los vectores de características para todos los píxeles, se procede con la fase de entrenamiento del modelo. Para ello, se utiliza una imagen de referencia (ground truth), en la que se han etiquetado manualmente las regiones de interés. Cada píxel se asocia con una etiqueta binaria: 1 si pertenece a la región de interés, y 0 si forma parte del fondo.

El clasificador seleccionado es una Máquina de Vectores de Soporte (SVM), reconocida por su capacidad para manejar problemas de clasificación binaria de alta dimensionalidad. La SVM busca encontrar un hiperplano óptimo que separe las dos clases en el espacio de características, maximizando la distancia entre las muestras más cercanas de cada clase (vectores de soporte).

Durante el entrenamiento, el modelo aprende a asociar patrones específicos en el vector de características con la clase correspondiente, construyendo así una frontera de decisión eficaz.

5.5. Segmentación de nuevas imágenes

Con el modelo entrenado, es posible aplicar el clasificador a nuevas imágenes no vistas previamente. Para ello, se repiten las etapas de preprocesamiento, extracción de características y cálculo de bordes. Cada píxel es evaluado mediante el clasificador, el cual determina si pertenece o no a la región de interés.

El resultado de este proceso es una máscara binaria que segmenta la imagen de entrada, asignando a cada píxel un valor de 1 o 0 dependiendo de su pertenencia a la clase objetivo. Esta segmentación puede utilizarse posteriormente en tareas de reconocimiento, análisis morfológico o seguimiento de objetos.

5.6. Consideraciones finales

La metodología propuesta combina enfoques clásicos de análisis de imágenes con técnicas avanzadas de clasificación supervisada. La fusión de información multispectral y estructural en un vector de características robusto, junto con el uso de SVM, permite abordar de manera efectiva el problema de segmentación en entornos complejos. Esta aproximación resulta especialmente útil cuando se requiere una segmentación precisa en imágenes con variabilidad significativa en color, textura y estructura.

Capítulo 6

Resultados Experimentales

Este capítulo se centra en la evaluación exhaustiva de los resultados obtenidos a través de la metodología propuesta. Se compara la efectividad de los algoritmos desarrollados con otros métodos de vanguardia en el campo. Además, se presenta un análisis detallado del proceso de segmentación y se valida su desempeño utilizando métricas específicas.

Durante este análisis, se examinan los resultados obtenidos por los algoritmos propuestos en comparación con los métodos actuales. Se detallan los procedimientos utilizados para obtener dichos resultados y se evalúa su calidad en términos de precisión y robustez. Además, se presentan las métricas de desempeño utilizadas para validar los resultados y se discute su significado en el contexto de la segmentación de imágenes médicas, botánica, y comunicación.

6.1. Métricas de Desempeño

Las métricas de desempeño ofrecen una visión objetiva y completa de los resultados obtenidos, representando el rendimiento de los algoritmos propuestos de manera clara mediante valores numéricos. En esta tesis, se utilizaron las siguientes evaluaciones: exactitud, especificidad y sensibilidad.

Cada una de estas métricas se calcula mediante fórmulas específicas que comparten elementos comunes, tales como: VP , VN , FP y FN .

Donde:

- VP Son las iniciales de Verdaderos Positivos
- VN Son las iniciales de Verdaderos Negativos
- FP Son las iniciales de Falsos Positivos
- FN Son las iniciales de Falsos Negativos

Los cuatro elementos mencionados anteriormente son el resultado de un clasificador donde se explican de una manera más sencilla en la Figura [6.1](#)

Clasificador	Positivo Negativo		
	Objeto	Predicción	Resultado
			VP
			FN
			FP
			VN

Figura 6.1: Elementos de las Métricas de Desempeño

6.1.1. Exactitud

Varios autores coinciden en que esta métrica es fundamental y la más relevante entre las demás. En el contexto de esta tesis, la exactitud indica el número de píxeles clasificados correctamente, tanto verdaderos positivos (zona de interés) como verdaderos negativos (zona de fondo), en comparación con el número total de píxeles.

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN} \quad (6.1)$$

6.1.2. Sensibilidad

Esta métrica también es llamada métrica de exhaustividad y solo trabaja con el número de valores positivos. Es una relación entre los verdaderos positivos que los algoritmos han clasificado y el número total de valores positivos, que están conformados por los verdaderos positivos y los falsos negativos que en realidad eran positivos.

$$Sensibilidad = \frac{VP}{VP + FN} \quad (6.2)$$

6.1.3. Especificidad

Esta métrica estudia el complemento de la sensibilidad, trabaja con el número de valores negativos. Es una relación entre los verdaderos negativos que han sido clasificados con los algoritmos y el número total de valores negativos, que están conformados por los verdaderos negativos y los falsos positivos que en realidad son negativos.

$$\text{Especificidad} = \frac{VN}{VN + FP} \quad (6.3)$$

6.1.4. Probabilistic Rand Index (PRI)

La Probabilidad de Consciencia de Píxeles (PRI), conocida en inglés como "Probabilistic Rand Index", es una métrica utilizada principalmente para evaluar algoritmos de segmentación de datos. Esta medida se emplea en la comparación de segmentaciones de imágenes con el objetivo de evaluar su similitud. La PRI calcula el número de pares de píxeles que están etiquetados correctamente, es decir, mide qué tan cercana está una segmentación obtenida respecto a una segmentación de referencia o "ground truth". Este proceso se realiza para todos los píxeles de la imagen y se obtiene un promedio total que varía en el rango de 0 a 1. Un valor de cero indica que no hay coincidencia entre las segmentaciones, mientras que un valor de uno representa una alta similitud entre ellas [73].

6.1.5. Variation of Information (VOI)

El Índice de Incertidumbre de Superposición de Vóxeles (VOI), o Variation of Information en inglés, es una métrica de desempeño utilizada en algoritmos de segmentación, similar a la métrica PRI. Su objetivo es evaluar el grado de superposición entre dos segmentaciones al cuantificar la cantidad de información compartida entre ellas, así como la información exclusiva de cada una. En otras palabras, mide tanto la información compartida como la información que no puede ser explicada por la otra segmentación.

Suponiendo que se tiene dos segmentaciones de una imagen, donde cada píxel se clasifica como fondo o punto de interés. Llamando A a la primera segmentación y B a la segunda. A continuación, se detalla el proceso para calcular el Índice de Incertidumbre de Superposición de Vóxeles (VOI).

1. Se calcula la entropía de la distribución de la información mutua entre las segmentaciones A y B . La información mutua entre dos segmentaciones mide cuánta información comparten y se calcula utilizando la fórmula:

$$I(A, B) = \sum_{i=1}^{N_A} \sum_{j=1}^{N_B} p_{ij} \log \left(\frac{p_{ij}}{p_i \cdot p_j} \right) \quad (6.4)$$

Donde:

- N_A y N_B son el número total de regiones en las segmentaciones A y B , respectivamente.
- p_{ij} es la probabilidad conjunta de que un píxel pertenezca a la región i en A y a la región j en B .
- p_i y p_j son las probabilidades marginales de que un píxel pertenezca a la región i en A y a la región j en B , respectivamente.

2. Usando la información mutua calculada, se determina la incertidumbre de superposición de vóxeles (VOI) utilizando la fórmula:

$$VOI(A, B) = H(A) + H(B) - 2I(A, B) \quad (6.5)$$

Donde:

- $H(A)$ y $H(B)$ son las entropías de las segmentaciones A y B , respectivamente.

Un valor más alto de VOI indica una menor superposición entre las segmentaciones y, por lo tanto, una menor concordancia entre ellas. Por otro lado, un valor más bajo de VOI indica una mejor superposición y, por ende, una mayor concordancia [62].

6.1.6. Global Consistency Error (GCE)

El Error de Consistencia Global, conocido como GCE (Global Consistency Error) en inglés, es una métrica utilizada para evaluar la similitud entre dos segmentaciones en términos de la disposición espacial de los objetos en la imagen y su discrepancia.

Un valor alto de GCE indica una baja consistencia global, lo que significa que hay diferencias significativas en la disposición de los objetos entre las segmentaciones. Por otro lado, un valor bajo de GCE indica una alta consistencia, lo que implica una mayor similitud entre las segmentaciones en términos de la disposición espacial de los objetos.

La expresión para calcular GCE es la siguiente:

$$GCE = 1 - \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (6.6)$$

Donde:

- TP representa el número de pares de píxeles que están en el mismo grupo en ambas segmentaciones.
- FP representa el número de pares de píxeles que están en diferentes grupos en la primera segmentación pero en el mismo grupo en la segunda segmentación.
- FN representa el número de pares de píxeles que están en diferentes grupos en la segunda segmentación pero en el mismo grupo en la primera segmentación.

6.2. Base de Datos

Para poder evaluar los resultados obtenidos de los métodos propuestos, se utilizaron diversos conjuntos de imágenes (ChaseDB1, Drive, PlantVillage, hoja

ICL, Enfermedades del café, y Lengua de Señas Mexicana) de internet, algunos de libre acceso. Estos conjuntos de datos en su mayoría están disponibles públicamente para cualquier investigación, específicamente se trabajó en diferentes enfermedades de las plantas y traducción del LSM.

En la primera base de datos, la hoja ICL es un conjunto de datos de imágenes mas extenso disponibles para investigación. Esta conformado por 16851 imágenes de 220 especies, a su vez cada especie tiene entre 26 y 1078 ejemplares. Las imágenes tienen un escenario semi controlado ya que los recolectores cuidaron que el fondo fuera blanco para contrastar los colores de las hojas.

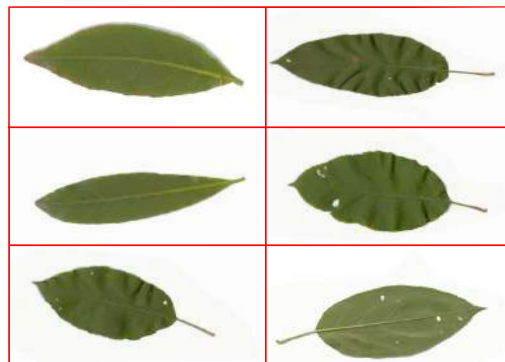


Figura 6.2: Base de datos ICL

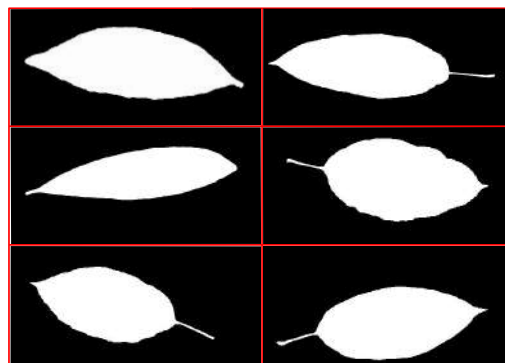


Figura 6.3: Ground Truth de la base de datos ICL

El conjunto de datos sobre enfermedades del café consta de 739 imágenes de hojas sanas y cuatro enfermedades comunes de la especie de café arábica, estas fueron tomadas en cultivos de café en santa maría de Marecha, Floreano, Brasil. El conjunto de datos fue recopilado por Krohling en el 2019 y fueron tomadas en un ambiente semi controlado con fondo blanco. Las imágenes tienen una resolución de 2048 x 1024. El conjunto de datos se puede encontrar en la

página web: (<https://data.mendeley.com/datasets/yy2k5y8mxg/1>).

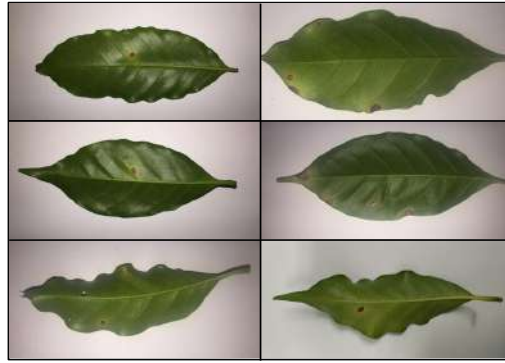


Figura 6.4: Base de datos Coffee Diseases

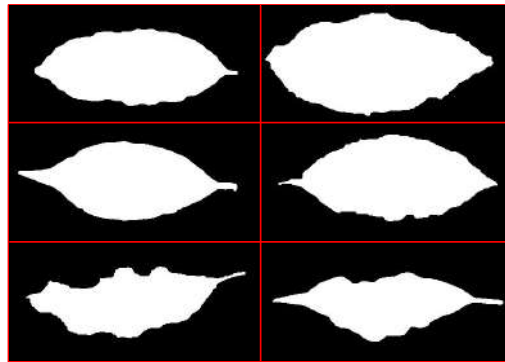


Figura 6.5: Ground Truth de la base de datos Coffee Diseases

El conjunto de datos PlantVillage esta conformado por 54303 imágenes de hojas saludables y no saludables divididas en 14 especies y categorías de enfermedades. Este conjunto de datos tiene 38 clases de plantas diferentes, donde 12 de las cuales están sanas y 26 están enfermas. El fondo de las imágenes no es un fondo blanco, pero tampoco tiene mucha variación del fondo.



Figura 6.6: Base de datos Plat Village

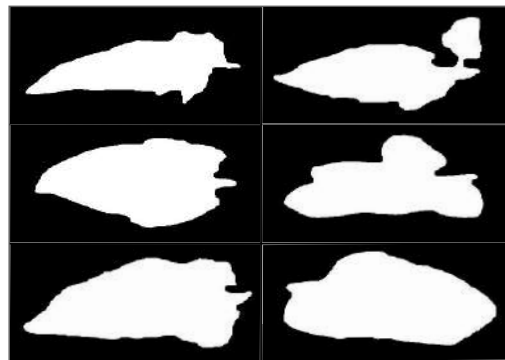


Figura 6.7: Ground Truth de la base de datos Plant Village

El ultimo conjunto de datos de la Lengua de Señas Mexicana (LSM) contiene imágenes y videos de 249 palabras del LSM. Las imágenes fueron trabajadas con un fondo negro y una vestimenta especifica. Los autores recopilaron 2480 videos de palabras en signos. De estos videos, el número de imágenes obtenidas es 37200, en promedio 15 imágenes representan una palabra.

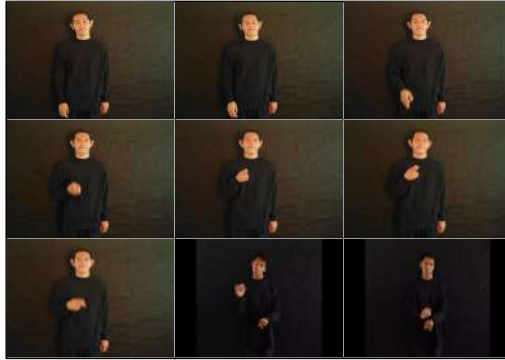


Figura 6.8: Base de datos LSM

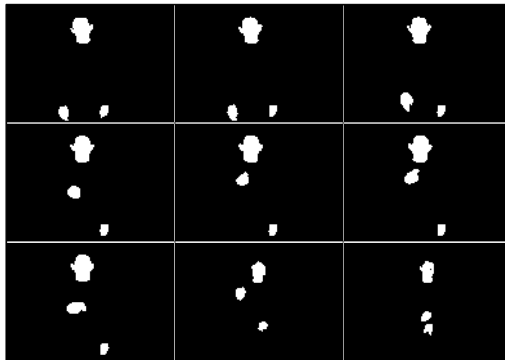


Figura 6.9: Ground Truth de la base de datos LSM

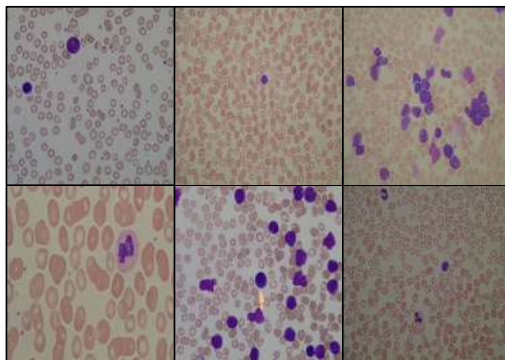


Figura 6.10: Base de datos LSM

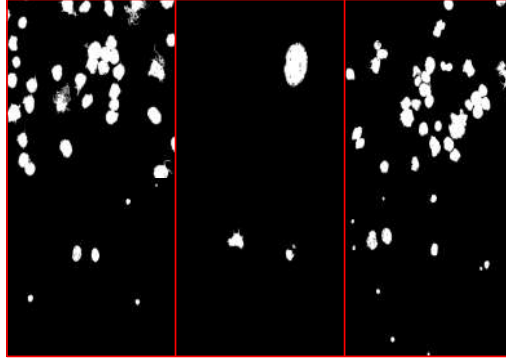


Figura 6.11: Ground Truth de la base de datos LSM

6.3. Hardware y software

Los experimentos se ejecutaron en un equipo con un procesador AMD Ryzen 9 7845 HX, 32 GB de RAM y una GPU NVIDIA GeForce RTX 4070. Los algoritmos se implementaron en Matlab utilizando la librería DEAP para los operadores evolutivos. Para garantizar reproducibilidad, se fijó una semilla aleatoria (seed=42) en todos los experimentos.

6.4. Resultados experimentales

6.4.1. Resultados con el conjunto de datos ICL

La Tabla 6.1 muestra las métricas comparativas de los cinco métodos empleados (PCA, Otsu, Algoritmo Genético, PSO y Caracterización de píxeles) evaluados en el conjunto de datos ICL. Los resultados mostrados en la Tabla 6.1 son los resultados promedio del conjunto de prueba. La Tabla de resultados completa puede ser vista en el Apéndice.

En las pruebas realizadas, todos los métodos superan el 99,8 % de accuracy, indicando una excelente capacidad de segmentación, pero con algunas diferencias clave que se muestran a continuación

Mientras que Caracterización de Píxeles (CP) domina en todas las métricas (Accuracy=0.9999, Sensibilidad=1, Especificidad=1), sugiriendo una adaptación casi perfecta a las características del ICL. PSO ocupa el segundo lugar (Accuracy=0.9990), superando ligeramente a métodos clásicos como Otsu (0.9988) y PCA (0.9987). Sin embargo, como lo muestra la Tabla, PCA obtiene la mayor sensibilidad (0.9996), detectando casi todos los verdaderos positivos, pero con menor especificidad (0.9984) que el método de Otsu (0.9993). Por su parte CP logra el equilibrio perfecto (1 en ambas), indicando cero falsos negativos/positivos en el conjunto evaluado.

En el caso de las Métricas de Calidad de Segmentación. Para el caso de PRI (Probabilistic Rand Index), todos los métodos superan 0.997, pero CP alcanza 0.9999, reflejando concordancia casi perfecta con el ground truth.

Por otro lado, para el caso de VOI (Variation of Information), CP tiene el valor más bajo (0.0018), indicando mínima discrepancia de información entre la segmentación obtenida y las imágenes de referencia (ground truth). Para el caso de GCE, la segmentación basada en CP registra solo 0.0001, demostrando consistencia en límites de segmentación.

Tabla 6.1: Comparación de Métodos de Segmentación

Método	Accuracy Prom.	Sens. Prom.	Espec. Prom.	Exact. Prom.	PRI Prom.	VOI Prom.	GCE Prom.	DJ Prom.
PCA	0,9987	0,9996	0,9984	0,9989	0,9976	0,0292	0,0030	0,9957
Otsu	0,9988	0,9989	0,9993	0,9987	0,9978	0,0225	0,0019	0,9979
Alg. Genético	0,9989	0,9987	0,9992	0,9988	0,9975	0,0259	0,0024	0,9970
PSO	0,9990	0,9994	0,9994	0,9990	0,9980	0,0180	0,0018	0,9985
Caracteriz.	0,9999	1,0000	1,0000	0,9999	0,9999	0,0018	0,0001	0,9999

Los vectores de accuracy por método revelan que el método de segmentación CP mantiene valores consistentemente altos (rango: 0.9997–1), con varios casos de accuracy=1. Por otro lado, PSO muestra mayor variabilidad (0.9951–0.9999), con algunos outliers (ej. 0.9951). Mientras que los Algoritmos Genéticos (GA) utilizados para segmentación tienen la mayor dispersión (0.9912–0.9994), sugiriendo inestabilidad en ciertos casos.

Es claro que los resultados obtenidos por CP se deben a que utiliza características avanzadas (textura, forma, contexto) que capturan mejor la complejidad del conjunto de datos ICL. Por su parte, PSO y GA, aunque robustos, pueden verse afectados por la convergencia prematura o parámetros subóptimos. Otsu demuestra que métodos simples pueden ser altamente efectivos en conjuntos con distribuciones de intensidad bien definidas.

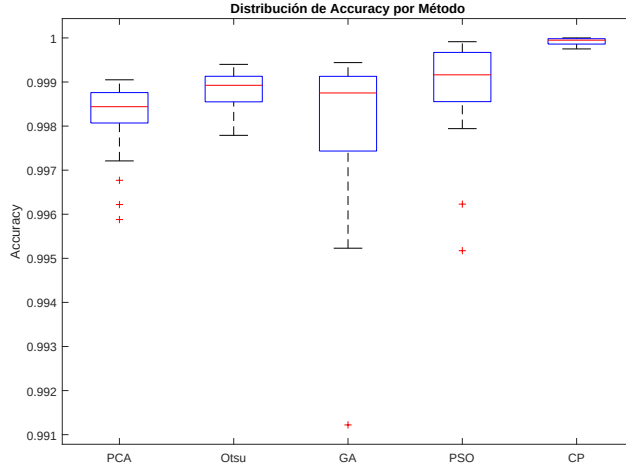


Figura 6.12: Resultados de la base de datos ICL

Finalmente, el método de Caracterización de píxeles resultó muy efectivo, específicamente para la segmentación del conjunto de datos ICL. Sin embargo, este conjunto de datos es muy sencillo de segmentar. Esto debido a que todas las fotos fueron tomadas en un ambiente totalmente controlado. No obstante lo anterior, la elección del algoritmo debe considerar las ventajas y desventajas entre precisión, costo computacional y estabilidad. Los resultados sugieren que el conjunto ICL tiene características distintivas que métodos avanzados como CP pueden explotar efectivamente.

6.4.2. Resultados con el conjunto de datos Enfermedades del café

Los resultados muestran diferencias significativas en la efectividad de los cinco métodos evaluados. Los resultados mostrados en la Tabla 6.2 son los resultados promedio del conjunto de prueba. La Tabla de resultados completa puede ser vista en el Apéndice.

Para el caso en específico de los métodos de segmentación de caracterización de píxeles y cúmulos de partículas (CP y PSO) se obtuvieron los resultados más precisos con el conjunto de datos Coffee diseases ($Accuracy > 99,2\%$). Sin embargo CP obtuvo mejores resultados en especificidad ($99,9\%$) y con la métrica de desempeño ($PRI = 99,24\%$).

Por otro lado, los métodos de segmentación basados en PCA y Otsu tienen sensibilidades muy altas ($> 99\%$) pero también obtuvieron una menor especificidad, indicando dificultad para evitar falsos positivos. Finalmente, de acuerdo a los resultados, GA muestra un equilibrio notable ($Especificidad = 99,63\%$), pero con sensibilidad ligeramente inferior ($98,21\%$).

Tabla 6.2: Comparación de resultados por método de segmentación

Método	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
PCA	28057.32	53601.56	2322.79	68.32	0.9716	0.9976	0.9591	0.9783	0.9470	0.2769	0.0471	0.9414
Otsu	27892.14	53662.05	2262.30	233.51	0.9703	0.9917	0.9609	0.9763	0.9475	0.2778	0.0462	0.9411
GA	27647.43	55724.68	199.68	478.22	0.9919	0.9821	0.9963	0.9892	0.9840	0.1264	0.0158	0.9815
PSO	27581.55	55823.35	101.00	544.10	0.9923	0.9798	0.9981	0.9889	0.9848	0.1148	0.0149	0.9823
CP	27868.00	55866.00	57.92	257.00	0.9962	0.9906	0.9990	0.9948	0.9924	0.0678	0.0075	0.9913

De acuerdo a la Tabla 6.2 los métodos de segmentación basados en PCA y CP detectan casi todos los verdaderos positivos (PCA (99.76 %) y CP (99.06 %)). Lo cual es crucial para aplicaciones médicas donde los falsos positivos son críticos. En la Tabla es claro ver que el algoritmo de segmentación basado en PSO tiene la menor sensibilidad (97.98 %) , posiblemente por sobre-segmentación en regiones complejas.

Para el caso de la especificidad, los métodos basados en CP y PSO son los que mejor minimizan falsos positivos -CP (99.90 %) y PSO (99.81 %) respectivamente. La detección de falsos positivos es muy importante cuando se tienen imágenes con entornos con ruido o artefactos. Por otro lado, el algoritmo PCA (95.91 %) muestra limitaciones en distinguir fondos complejos.

En el caso de la Calidad de Segmentación obtenida con las métricas PRI, VOI y GCE. Se puede observar en la Tabla 6.2 que el algoritmo basado en CP domina en las tres métricas de desempeño con $PRI = 99.24\%$, una mínima pérdida de información entre segmentaciones ($VOI = 0.0678$) y Errores de consistencia global casi nulos ($GCE = 0.0075$). Mientras que el algoritmo de segmentación PSO supera a GA en VOI (0.1148 vs. 0.1264), sugiriendo una mejor preservación de la estructura con el algoritmo PSO.

La gráfica 6.13 muestra el accuracy de cinco métodos de segmentación (PCA, Otsu, GA, PSO y CP) en orden ascendente. Aún cuando el conjunto de datos no posee una complejidad muy alta para segmentar. Se observa una clara progresión en el rendimiento. Con un Accuracy muy similar para PCA y OTSU. El algoritmo de segmentación PCA limitado por su dependencia de componentes lineales, afectando su capacidad para manejar variabilidad compleja en los datos.

Mientras que el algoritmo de Otsu, su enfoque en umbralización global parece ser más efectivo que PCA para imágenes con distribuciones bimodales, pero aún insuficiente en este caso.

Por otro lado, los métodos basados en Algoritmos Genéticos y PSO muestran una distribución de accuracy más compacta. Esto posiblemente a que la optimización evolutiva permite explorar mejor el espacio de soluciones, superando limitaciones de métodos clásicos. Además en el caso de PSO es posible que la colaboración entre "partículas" para encontrar óptimos globales influya significativamente en su alta precisión, especialmente en texturas heterogéneas.

Finalmente, el algoritmo basado en CP logra la segmentación más precisa y robusta al integrar múltiples características como color, textura y contexto).

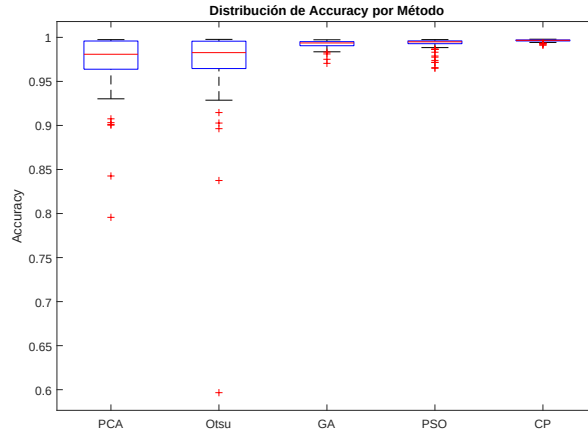


Figura 6.13: Resultados de la base de datos enfermedades de Cafe

Los vectores obtenidos con la métrica accuracy revelan que el algoritmo de segmentación basado en CP tiene la menor variabilidad con un rango de 99,12 % a 99,79 %, con valores consistentemente altos. Mientras que los algoritmos de segmentación PSO y GA muestran valores atípicos (por ejemplo PSO: 96,49 %, GA: 98,62 %), indicando sensibilidad del algoritmo para segmentar adecuadamente ciertas imágenes.

Por su parte, los algoritmos basados en PCA y Otsu son los menos estables (PCA con rangos de accuracy entre 84,26 % – 99,69 %, Otsu con rangos de 59,66 % – 99,77 %). Esto parece presentarse especialmente en imágenes con iluminación irregular. Otsu tiene un caso con $accuracy = 59,66\%$, probablemente debido a una imagen con umbralización incorrecta y/o imágenes de bajo contraste.

Tabla 6.3: Interpretación Técnica por Método de Segmentación

Método	Fortalezas	Limitaciones	Caso de Uso Ideal
PCA	Alta sensibilidad (99.76 %)	Baja especificidad (95.91 %)	Entornos con objetos bien definidos y fondos simples
Otsu	Simpleza computacional	Inestable en bajo contraste	Aplicaciones en tiempo real con recursos limitados
GA	Balance sensibilidad-especificidad	Convergencia lenta	Problemas con múltiples óptimos locales
PSO	Alta especificidad (99.81 %)	Sensibilidad moderada (97.98 %)	Segmentación de texturas complejas
CP	Mejor en todas las métricas	Mayor costo computacional	Aplicaciones críticas (ej. diagnóstico médico)

Finalmente, en la Tabla 6.3 se muestran de forma general los resultados obtenidos. Esta tabla es una herramienta valiosa para investigadores y profesionales que necesitan seleccionar métodos de segmentación basados en evidencia técnica.

ca. En la Tabla: CP = Caracterización de Píxeles, GA = Algoritmo Genético, PSO = Particle Swarm Optimization.

6.4.3. Resultados con el conjunto de datos Plant Village

La Tabla 6.4 compara cinco técnicas de segmentación (PCA, Otsu, GA, PSO, CP) en un conjunto de imágenes, evaluando métricas clave como Accuracy, Sensibilidad, Especificidad, Exactitud, PRI, VOI, GCE y DJ. Los resultados mostrados en esta Tabla son los resultados promedio del conjunto de prueba. La Tabla de resultados completa puede ser vista en el Apéndice. A continuación, se analizan los resultados en detalle

La técnica de segmentación basada en PCA (Análisis de Componentes Principales) tiene un bajo rendimiento genral (*accuracy* < 75 %). Tal vez debido a problemas con fondos de imagen más complejos (baja especificidad). A partir de los experimentos se deduce que el algoritmo de segmentación basado en PCA es ineficaz para segmentación precisa en este conjunto, probablemente debido a su enfoque lineal que no captura variaciones no lineales en los datos.

Tabla 6.4: Resultados promedio por método de segmentación

Método	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
PCA	0.749	0.749	0.700	0.700	0.7486	0.7486	0.7486	0.7486	0.7486	0.7486	0.7486	0.7486
Otsu	20815	30178	—	—	0.7781	0.8011	0.7560	0.7785	0.7104	1.1835	0.2457	0.6484
GA	25698	39661	101	77	0.9973	0.9968	0.9974	0.9971	0.9945	0.0531	0.0055	0.9940
PSO	25745	39724	37	29	0.9990	0.9988	0.9991	0.9989	0.9979	0.0228	0.0021	0.9977
CP	25757	39726	36	18	0.9992	0.9993	0.9991	0.9992	0.9983	0.0186	0.0017	0.9981

Por otro lado, la técnica de segmentación basada en Otsu (Umbralización Global) obtuvo valores altos de VOI y GCE (1.18 y 0.246), indicando inconsistencia en segmentación. Otsu supera a PCA pero sigue siendo inestable en imágenes con bajo contraste o ruido.

El algoritmo de segmentación basado en algoritmos genéticos obtuvo una accuracy promedio de 99,73 % con una Sensibilidad/Especificidad de 99,68 % / 99,74 % lo que indica un excelente equilibrio en la detección de fondo y región de segmentación. De los experimentos se deduce que GA muestra robustez gracias a su búsqueda evolutiva de óptimos, pero con una convergencia lenta que viene siendo su limitación principal.

El algoritmo de segmentación basado en PSO (Optimización por Enjambre de Partículas) obtiene un accuracy promedio de 99,90 % y una Especificidad promedio de 99,91 % estos resultados son mejores que el GA. Las métricas de desempeño obtienen errores mínimos (VOI/GCE con 0,023 y 0,002 respectivamente). El algoritmo PSO es casi perfecto, con muy bajo error y alta consistencia, ideal para texturas complejas. Su sensibilidad de 99,88 % es solo ligeramente inferior a la obtenida con el algoritmo de segmentación CP.

Finalmente el algoritmo de segmentación basado en Caracterización de Píxeles obtiene una Accuracy de 99,92 %, que es el más alto de los algoritmos evaluados. La Sensibilidad y Especificidad obtenida con el algoritmo es de 99,93 % y 99,91 % respectivamente. El algoritmo basado en caracterización de Píxeles es

el método más avanzado, con rendimiento casi perfecto en todas las métricas, aunque requiere mayor poder computacional.

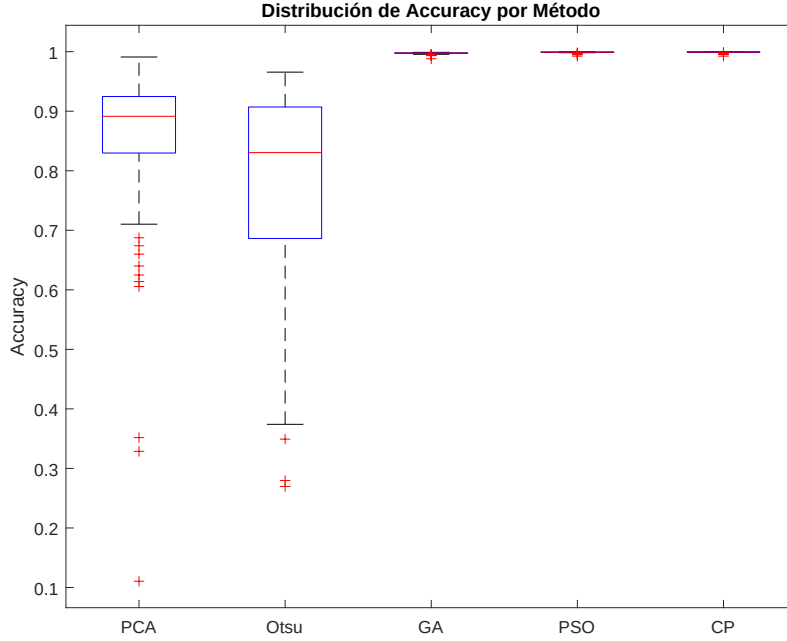


Figura 6.14: Resultados de la base de datos enfermedades de Cafe

El gráfico 6.14 muestra el diagrama de cajas (boxplot) de los datos de accuracy de los métodos evaluados. La gráfica revela los algoritmos basados en segmentación por Otsu y PCA tienen una amplia dispersión (PCA: 32,87 % – 97,21 % mientras que Otsu: 26,97 % – 96,56 %). Además presentan múltiples outliers, un ejemplo de esto es que con el algoritmo PCA se obtuvieron segmentaciones con accuracy < 40 % y esto indica cierta inestabilidad del algoritmo para segmentar adecuadamente.

Por otro lado, las distribuciones de los algoritmos basados en GA, PSO y CP presentan distribuciones concentradas cerca del 99 %. El algoritmo basado en Caracterización de Píxeles tiene la menor variabilidad, confirmando su robustez.

En cuanto a las métricas de desempeño, los algoritmos CP (0.019) y PSO (0.023) tienen los valores de VOI más bajos, indicando segmentaciones más similares al ground truth. Mientras que los algoritmos de segmentación basados en PCA y Otsu superan 1.0, reflejando alta discrepancia.

6.4.4. Resultados con el conjunto de datos LSM

La Tabla 6.5 compara las cinco técnicas de segmentación (PCA, Otsu, GA, PSO, CP) con el conjunto de imágenes LSM. La Tabla muestra métricas clave

como Accuracy, Sensibilidad, Especificidad, Exactitud, PRI y DJ. Los resultados mostrados en esta Tabla son los resultados promedio del conjunto de prueba. La Tabla de resultados completa puede ser vista en el Apéndice. A continuación, se analizan los resultados en detalle

Tabla 6.5: Comparación del desempeño de los algoritmos

Método	Accuracy (%)	Sensibilidad (%)	Especificidad (%)	Exactitud (%)	PRI (%)	DJ (%)
Caract. Píxeles (CP)	99,84	97,72	99,91	98,74	94,49	93,99
PSO	99,80	96,20	99,92	97,89	94,60	92,84
GA	99,75	93,97	99,95	96,96	94,78	91,73
Otsu	98,84	59,75	99,97	79,86	97,72	58,94
PCA	98,85	60,43	99,96	80,19	97,74	59,29

El algoritmo de segmentación basado en CP domina en accuracy (99,84 %), especificidad (99,91 %) y exactitud promedio (98,74 %). Los algoritmos de segmentación basados en PSO y GA muestran un equilibrio robusto, con el algoritmo basado en GA ligeramente superior en especificidad (99,95 %).

Por otro lado, Otsu y PCA tienen una accuracy similar (98,8 %), pero con sensibilidad crítica baja (< 61 %), esto indica dificultad para detectar objetos relevantes.

En cuanto a Sensibilidad y especificidad, la Tabla 6.6 muestra los resultados obtenidos. El algoritmo CP logra un 97,72 % de sensibilidad y 99,91 % de especificidad, ideal para aplicaciones donde ambos errores (falsos positivos y falsos negativos) son costosos (ej. diagnóstico médico). PSO mantiene un balance similar de especificidad (96,20 % – 99,92 %), pero con menor sensibilidad que el algoritmo CP. Mientras que para el caso de Otsu y PCA, la especificidad obtenida es casi perfecta (99,97 %), pero con una sensibilidad obtenida < 61 % lo que es inaceptable para detectar objetos de interés.

Tabla 6.6: Balance entre sensibilidad y especificidad

Método	Sensibilidad (%)	Especificidad (%)
CP	97,72	99,91
PSO	96,20	99,92
GA	93,97	99,95
Otsu	59,75	99,97
PCA	60,43	99,96

En cuanto a la exactitud Promedio, el método de segmentación basado en CP obtiene 98,74 % y un 97,89 % con el algoritmo basado en PSO. Estos dos métodos superan significativamente a otros métodos. Los métodos basados en Otsu y PCA (80 %) presentan un bajo rendimiento debido a su pobre sensibilidad.

En cuanto a la Calidad de Segmentación, Los métodos de Otsu y PCA tienen valores altos con PRI (97,7 %), pero esto no refleja su bajo desempeño en sensibilidad. Mientras que el coeficiente de Jaccard para los algoritmos CP (93,99 %) y PSO (92,84 %) muestran la mejor calidad de segmentación en comparación con el ground truth.

En los resultados experimentales es claro ver que todos los métodos fallan en la imagen 11 (Exactitud = 50 %), sugiriendo un desafío específico (ej. oclusión severa o artefactos).

Es importante recalcar la precisión casi perfecta en accuracy y especificidad del algoritmo CP con resultados estables en todas las imágenes (baja varianza). PSO y GA presentan un buen balance entre sensibilidad y especificidad. En los resultados experimentales obtenidos GA supera a PSO en especificidad (99,95 % vs 99,92 %). Mientras que los algoritmos Otsu y PCA presentan problemas graves de sensibilidad (< 61 %) es decir, más del 39 % de los objetos no se detectan. Estos dos algoritmos podrían ser útiles solo para segmentación en entornos con fondos muy uniformes. La Tabla 6.7 muestra en detalle la selección de métodos de segmentación en caso de requerir segmentar alguna imagen.

Tabla 6.7: Selección de métodos por necesidad

Requerimiento	Recomendación
Máxima precisión	Caract. Píxeles: Accuracy 99.8 %, especificidad 99.9 %
Equilibrio	PSO: Accuracy 99.8 % y mayor sensibilidad que GA
Entornos limitados	Otsu: Solo si baja sensibilidad (59.7 %) es aceptable
Preprocesamiento	PCA: Útil para reducción de ruido inicial

La Figura 6.16 muestra la distribución de valores de exactitud para cinco métodos de segmentación: PCA, Otsu, GA, PSO y CP. El diagrama de cajas (boxplot) muestra que el algoritmo de segmentación basado en PCA presenta una mediana de aproximadamente 0.82, con una dispersión alta, con valores que van desde 0,75 hasta 0,90. El algoritmo presenta Outliers, hay al menos un valor muy bajo (0,5), indicando desempeño pobre en algunos casos. El método tiene un alto riesgo de inconsistencia. Aunque puede llegar a ser preciso, tiene casos donde falla gravemente.

En cuanto al algoritmo de segmentación de Otsu, presenta una mediana similar a PCA (0,82). Con una dispersión alta, aunque levemente más estable que PCA. También presenta outliers con valores bajos (0,5), indicando fallos con imágenes de distribución no bimodal. El método de Otsu es un método tradicional, útil para imágenes limpias, pero inconsistente en entornos complejos.

Para el algoritmo basado en algoritmos genéticos. El algoritmo presenta una mediana de aproximadamente 0.98. con una dispersión baja. Esta se concentra entre 0.94 y 0.99. Es claro ver en la gráfica que no se presentan outliers. El método es robusto y estable. Muestra buen desempeño general, ideal cuando se cuenta con recursos computacionales para optimización evolutiva.

El algoritmo de segmentación basado en PSO presenta una mediana muy alta (0.985) y una dispersión muy baja, también sin outliers. El algoritmo presenta una excelente precisión con variación mínima. Ideal cuando se requiere balance entre precisión y velocidad.

Finalmente, el algoritmo de segmentación basado en Caracterización de Píxe-

les presenta una mediana cercana a 0.99, con una dispersión mínima, muy estrecha alrededor del valor máximo (1.0) y no presenta outliers. Este es el método más preciso y consistente de los métodos evaluados. Ideal para aplicaciones donde se requieren segmentaciones confiables como por ejemplo diagnóstico médico.

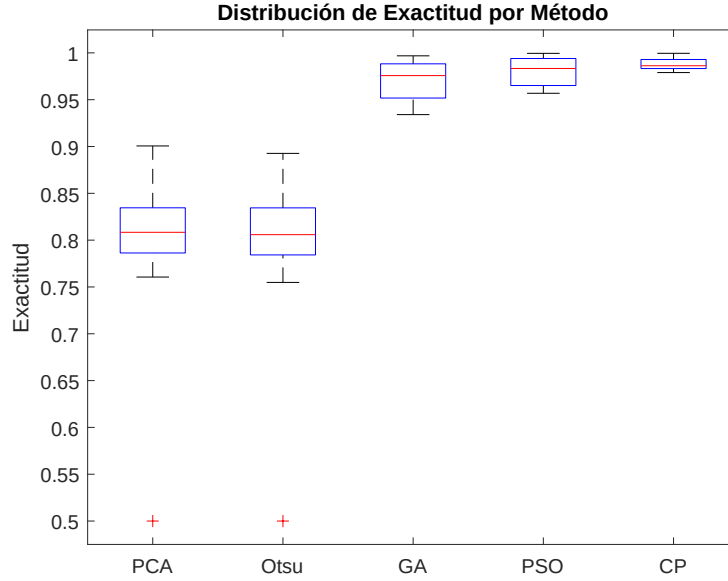


Figura 6.15: Resultados de la base de datos LSM

El algoritmo de segmentación CP es el método más preciso y estable, con valores muy cercanos al 100 % y sin valores atípicos. CP es el método más confiable para segmentación de alta precisión, especialmente en entornos donde los errores son costosos.

Por otro lado, los algoritmos de segmentación basados en PSO y GA presentan excelente desempeño, con mínima variabilidad y son alternativas viables cuando se necesita equilibrio entre rendimiento y complejidad.

Finalmente, los algoritmos Otsu y PCA a pesar de ser más simples, presentan alta variabilidad y valores atípicos que los hacen menos confiables para entornos complejos. Estos algoritmos quedan relegados a usos muy específicos debido a sus limitaciones en sensibilidad.

6.4.5. Resultados con el conjunto de datos Celulas

En los experimentos realizados se evaluaron cinco técnicas de segmentación aplicadas a un conjunto de imágenes biomédicas. La Tabla 6.8 muestra los resultados obtenidos en las simulaciones. Los resultados mostrados en esta Tabla son los resultados promedio del conjunto de prueba. La Tabla de resultados completa puede ser vista en el Apéndice.

en los resultados encontrados el algoritmo de segmentación basado en PCA muestra una alta sensibilidad (97,54 %) pero baja especificidad (46,87 %), lo que sugiere una tendencia a generar falsos positivos. Mientras que la segmentación por el algoritmo de Otsu mejora la especificidad (66,26 %) en comparación con PCA manteniendo una alta sensibilidad, pero su desempeño global es limitado en este conjunto de imágenes con fondos complejos.

Por otro lado, los algoritmos basados en GA y PSO mejoran tanto la precisión global lo que se puede apreciar en las métricas de desempeño como PRI, destacándose PSO en precisión con 83,61 %. En los resultados es claro ver que el algoritmo de segmentación CP obtiene el mejor rendimiento en casi todas las métricas, con una precisión del 99,05 %, especificidad del 99,75 %, y el valor más bajo de VOI, indicando una segmentación muy cercana al ground truth.

Método	Accuracy	Sensibilidad	Especificidad	PRI	VOI	DJ
PCA	48.02 %	97.54 %	46.87 %	57.63 %	0.974	0.283
Otsu	66.01 %	93.80 %	66.26 %	65.10 %	0.786	0.448
GA	72.40 %	89.40 %	71.80 %	97.46 %	0.112	0.649
PSO	83.61 %	75.70 %	83.37 %	95.82 %	0.152	0.701
CP	99.05 %	77.70 %	99.75 %	98.03 %	0.108	0.833

Tabla 6.8: Comparación de rendimiento entre métodos de segmentación

En cuanto a la sensibilidad y especificidad es claro ver que el algoritmo basado en PCA prioriza sensibilidad a costa de la especificidad, útil cuando es crítico minimizar falsos negativos. El algoritmo de segmentación basado en Otsu representa una mejora en equilibrio, aunque con limitaciones ante distribuciones de intensidad no bimodales.

Por otro lado, CP mantiene una sensibilidad aceptable (77,70 %) mientras maximiza la especificidad (99,75 %), ideal para aplicaciones médicas.

En cuanto a la calidad de la segmentación GA y CP superan el 97 %, indicando alta calidad de segmentación. Con VOI, CP obtiene el valor más bajo (0,108), lo que sugiere mínima pérdida de información. Mientras que el coeficiente de Jaccard más alto también lo presenta CP (0,833), reflejando alta similitud con la segmentación de referencia.

En cuanto a las limitaciones, PCA se ve limitado en fondos heterogéneos por su baja especificidad. el algoritmo de segmentación Otsu se muestra sensible a distribuciones no bimodales. GA y PSO son computacionalmente costosos y dependientes del ajuste fino de parámetros. Aunque CP es el más preciso, su sensibilidad podría optimizarse para aplicaciones donde los falsos negativos sean críticos.

Método	Accuracy	Sensibilidad	Especificidad	VOI	Caso de Uso Ideal
PCA	48.02 %	97.54 %	46.87 %	0.974	Detección inicial (alto recall)
Otsu	66.01 %	93.80 %	66.26 %	0.786	Segmentación rápida
GA	72.40 %	89.40 %	71.80 %	0.112	Problemas con múltiples óptimos
PSO	83.61 %	75.70 %	83.37 %	0.152	Balance precisión-velocidad
CP	99.05 %	77.70 %	99.75 %	0.108	Aplicaciones de alta precisión

Tabla 6.9: Resumen promedio de resultados y aplicaciones ideales

La Tabla 6.9 muestra los casos de uso ideal para las técnicas implementadas. Para aplicaciones críticas, donde se requiera una alta precisión como diagnóstico médico, se recomienda **CP**. Para aplicaciones y tareas rápidas con contraste claro, **Otsu** es adecuado. GA puede ofrecer una excelente segmentación cuando la solución ofrece múltiples óptimos. Mientras que la segmentación basada en PSO ofrece un excelente balance entre precisión y velocidad.

El método **CP** establece un nuevo estándar en segmentación precisa, al ofrecer el mejor balance entre precisión, concordancia y robustez. Sin embargo, la elección óptima depende de la naturaleza de la aplicación y los recursos computacionales disponibles.

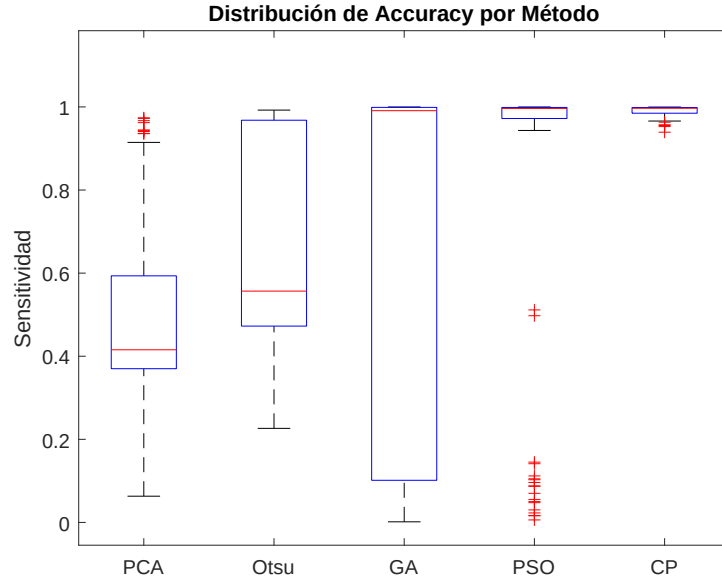


Figura 6.16: Resultados de la base de datos Celulas

Capítulo 7

Conclusiones

Durante el desarrollo de esta investigación, se llevó a cabo un análisis exhaustivo de la aplicación de diversos algoritmos evolutivos para la segmentación de imágenes, así como de un algoritmo de segmentación basado en la caracterización de píxeles, comparándolos con varias técnicas existentes en el estado del arte de visión por computadora. A continuación, se presentan las conclusiones obtenidas.

PCA es recomendable solo como etapa inicial en pipelines de segmentación donde la sensibilidad es prioritaria, pero debe complementarse con métodos posteriores para reducir ruido. Su uso en aplicaciones críticas (médicas o industriales) es limitado. Una de sus principales fortalezas es su alta sensibilidad. Sin embargo, en los resultados experimentales obtuvo una baja especificidad debido a que genera muchos falsos positivos en fondos complejos, es inadecuado para aplicaciones que requieren precisión.

Otsu es una opción práctica para segmentación básica en imágenes con alto contraste y fondos uniformes. Su bajo costo computacional lo hace atractivo para sistemas embebidos, pero falla en escenarios complejos. Su principal fortaleza es debido a que tiene un buen balance sensibilidad-especificidad, es decir este se desempeña mejor que PCA al evitar falsos positivos. Otra importante fortaleza es su simplicidad computacional, la cual es ideal para aplicaciones en tiempo real o con recursos limitados. Sin embargo, el algoritmo pierde efectividad en imágenes con iluminación irregular o fondos heterogéneos.

GA es robusto para problemas de segmentación no lineales, pero su implementación debe considerar trade-offs entre tiempo de procesamiento y precisión. Recomendado para entornos donde la exactitud es prioritaria y se dispone de recursos suficientes. Su principal fortaleza radica en que puede manejar múltiples óptimos locales en espacios de búsqueda complejos. Mientras que sus principales limitaciones son su convergencia lenta, ya que requiere más iteraciones y recursos computacionales y su inestabilidad en especificidad.

PSO es ideal para aplicaciones que requieren un equilibrio entre precisión y velocidad, como análisis de texturas en imágenes satelitales o biológicas. Su rendimiento justifica el costo computacional en escenarios que demandan robustez.

CP es el método de referencia para aplicaciones críticas donde los errores son inaceptables (diagnóstico médico, control de calidad industrial). Su implementación en sistemas de alto rendimiento está justificada por su precisión sin precedentes. Sin embargo, su principal limitación radica en su complejidad computacional.

Los resultados demuestran que la elección del algoritmo depende del contexto. No existe un método "universal". CP establece un nuevo estándar en precisión, pero su adopción debe evaluarse contra requisitos de velocidad y coste. Métodos clásicos (Otsu, PCA) siguen siendo relevantes en escenarios con limitaciones técnicas. En resumen, este estudio proporciona un marco claro para seleccionar algoritmos de segmentación según necesidades específicas, destacando el equilibrio fundamental entre precisión, robustez y eficiencia.

Como Trabajo Futuro, se pueden implementar híbridos, es decir combinar CP con PSO/GA para mejorar sensibilidad sin sacrificar especificidad. Otro trabajo futuro podría ser la optimización o reducción de los costos computacionales de CP mediante hardware especializado (GPUs).

Bibliografía

- [1] Mohd Nadhir Ab Wahab, Samia Nefti-Meziani, and Adham Atyabi. A comprehensive review of swarm optimization algorithms. *PloS one*, 10(5):e0122827, 2015.
- [2] S Abdel-Khalek, Anis Ben Ishak, Osama A Omer, and A-SF Obada. A two-dimensional image segmentation method based on genetic algorithm and entropy. *Optik*, 131:414–422, 2017.
- [3] Alex Alexandridis, Panagiotis Patrinos, Haralambos Sarimveis, and George Tsekouras. A two-stage evolutionary algorithm for variable selection in the development of rbf neural network models. *Chemometrics and intelligent laboratory systems*, 75(2):149–162, 2005.
- [4] Helon Vicente Hultmann Ayala, Fernando Marins dos Santos, Viviana Cocco Mariani, and Leandro dos Santos Coelho. Image thresholding segmentation based on a novel beta differential evolution approach. *Expert Systems with Applications*, 42(4):2136–2142, 2015.
- [5] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.
- [6] Abhishta Bhandari, Jarrad Koppen, and Marc Agzarian. Convolutional neural networks for brain tumour segmentation. *Insights into Imaging*, 11(1):1–9, 2020.
- [7] Bir Bhanu, Sungkee Lee, and John Ming. Adaptive image segmentation using a genetic algorithm. *IEEE Transactions on systems, man, and cybernetics*, 25(12):1543–1567, 1995.
- [8] Surina Borjigin and Prasanna K Sahoo. Color image segmentation based on multi-level tsallis–havrda–charvát entropy and 2d histogram using pso algorithms. *Pattern Recognition*, 92:107–118, 2019.
- [9] Toby Breckon and Chris Solomon. *Fundamentals of digital image processing*. 978 0 470 84472 4, 2011.

- [10] Jared Cervantes Canales, Jair Cervantes Canales, Farid García-Lamont, Arturo Yee-Rendon, José Sergio Ruiz Castilla, and Lisbeth Rodriguez Mazahua. Optimal segmentation of image datasets by genetic algorithms using color spaces. *Expert Systems with Applications*, 238:121950, March 2024.
- [11] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.
- [12] Kenneth R Castleman. *Digital image processing*. Prentice Hall Press, 1996.
- [13] Jair Cervantes, Jared Cervantes, Farid García-Lamont, Arturo Yee-Rendon, Josué Espejel Cabrera, and Laura Domínguez Jalili. A comprehensive survey on segmentation techniques for retinal vessel segmentation. *Neurocomputing*, 556:126626, November 2023.
- [14] Jair Cervantes, Xiaoou Li, and Yu Wen. Support vector classification for large data sets by reducing training data with change of classes. In *2008 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, oct 2008.
- [15] Jair Cervantes, Xiaoou Li, and Wen Yu. Support vector machine classification based on fuzzy clustering for large data sets. In *Lecture Notes in Computer Science*, pages 572–582. Springer Berlin Heidelberg, 2006.
- [16] Jair Cervantes, Xiaoou Li, and Wen Yu. SVM classification for large data sets by considering models of classes distribution. In *2007 Sixth Mexican International Conference on Artificial Intelligence, Special Session (MICAI)*. IEEE, nov 2007.
- [17] Jair Cervantes, Xiaoou Li, and Wen Yu. Using genetic algorithm to improve classification accuracy on imbalanced data. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, oct 2013.
- [18] Jair Cervantes, Xiaoou Li, and Wen Yu. Imbalanced data classification via support vector machines and genetic algorithms. *Connection Science*, 26(4):335–348, jun 2014.
- [19] Jair Cervantes, Xiaoou Li, Wen Yu, and Kang Li. Support vector machine classification for large data sets via minimum enclosing ball clustering. *Neurocomputing*, 71(4-6):611–619, jan 2008.
- [20] Jared Cervantes Canales. Desarrollo de algoritmos evolutivos para la segmentación automática de imágenes.
- [21] Tiyasa Chakraborty, Samiran Kumar Banik, Ashok Kumar Bhadra, and Debashis Nandi. Dynamically learned pso based neighborhood influenced fuzzy c-means for pre-treatment and post-treatment organ segmentation from ct images. *Computer Methods and Programs in Biomedicine*, 202:105971, 2021.

- [22] Akhilesh Chander, Amitava Chatterjee, and Patrick Siarry. A new social and momentum component adaptive pso algorithm for image segmentation. *Expert Systems with Applications*, 38(5):4998–5004, 2011.
- [23] Jiaochen Chen, Zhennao Cai, Ali Asghar Heidari, Huiling Chen, Qiuxiang He, José Escorcia-Gutierrez, and Romany F Mansour. Multi-threshold image segmentation based on an improved differential evolution: Case study of thyroid papillary carcinoma. *Biomedical Signal Processing and Control*, 85:104893, 2023.
- [24] Liping Chen, Jinhui Gao, António M Lopes, Zhiqiang Zhang, Zhaobi Chu, and Ranchao Wu. Adaptive fractional-order genetic-particle swarm optimization otsu algorithm for image segmentation. *Applied Intelligence*, 53(22):26949–26966, 2023.
- [25] James M Coggins and Anil K Jain. A spatial filtering approach to texture analysis. *Pattern recognition letters*, 3(3):195–203, 1985.
- [26] Mary L Comer and Edward J Delp III. Morphological operations for color image processing. *Journal of electronic imaging*, 8(3):279–289, 1999.
- [27] Erik Cuevas, Daniel Zaldivar, and Marco Pérez-Cisneros. A novel multi-threshold segmentation approach based on differential evolution optimization. *Expert Systems with Applications*, 37(7):5265–5271, 2010.
- [28] Antonios Danelakis, Theoharis Theoharis, and Dimitrios A Verganelakis. Survey of automated multiple sclerosis lesion segmentation techniques on magnetic resonance imaging. *Computerized Medical Imaging and Graphics*, 70:83–100, 2018.
- [29] Poonam Dhankhar and Neha Sahu. A review and research of edge detection techniques for image segmentation. *International Journal of Computer Science and Mobile Computing*, 2(7):86–92, 2013.
- [30] Dennis Dunn and William E Higgins. Optimal gabor filters for texture segmentation. *IEEE Transactions on image processing*, 4(7):947–964, 1995.
- [31] Josué Espejel-Cabrera, Jair Cervantes, Farid García-Lamont, José Sergio Ruiz Castilla, and Laura D Jalili. Mexican sign language segmentation using color based neuronal networks to detect the individual skin color. *Expert Systems with Applications*, 183:115295, nov 2021.
- [32] Du Feng, Shi Wenkang, Chen Liangzhou, Deng Yong, and Zhu Zhenfu. Infrared image segmentation with 2-d maximum entropy method based on particle swarm optimization (pso). *Pattern Recognition Letters*, 26(5):597–603, 2005.

- [33] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019.
- [34] Farid Garcia-Lamont, Jair Cervantes, Asdrúbal López, and Lisbeth Rodriguez. Segmentation of images by color features: A survey. *Neurocomputing*, 292:1–27, 2018.
- [35] Ernesto García Amaro, Jair Cervantes Canales, Farid García Lamont, Francisco Marcelo Lara Viveros, José Sergio Ruiz Castilla, and Josué Espejel Cabrera. Use of computer vision techniques for recognition of diseases and pests in tomato plants. *Computación y Sistemas*, 28(2), June 2024.
- [36] Payel Ghosh, Melanie Mitchell, James A Tanyi, and Arthur Y Hung. Incorporating priors for medical image segmentation using a genetic algorithm. *Neurocomputing*, 195:181–194, 2016.
- [37] David E Goldberg. Genetic algorithms in search. *Optimization, Machine Learning*, 1989.
- [38] Darsana. K. Gopidas and Dr. R. Priya. Hybrid segmentation method for boundary delineation of agricultural fields in multitemporal satellite image using hs-pso-fcnn. *Materials Today: Proceedings*, 51:2272–2276, 2022.
- [39] Jatin Gupta, Sumindar Kaur Saini, and Mamta Juneja. Survey of denoising and segmentation techniques for mri images of prostate for improving diagnostic tools in medical applications. *Materials Today: Proceedings*, 28:1667–1672, 2020.
- [40] Michal Haindl and Stanislav Mikeš. Model-based texture segmentation. In *International Conference Image Analysis and Recognition*, pages 306–313. Springer, 2004.
- [41] Ali Hatamizadeh, Assaf Hoogi, Debleena Sengupta, Wuyue Lu, Brian Wilcox, Daniel Rubin, and Demetri Terzopoulos. Deep active lesion segmentation. In *International Workshop on Machine Learning in Medical Imaging*, pages 98–105. Springer, 2019.
- [42] Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multi-scale filters for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3562–3572, 2019.
- [43] Rui Huang, Nong Sang, Dapeng Luo, and Qiling Tang. Image segmentation via coherent clustering in lab color space. *Pattern Recognition Letters*, 32(7):891–902, May 2011.

- [44] Che-Lun Hung and Yuan-Huai Wu. Parallel genetic-based algorithm on multiple embedded graphic processing units for brain magnetic resonance imaging segmentation. *Computers & Electrical Engineering*, 61:373–383, 2017.
- [45] Bram Huygens, Kyriakos Efthymiadis, Ann Nowé, and Gert Desmet. Application of evolutionary algorithms to optimise one-and two-dimensional gradient chromatographic separations. *Journal of Chromatography A*, 1628:461435, 2020.
- [46] Monzurul Islam, Anh Dinh, Khan Wahid, and Pankaj Bhowmik. Detection of potato diseases using image segmentation and multiclass support vector machine. In *2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE)*, pages 1–4. IEEE, 2017.
- [47] Laura D. Jalili, Alfredo Morales, Jair Cervantes, and José S. Ruiz-Castilla. Improving the performance of leaves identification by features selection with genetic algorithms. In *Communications in Computer and Information Science*, pages 103–114. Springer International Publishing, 2016.
- [48] Huilin Jiang, Pei Li, Zhihang Li, En Tong, Zhiwen Pan, Nan Liu, and Xiaohu You. Improved mpso based eicic algorithm for lte-a ultra dense hetnets. In *2014 IEEE Global Communications Conference*, pages 3678–3683. IEEE, 2014.
- [49] S Jose and C Vijayalakshmi. Design and analysis of multi-objective optimization problem using evolutionary algorithms. *Procedia Computer Science*, 172:896–899, 2020.
- [50] Delphine Jouan-Rimbaud, Desire-Luc Massart, Riccardo Leardi, and Onno E De Noord. Genetic algorithms as a tool for wavelength selection in multivariate calibration. *Analytical Chemistry*, 67(23):4295–4301, 1995.
- [51] Ahmad Khan and Muhammad Arfan Jaffar. Genetic algorithm and self organizing map based fuzzy hybrid intelligent method for color image segmentation. *Applied soft computing*, 32:300–310, 2015.
- [52] Muhammad Waseem Khan. A survey: Image segmentation techniques. *International Journal of Future Computer and Communication*, 3(2):89, 2014.
- [53] Akbar Shahrzad Khashandarag, Mirkamal Mirnia, and Aidin Sakhavati. A new method for medical image clustering using genetic algorithm. *International Journal of Computer Science Issues (IJCSI)*, 10(1):551, 2013.
- [54] K Krishna and M Narasimha Murty. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439, 1999.

- [55] Janus J Kulikowski, S Marčelja, and Peter O Bishop. Theory of spatial position and spatial frequency relations in the receptive fields of simple cells in the visual cortex. *Biological cybernetics*, 43(3):187–198, 1982.
- [56] Paulo Roberto Gardel Kurka and Aldo André Díaz Salazar. Applications of image processing in robotics and instrumentation. *Mechanical Systems and Signal Processing*, 124:142–169, 2019.
- [57] Zeki Kuş, Berna Kiraz, Tuğçe Koçak Göksu, Musa Aydın, Esra Özkan, Atay Vural, Alper Kiraz, and Burhanettin Can. Differential evolution-based neural architecture search for brain vessel segmentation. *Engineering Science and Technology, an International Journal*, 46:101502, 2023.
- [58] Xiaoou Li, Jair Cervantes, and Wen Yu. Two-stage svm classification for large data sets via randomly reducing and recovering training data. In *2007 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2007.
- [59] Xiaoou Li, Jair Cervantes, and Wen Yu. A novel SVM classification method for large data sets. In *2010 IEEE International Conference on Granular Computing*. IEEE, aug 2010.
- [60] Madhubanti Maitra and Amitava Chatterjee. A hybrid cooperative-comprehensive learning based pso algorithm for image segmentation using multilevel thresholding. *Expert Systems with Applications*, 34(2):1341–1350, 2008.
- [61] Veena Mayya, Sowmya Kamath, and Uma Kulkarni. Automated micro-aneurysms detection for early diagnosis of diabetic retinopathy: A comprehensive review. *Computer Methods and Programs in Biomedicine Update*, 1:100013, 2021.
- [62] Marina Meilă. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd international conference on Machine learning*, pages 577–584, 2005.
- [63] Ranjan Muthukrishnan and Miyilsamy Radha. Edge detection techniques for image segmentation. *International Journal of Computer Science & Information Technology*, 3(6):259, 2011.
- [64] Songyot Nakariyakul and David P Casasent. An improvement on floating search algorithms for feature subset selection. *Pattern Recognition*, 42(9):1932–1940, 2009.
- [65] Hossein Nematzadeh, José García-Nieto, Ismael Navas-Delgado, and José F Aldana-Montes. Ensemble-based genetic algorithm explainer with automatized image segmentation: A case study on melanoma detection dataset. *Computers in Biology and Medicine*, 155:106613, 2023.

- [66] Taher Niknam and Babak Amiri. An efficient hybrid approach based on pso, aco and k-means for cluster analysis. *Applied soft computing*, 10(1):183–197, 2010.
- [67] Jifeng Ning, Lei Zhang, David Zhang, and Chengke Wu. Interactive image segmentation by maximal similarity based region merging. *Pattern Recognition*, 43(2):445–456, 2010.
- [68] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
- [69] Stelios E Papadakis, Panagiotis Tzionas, Vassilis G Kaburlasos, and John B Theocharis. A genetic based approach to the type i structure identification problem. *Informatica*, 16(3):365–382, 2005.
- [70] Danilo Cesar Pereira, Rodrigo Pereira Ramos, and Marcelo Zanchetta Do Nascimento. Segmentation and detection of breast cancer in mammograms combining wavelet analysis and genetic algorithm. *Computer methods and programs in biomedicine*, 114(1):88–101, 2014.
- [71] Thuy Xuan Pham, Patrick Siarry, and Hamouche Oulhadj. Integrating fuzzy entropy clustering with an improved pso for mri brain image segmentation. *Applied Soft Computing*, 65:230–242, 2018.
- [72] Kenneth Price, Rainer M Storn, and Jouni A Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.
- [73] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.
- [74] Colin R Reeves. *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, Inc., 1993.
- [75] Lili Ren, Dong Zhao, Xuehua Zhao, Weibin Chen, Lingzhi Li, TaiSong Wu, Guoxi Liang, Zhennao Cai, and Suling Xu. Multi-level thresholding segmentation for pathological images: Optimal performance design of a new modified differential evolution. *Computers in Biology and Medicine*, 148:105910, 2022.
- [76] JE Ruiz, S Paciornik, LD Pinto, F Ptak, MP Pires, and PL Souza. Optimization of digital image processing to determine quantum dots’ height and density from atomic force microscopy. *Ultramicroscopy*, 184:234–241, 2018.
- [77] Mehmet Sezgin and Buğlent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, 13(1):146–168, 2004.

- [78] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, pages 731–737. IEEE, 1997.
- [79] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis and machine vision*. Springer, 2013.
- [80] Rainer Storn. Modeling and optimization of pet-redundancy assignment for mpeg sequences. *International Computer Science Institute-Publications-TR*, 1995.
- [81] Rainer Storn and Kenneth Price. Minimizing the real functions of the icec’96 contest by differential evolution. In *Proceedings of IEEE international conference on evolutionary computation*, pages 842–844. IEEE, 1996.
- [82] Rainer Storn and Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11:341–359, 1997.
- [83] Ran Tao, Mostafa Sharifzadeh, Yan Zhang, and Xia-Ting Feng. Analysis of mafic rocks microstructure damage and failure process under compression test using quantitative scanning electron microscopy and digital images processing. *Engineering Fracture Mechanics*, 231:107019, 2020.
- [84] Omid Tarkhaneh and Haifeng Shen. An adaptive differential evolution algorithm to optimal multi-level thresholding for mri brain image segmentation. *Expert Systems with Applications*, 138:112820, 2019.
- [85] Leslie Ching Ow Tiong, Seong Tae Kim, and Yong Man Ro. Multimodal facial biometrics recognition: Dual-stream convolutional neural networks with multi-feature fusion layers. *Image and Vision Computing*, 102:103977, 2020.
- [86] Thaína A Azevedo Tosta, Paulo Rogério Faria, Leandro Alves Neves, and Marcelo Zanchetta do Nascimento. Computational method for unsupervised segmentation of lymphoma histological images based on fuzzy 3-partition entropy and genetic algorithm. *Expert Systems with Applications*, 81:223–243, 2017.
- [87] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Graph cut based image segmentation with connectivity priors. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.
- [88] Hongzhi Wang and John Oliensis. Generalizing edge detection to contour detection for image segmentation. *Computer Vision and Image Understanding*, 114(7):731–744, 2010.

- [89] Jie Wang, Lili Ju, and Xiaoqiang Wang. An edge-weighted centroidal voronoi tessellation model for image segmentation. *IEEE Transactions on Image Processing*, 18(8):1844–1858, 2009.
- [90] Min Xian, Yingtao Zhang, Heng-Da Cheng, Fei Xu, Boyu Zhang, and Jianrui Ding. Automatic breast ultrasound image segmentation: A survey. *Pattern Recognition*, 79:340–355, 2018.
- [91] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12193–12202, 2020.
- [92] Fengying Xie and Alan C Bovik. Automatic segmentation of dermoscopy images using self-generating neural networks seeded by genetic algorithm. *Pattern Recognition*, 46(3):1012–1019, 2013.
- [93] Jihoon Yang and Vasant Honavar. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems and their Applications*, 13(2):44–49, 1998.
- [94] Jinzhu Yang, Chunhui Lou, Jie Fu, and Chaolu Feng. Vessel segmentation using multiscale vessel enhancement and a region based level set model. *Computerized Medical Imaging and Graphics*, 85:101783, 2020.
- [95] Xiao Yang, Rui Wang, Dong Zhao, Fanhua Yu, Ali Asghar Heidari, Zhangze Xu, Huiling Chen, Abeer D Algarni, Hela Elmannai, and Suling Xu. Multi-level threshold segmentation framework for breast cancer images using enhanced differential evolution. *Biomedical Signal Processing and Control*, 80:104373, 2023.
- [96] Ian T Young and Lucas J Van Vliet. Recursive implementation of the gaussian filter. *Signal processing*, 44(2):139–151, 1995.
- [97] Xiaofeng Yue and Hongbo Zhang. Modified hybrid bat algorithm with genetic crossover operation and smart inertia weight for multilevel image segmentation. *Applied soft computing*, 90:106157, 2020.
- [98] Nida M Zaitoun and Musbah J Aqel. Survey on image segmentation techniques. *Procedia Computer Science*, 65:797–806, 2015.
- [99] Longfei Zeng and Sabah Mohammed. Eolo: Deep machine learning algorithm for embedded object segmentation that only looks once. *International Journal of Multimedia and Ubiquitous Engineering*, 15(1):35–48, 2020.
- [100] Haixia Zhang and Qingxiu Peng. Pso and k-means-based semantic segmentation toward agricultural products. *Future Generation Computer Systems*, 126:82–87, 2022.

- [101] Yong Zhang, Dan Huang, Min Ji, and Fuding Xie. Image segmentation using pso and pcm with mahalanobis distance. *Expert systems with applications*, 38(7):9036–9040, 2011.
- [102] Zhen Zhang, Meijuan Xu, Feifei Liu, Xinyu Wu, Yongfei Zhang, and Yuhao Shen. Multi-color space features and data regression neural network-based method for molten salt temperature detection in rare earth electrolytic processes. *International Communications in Heat and Mass Transfer*, 159:108069, December 2024.

Apéndice A

Artículos publicados

Neurocomputing 556 (2023) 126626



Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A comprehensive survey on segmentation techniques for retinal vessel segmentation

Jair Cervantes^{a,*}, Jared Cervantes^a, Farid García-Lamont^a, Arturo Yee-Rendon^b,
 Josué Espejel Cabrera^a, Laura Domínguez Jalili^a

^a UAEMEX (Autonomous University of Mexico State), Texcoco, 56259, Mexico

^b Facultad de Informática Culiacán, Universidad Autónoma de Sinaloa, Culiacán, Sinaloa, 80013, Mexico

ARTICLE INFO

Keywords:

Vessel segmentation
 Preprocessing techniques
 Segmentation techniques

ABSTRACT

In recent years, enormous research has been carried out on the segmentation of blood vessels. Segmentation of blood vessels in retinal images is crucial for diagnosing, treating, evaluating clinical results, and early detection of eye disorders. A successful segmentation accurately reflects the blood vessels' structure and helps obtain patterns that can be used to identify retinal disorders and diseases. Most recent research on vessel segmentation employs multiple processes to determine the appropriate segmentation. Finding the best techniques for segmentation is a complex process. In certain circumstances, it requires a thorough understanding of every step that can only be acquired through years of training. Comprehension and expertise in segmentation procedures are essential for accurate segmentation. This paper briefly introduces the segmentation of blood vessels in retinal images, describes many preprocessing and segmentation techniques, and summarizes challenges and trends. Furthermore, the limitations of the current systems will be identified.

1. Introduction

The current development of approaches based on artificial intelligence and digital image processing has been made possible by the growth in the processing power of computers and memory capacity. Many researchers have been interested in developing vision and image processing systems to address challenging issues in real-life scenarios.

Vision systems are widely used in many applications in real life. Vision techniques have been implemented to solve problems in many areas of medicine [1–5]. The processing and analysis of data used in decision-making are aided and facilitated by vision systems. Artificial vision plays a crucial role in medical imaging and healthcare applications. It involves using advanced algorithms to interpret medical images, enabling healthcare professionals to make more accurate diagnoses and provide better patient care. Thus, computer vision algorithms are used in medical image analysis to detect and diagnose specific diseases or conditions.

In neurology, there are computer-aided diagnosis (CAD) systems to help doctors to make rapid diagnoses by classifying normal/abnormal Magnetic Resonance (MR) brain images [6], there are also images acquired from invasive methods like surgeries, craniotomy, and in some cases after the resection of brain tissue for tumor treatments, the images are used not only to segment brain vessels [7] but also to

classify tumors and parts of the brain [8], cerebral small vessel disease (SVD) [9–11].

The Oncology area encompasses diverse body organ studies; vessel segmentation can help doctors identify tumor-associated blood vessels more accurately and obtain early diagnosis to select an appropriate treatment. Lung cancer is one of the most common diseases suffered by the people. The vessel segmentation can help in treatment finding the tumor vessels [12,13]. For liver cancer, computer vision can help to obtain the complex of the structure [14–16] and obtain the vessels to irrigate the tumor.

In recent years COVID-19 caught the researcher's attention [17–19]. One of the sequelae of the disease is fibrosis, which can be abroached using the vessel segmentation of the lung arteries. Another kind of fibrosis we can find from human illnesses that medics can improve with vessel segmentation is idiopathic pulmonary fibrosis (IPF) [20,21].

The medical surgery process can be improved using vessel segmentation on medical images, obtaining more accurate diagnoses and more efficient steps in surgical planning using computed tomography (CT) images. From those images, the doctors obtain the vessel's structure, and they can observe anomalies to plan the liver surgery [22]. The vessel segmentation can guide medics with aortic ruptures [23] to plan the surgery.

* Corresponding author.

E-mail addresses: jcervantes@uaemex.mx (J. Cervantes), jcervantes002@alumno.uaemex.mx (J. Cervantes), fgarcial@uaemex.mx (F. García-Lamont),



Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswaOptimal segmentation of image datasets by genetic algorithms using color spaces Jared Cervantes Canales^a, Jair Cervantes Canales^{a,*}, Farid García-Lamont^a,
Arturo Yee-Rendon^b, José Sergio Ruiz Castilla^a, Lisbeth Rodríguez Mazahua^c^a Department of Computer Science, Autonomous University of Mexico State, Av. Zumpango s/n, Fraccionamiento el Tejocote, Texcoco, 56255, Mexico^b Facultad de Informática Culiacán, Universidad Autónoma de Sinaloa, Culiacán, Sinaloa, 80007, Mexico^c Division of Research and Postgraduate Studies, Instituto Tecnológico de Orizaba, Veracruz 94320, Mexico

ARTICLE INFO

Keywords:

Automatic segmentation
Images datasets
Genetic algorithm

ABSTRACT

A vision system comprises several steps, with each step exerting a significant impact on the final outcome. One of these crucial steps is segmentation, which isolates the region of interest within an object and removes the background. Segmentation is vital because it enhances the quality of the isolated region, improves the accuracy of extracted features, and reduces the noise introduced by poor-quality features. Several segmentation techniques are available in the literature, each requiring one or more adjustable parameters. Selecting the most appropriate technique for segmenting a particular image dataset can be challenging. Several factors can affect the segmentation quality, including preprocessing, the chosen segmentation method, parameter fine-tuning approaches, and the characteristics of the dataset. Moreover, variations in lighting and intensity can further influence segmentation quality. At times, an expert may need to manually choose the segmentation technique and fine-tune its associated parameters. This paper presents the development of an automated algorithm for the selection of segmentation techniques and their associated parameters. The developed techniques are implemented and compared using diverse datasets, and the resulting experimental outcomes are thoroughly discussed and analyzed. The algorithm aims to streamline and simplify the process of selecting appropriate segmentation techniques, determining the required parameters, and selecting suitable pre-processing techniques.

1. Introduction

The development of automatic vision and image processing systems has garnered significant attention due to their potential to address real-life problems. Many areas such as medicine (Abdel-Basset, Hawash, & Chang, 2023; Danelakis, Theoharis, & Verganelakis, 2018; Gupta, Saini, & Juneja, 2020; Xian et al., 2018), civil engineering (Tao, Sharifzadeh, Zhang, & Feng, 2020), Astronomy (Ruiz et al., 2018), security (Tiong, Kim, & Ro, 2020), robotics (Kurka & Salazar, 2019), and agriculture (Martinez et al., 2020). These areas frequently work together and make use of genetic data, visual data, and image processing tools to process and evaluate information necessary for well-informed decision-making.

Indeed, despite the substantial progress made in digital image processing and vision systems, they continue to face several challenges

and issues. Vision systems depend on several steps, and each step is carried out by some technique that depends on several parameters and variables. A vision system generally consists of pre-processing, segmentation, feature extraction, feature selection, and classification. Segmentation is undeniably a pivotal phase in the operation of a vision system. The segmentation process in a visual system is both crucial and intricate. Nevertheless, even minor changes can sometimes lead to less-than-optimal segmentation results. These changes can be in the pre-processing, variables of each technique, lighting or intensity of the images, or even morphological operations such as dilation or erosion to polish the final image. The primary objective of this work is to develop an automatic algorithm for the segmentation of image datasets.

Several segmentation techniques are in the literature, each with one or more variables to refine. Choosing the proper method to segment a specific dataset is a challenging task. Multiple factors can influence the

The code (and data) in this article has been certified as Reproducible by Code Ocean: (<https://codeocean.com/>). More information on the Reproducibility Badge Initiative is available at <https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals>.

* Corresponding author.

E-mail addresses: jcervantesc002@alumno.uaemex.mx (J.C. Canales), jcervantesc@uaemex.mx (J.C. Canales), fgarcial@uaemex.mx (F. García-Lamont), arturo.yee@uas.edu.mx (A. Yee-Rendon), jrsruizc@uaemex.mx (J.S.R. Castilla), lrodriguez@orizaba.tecnm.mx (L.R. Mazahua).

<https://doi.org/10.1016/j.eswa.2023.121950>

Received 29 August 2021; Received in revised form 29 September 2023; Accepted 29 September 2023

Available online 14 October 2023

0957-4174/© 2023 Elsevier Ltd. All rights reserved.



International Conference on Intelligent Computing
 ICIC 2022: **Intelligent Computing Theories and Application** pp 391–400 | [Cite as](#)

Optimization of Vessel Segmentation Using Genetic Algorithms

Jared Cervantes, Dalia Luna, Jair Cervantes  & Farid García-Lamont

Conference paper | [First Online: 15 August 2022](#)
 519 Accesses

Part of the [Lecture Notes in Computer Science](#) book series (LNCS, volume 13393)

Abstract

Vessel segmentation is an important task to extract helpful information from retinal images that can help make a retinopathy diagnosis. A good segmentation perfectly represents the structure and obtains patterns that diagnose retinal diseases. Most of the current methods require many parameters, and the final quality of vessel segmentation depends on these parameters, which increases the complexity of the methods. We propose a new Vessel segmentation algorithm to address these issues using genetic algorithms. The method uses

[Access via your institution](#) →

Chapter USD 29.95
Price excludes VAT (Mexico)

- DOI: 10.1007/978-3-031-13870-6_32
- Chapter length: 19 pages
- Instant PDF download
- Readable on all devices
- Own it forever
- Exclusive offer for individuals only
- Tax calculation will be finalised during checkout

[Buy Chapter](#)

[eBook](#) USD 99.00
[Softcover Book](#) USD 129.99

[Learn about institutional subscriptions](#)

Figura A.3: ICIC 2022

Apéndice B

Tablas de resultados

Tabla B.1: Resultados detallados del método de PCA para el conjunto de datos ICL

Imagen	TP	TN	FP	FN	Accuracy	Sensib.	Especif.	Exactitud	PRI	VOI	GCE	DJ
001.jpg	33 326	62 376	302	8	0,996 771	0,999 760	0,995 182	0,997 471	0,993 418	0,058 015	0,006 544	0,992 760
002.jpg	34 533	111 186	132	7	0,999 047	0,999 797	0,998 814	0,999 306	0,998 082	0,020 196	0,001 913	0,997 352
003.jpg	34 249	73 908	140	38	0,998 357	0,998 892	0,998 109	0,998 501	0,996 646	0,034 589	0,003 348	0,996 128
004.jpg	22 566	66 946	140	9	0,998 338	0,999 601	0,997 913	0,998 757	0,996 637	0,032 939	0,003 348	0,995 551
005.jpg	30 968	74 729	116	27	0,998 649	0,999 129	0,998 450	0,998 790	0,997 207	0,029 293	0,002 788	0,996 633
006.jpg	30 270	77 180	79	61	0,998 699	0,997 989	0,998 977	0,998 483	0,997 345	0,028 786	0,002 653	0,996 724
007.jpg	34 586	83 545	89	52	0,998 808	0,998 499	0,998 936	0,998 717	0,997 517	0,027 084	0,002 481	0,997 005
008.jpg	41 754	75 944	130	28	0,998 659	0,999 330	0,998 291	0,998 810	0,997 238	0,029 150	0,002 759	0,996 985
009.jpg	22 836	93 678	114	16	0,998 885	0,999 300	0,998 785	0,999 042	0,997 619	0,024 639	0,002 372	0,996 233
010.jpg	51 014	227 482	346	8	0,998 730	0,999 843	0,998 481	0,999 162	0,997 414	0,025 426	0,002 572	0,995 694
011.jpg	56 808	150 374	433	10	0,997 866	0,999 824	0,997 129	0,998 476	0,995 646	0,040 459	0,004 331	0,994 542
012.jpg	20 870	54 950	212	0	0,997 212	1,000 000	0,996 157	0,998 078	0,994 413	0,049 084	0,005 546	0,993 018
013.jpg	30 388	57 730	83	17	0,998 866	0,999 441	0,998 564	0,999 003	0,997 600	0,025 726	0,002 397	0,997 345
014.jpg	26 898	48 285	86	7	0,998 765	0,999 740	0,998 222	0,998 981	0,997 479	0,026 228	0,002 516	0,997 258
015.jpg	40 795	79 920	196	16	0,998 247	0,999 608	0,997 554	0,998 581	0,996 450	0,035 072	0,003 540	0,996 037
016.jpg	52 362	121 547	440	1	0,997 471	0,999 981	0,996 393	0,998 187	0,994 897	0,045 804	0,005 073	0,993 949
017.jpg	53 034	94 990	253	27	0,998 112	0,999 491	0,997 344	0,998 417	0,996 177	0,037 777	0,003 813	0,995 846
018.jpg	36 982	77 643	147	34	0,998 423	0,999 081	0,998 110	0,998 596	0,996 765	0,033 378	0,003 229	0,996 303
019.jpg	15 770	148 417	210	13	0,998 644	0,999 176	0,998 587	0,998 882	0,997 255	0,026 137	0,002 715	0,992 174
020.jpg	10 600	105 313	222	2	0,998 071	0,999 811	0,997 896	0,998 854	0,996 098	0,034 108	0,003 829	0,988 439
021.jpg	19 804	62 141	310	1	0,996 219	0,999 950	0,995 036	0,997 493	0,992 346	0,063 583	0,007 563	0,989 615
022.jpg	51 171	108 822	295	2	0,998 147	0,999 961	0,997 296	0,998 629	0,996 239	0,035 617	0,003 746	0,995 652
023.jpg	49 919	140 048	285	8	0,998 460	0,999 840	0,997 969	0,998 904	0,996 809	0,031 076	0,003 177	0,995 890
024.jpg	36 595	98 980	106	47	0,998 873	0,998 717	0,998 930	0,998 824	0,997 675	0,025 366	0,002 323	0,997 053
025.jpg	7808	19 726	106	8	0,995 877	0,998 976	0,994 655	0,996 816	0,991 500	0,072 753	0,008 425	0,989 579
1.jpg	41 199	106 364	151	8	0,998 924	0,999 806	0,998 582	0,999 194	0,997 769	0,023 171	0,002 226	0,997 231

Nota: Sensib. = Sensibilidad, Especif. = Especificidad. Valores destacados cercanos a 1.0 indican mejor desempeño.

Tabla B.2: Resultados detallados del método de Otsu para el conjunto de datos ICL

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
001.jpg	33 254	62 591	87	80	0.998 261	0.997 600	0.998 612	0.998 106	0.996 382	0.037 963	0.003 617	0.996 011
002.jpg	34 462	111 297	21	78	0.999 321	0.997 742	0.999 811	0.998 777	0.998 630	0.015 739	0.001 369	0.998 104
003.jpg	34 156	74 022	26	131	0.998 551	0.996 179	0.999 649	0.997 914	0.997 032	0.030 986	0.002 963	0.996 569
004.jpg	22 517	67 046	40	58	0.998 907	0.997 431	0.999 404	0.998 417	0.997 772	0.024 597	0.002 227	0.997 044
005.jpg	30 877	74 827	18	118	0.998 715	0.996 193	0.999 760	0.997 976	0.997 377	0.027 601	0.002 618	0.996 832
006.jpg	30 183	77 247	12	148	0.998 513	0.995 121	0.999 845	0.997 483	0.996 975	0.030 599	0.003 017	0.996 261
007.jpg	34 443	83 618	16	195	0.998 216	0.994 370	0.999 809	0.997 090	0.996 371	0.035 885	0.003 617	0.995 616
008.jpg	41 638	76 054	20	144	0.998 608	0.996 554	0.999 737	0.998 145	0.997 136	0.029 907	0.002 859	0.996 870
009.jpg	22 768	93 782	10	84	0.999 194	0.996 324	0.999 893	0.998 109	0.998 235	0.019 326	0.001 761	0.997 199
010.jpg	50 933	227 751	77	89	0.999 405	0.998 256	0.999 662	0.998 959	0.998 760	0.014 548	0.001 239	0.997 928
011.jpg	56 678	150 736	71	140	0.998 984	0.997 536	0.999 529	0.998 533	0.997 873	0.023 630	0.002 125	0.997 326
012.jpg	20 860	55 106	56	10	0.999 132	0.999 521	0.998 985	0.999 253	0.998 239	0.019 448	0.001 758	0.997 792
013.jpg	30 298	57 795	18	107	0.998 583	0.996 481	0.999 689	0.998 085	0.997 034	0.031 037	0.002 961	0.996 716
014.jpg	26 853	48 357	14	52	0.999 123	0.998 067	0.999 711	0.998 889	0.998 195	0.020 380	0.001 804	0.998 035
015.jpg	40 671	80 079	37	140	0.998 536	0.996 570	0.999 538	0.998 054	0.997 027	0.031 339	0.002 969	0.996 675
016.jpg	52 318	121 852	135	45	0.998 968	0.999 141	0.998 893	0.999 017	0.997 880	0.023 339	0.002 118	0.997 480
017.jpg	52 967	95 177	66	94	0.998 921	0.998 228	0.999 307	0.998 768	0.997 791	0.024 740	0.002 209	0.997 596
018.jpg	36 877	77 764	26	139	0.998 563	0.996 245	0.999 666	0.997 955	0.997 078	0.030 547	0.002 918	0.996 654
019.jpg	15 687	148 585	42	96	0.999 161	0.993 918	0.999 717	0.996 817	0.998 286	0.018 469	0.001 708	0.995 071
020.jpg	10 542	105 483	52	60	0.999 036	0.994 341	0.999 507	0.996 924	0.998 022	0.021 038	0.001 974	0.994 058
021.jpg	19 743	62 392	59	62	0.998 529	0.996 869	0.999 055	0.997 962	0.996 941	0.032 369	0.003 057	0.995 821
022.jpg	51 133	109 060	57	40	0.999 395	0.999 218	0.999 478	0.999 348	0.998 728	0.015 177	0.001 272	0.998 537
023.jpg	49 884	140 251	82	43	0.999 343	0.999 139	0.999 416	0.999 277	0.998 571	0.016 595	0.001 428	0.998 156
024.jpg	36 513	99 070	16	129	0.998 932	0.996 479	0.999 839	0.998 159	0.997 792	0.023 597	0.002 204	0.997 198
025.jpg	7776	19 811	21	40	0.997 794	0.994 882	0.998 941	0.996 912	0.995 309	0.046 939	0.004 685	0.994 218
1.jpg	41 093	106 488	27	114	0.999 046	0.997 233	0.999 747	0.998 490	0.998 012	0.021 944	0.001 986	0.997 528

Tabla B.3: Resultados detallados del método de AG para el conjunto de datos ICL

Imagen	TP	TN	FP	FN	Accuracy	Sens.	Espec.	Exact.	PRI	VOI	GCE	DJ
001.jpg	32 907	62 647	31	427	0,995 230	0,987 190	0,999 505	0,993 348	0,990 361	0,082 012	0,009 572	0,989 348
002.jpg	34 500	111 268	50	40	0,999 383	0,998 842	0,999 551	0,999 196	0,998 753	0,014 779	0,001 247	0,998 276
003.jpg	34 022	74 035	13	265	0,997 434	0,992 271	0,999 824	0,996 048	0,994 807	0,048 159	0,005 168	0,993 992
004.jpg	22 400	67 063	23	175	0,997 792	0,992 248	0,999 657	0,995 953	0,995 549	0,042 774	0,004 431	0,994 088
005.jpg	30 915	74 810	35	80	0,998 913	0,997 419	0,999 532	0,998 476	0,997 735	0,024 963	0,002 263	0,997 266
006.jpg	30 218	77 252	7	113	0,998 885	0,996 274	0,999 909	0,998 092	0,997 716	0,023 936	0,002 279	0,997 178
007.jpg	34 406	83 622	12	232	0,997 937	0,993 302	0,999 857	0,996 579	0,995 815	0,040 105	0,004 168	0,994 943
008.jpg	41 514	76 065	9	268	0,997 650	0,993 586	0,999 882	0,996 734	0,995 226	0,044 838	0,004 756	0,994 779
009.jpg	22 602	93 775	17	250	0,997 711	0,989 060	0,999 819	0,994 439	0,995 313	0,043 598	0,004 651	0,992 552
010.jpg	50 823	227 784	44	199	0,999 129	0,996 100	0,999 807	0,997 953	0,998 209	0,019 586	0,001 787	0,997 004
011.jpg	56 710	150 678	129	108	0,998 859	0,996 099	0,999 145	0,998 622	0,997 624	0,026 125	0,002 375	0,997 013
012.jpg	20 812	55 130	32	58	0,998 816	0,997 221	0,999 420	0,998 320	0,997 609	0,026 160	0,002 389	0,996 999
013.jpg	30 376	57 779	34	29	0,999 286	0,999 046	0,999 412	0,999 229	0,998 437	0,018 257	0,001 563	0,998 270
014.jpg	26 889	48 336	35	16	0,999 322	0,999 405	0,999 276	0,999 341	0,998 593	0,016 527	0,001 407	0,998 469
015.jpg	40 793	80 033	83	16	0,999 165	0,999 559	0,998 964	0,999 261	0,998 248	0,019 591	0,001 750	0,998 043
016.jpg	52 347	121 869	118	16	0,999 231	0,999 694	0,999 033	0,999 364	0,998 407	0,017 739	0,001 591	0,998 106
017.jpg	51 826	95 176	67	1235	0,991 221	0,976 725	0,999 297	0,988 011	0,982 596	0,131 821	0,017 171	0,980 999
018.jpg	36 716	77 609	181	300	0,995 810	0,991 895	0,997 673	0,994 784	0,991 569	0,077 768	0,008 416	0,990 356
019.jpg	15 111	148 612	15	672	0,995 821	0,957 423	0,999 899	0,978 661	0,991 642	0,064 931	0,008 045	0,975 851
020.jpg	10 199	105 514	21	403	0,996 349	0,961 988	0,998 801	0,980 895	0,992 708	0,058 779	0,007 056	0,977 988
021.jpg	19 687	62 346	105	118	0,997 289	0,994 042	0,998 319	0,996 180	0,994 472	0,053 818	0,005 522	0,992 452
022.jpg	51 099	109 032	85	74	0,999 008	0,998 554	0,999 221	0,998 887	0,997 956	0,023 055	0,002 044	0,997 649
023.jpg	49 874	140 187	146	53	0,998 954	0,998 938	0,998 960	0,998 949	0,997 795	0,024 053	0,002 202	0,997 155
024.jpg	36 568	99 084	2	74	0,999 440	0,997 980	0,999 980	0,998 980	0,998 807	0,013 613	0,001 191	0,998 486
025.jpg	7768	19 825	7	48	0,998 011	0,993 859	0,999 647	0,996 753	0,995 741	0,042 063	0,004 246	0,994 745
1.jpg	41 020	106 508	7	187	0,998 687	0,995 462	0,999 934	0,997 698	0,997 296	0,027 512	0,002 696	0,996 637

Nota: Sens. = Sensibilidad, Espec. = Especificidad, Exact. = Exactitud. Valores en negrita destacan los mejores resultados por columna.

Tabla B.4: Resultados detallados del método de PSO para el conjunto de datos ICL

Imagen	TP	TN	FP	FN	Accuracy	Sens.	Espec.	Exact.	PRI	VOI	GCE	DJ
001.jpg	33 311	62 339	339	23	0,996 230	0,999 310	0,994 591	0,996 951	0,992 343	0,067 126	0,007 612	0,991 579
002.jpg	34 535	111 304	14	5	0,999 870	0,999 855	0,999 874	0,999 865	0,999 726	0,003 799	0,000 274	0,999 621
003.jpg	34 225	74 046	2	62	0,999 409	0,998 192	0,999 973	0,999 082	0,998 745	0,014 319	0,001 253	0,998 550
004.jpg	22 571	67 075	11	4	0,999 833	0,999 823	0,999 836	0,999 829	0,999 621	0,005 078	0,000 379	0,999 497
005.jpg	30 961	74 794	51	34	0,999 197	0,998 903	0,999 319	0,999 111	0,998 301	0,019 494	0,001 698	0,997 950
006.jpg	30 250	77 227	32	81	0,998 950	0,997 329	0,999 586	0,998 458	0,997 846	0,023 784	0,002 152	0,997 340
007.jpg	34 542	83 627	7	96	0,999 129	0,997 228	0,999 916	0,998 572	0,998 158	0,020 122	0,001 839	0,997 776
008.jpg	41 674	76 069	5	108	0,999 041	0,997 415	0,999 934	0,998 675	0,998 000	0,021 548	0,001 998	0,997 814
009.jpg	22 846	93 788	4	6	0,999 914	0,999 737	0,999 957	0,999 847	0,999 674	0,004 428	0,000 326	0,999 483
010.jpg	50 993	227 765	63	29	0,999 670	0,999 432	0,999 723	0,999 578	0,999 290	0,008 807	0,000 709	0,998 814
011.jpg	56 774	150 741	66	44	0,999 470	0,999 226	0,999 562	0,999 394	0,998 845	0,013 850	0,001 155	0,998 548
012.jpg	20 869	55 025	137	1	0,998 185	0,999 952	0,997 516	0,998 734	0,996 350	0,034 505	0,003 633	0,995 432
013.jpg	30 387	57 802	11	18	0,999 671	0,999 408	0,999 810	0,999 609	0,999 207	0,010 052	0,000 793	0,999 122
014.jpg	26 893	48 367	4	12	0,999 787	0,999 554	0,999 917	0,999 736	0,999 522	0,006 375	0,000 478	0,999 480
015.jpg	40 741	80 059	57	70	0,998 950	0,998 285	0,999 289	0,998 787	0,997 852	0,024 123	0,002 147	0,997 599
016.jpg	52 350	121 864	123	13	0,999 230	0,999 752	0,998 992	0,999 372	0,998 384	0,017 846	0,001 614	0,998 079
017.jpg	53 041	95 081	162	20	0,998 773	0,999 623	0,998 299	0,998 961	0,997 495	0,026 371	0,002 501	0,997 276
018.jpg	37 002	77 734	56	14	0,999 390	0,999 622	0,999 280	0,999 451	0,998 694	0,015 182	0,001 305	0,998 507
019.jpg	15 743	148 442	185	40	0,998 631	0,997 466	0,998 755	0,998 110	0,997 230	0,027 305	0,002 746	0,992 094
020.jpg	10 581	105 317	218	21	0,997 942	0,998 019	0,997 934	0,997 977	0,995 841	0,037 439	0,004 090	0,987 671
021.jpg	19 805	62 054	397	0	0,995 174	1,000 000	0,993 643	0,996 822	0,990 273	0,077 203	0,009 580	0,986 832
022.jpg	51 173	108 816	301	0	0,998 122	1,000 000	0,997 241	0,998 621	0,996 189	0,035 797	0,003 795	0,995 626
023.jpg	49 927	140 058	275	0	0,998 555	1,000 000	0,998 040	0,999 020	0,996 998	0,028 973	0,002 989	0,996 133
024.jpg	36 639	99 048	38	3	0,999 698	0,999 918	0,999 616	0,999 767	0,999 322	0,008 233	0,000 677	0,999 141
025.jpg	7804	19 798	34	12	0,998 336	0,998 465	0,998 286	0,998 375	0,996 390	0,036 882	0,003 603	0,995 556
1.jpg	41 197	106 491	24	10	0,999 770	0,999 757	0,999 775	0,999 766	0,999 459	0,007 012	0,000 541	0,999 327

Nota: Sens. = Sensibilidad, Espec. = Especificidad, Exact. = Exactitud. Los valores en negrita indican los mejores resultados por columna.

Tabla B.5: Resultados detallados del método de Caracterización de Píxeles para el conjunto de datos ICL

Imagen	TP	TN	FP	FN	Accuracy	Sens.	Espec.	Exact.	PRI	VOI	GCE	DJ
001.jpg	33 334	62 656	0	0	0,999 771	1,000 000	1,000 000	1,000 000	0,999 396	0,007 288	0,000 604	0,999 334
002.jpg	34 538	111 318	0	2	0,999 986	0,999 942	1,000 000	0,999 971	0,999 959	0,000 687	0,000 041	0,999 943
003.jpg	34 287	74 048	0	0	1,000 000	1,000 000	1,000 000	0,999 926	0,001 112	0,000 074	0,000 074	0,999 915
004.jpg	22 573	67 086	0	2	0,999 978	0,999 911	1,000 000	0,999 956	0,999 911	0,001 400	0,000 089	0,999 882
005.jpg	30 980	74 845	0	15	0,999 858	0,999 516	1,000 000	0,999 758	0,999 622	0,005 097	0,000 378	0,999 544
006.jpg	30 316	77 259	0	15	0,999 861	0,999 505	1,000 000	0,999 753	0,999 665	0,004 512	0,000 334	0,999 587
007.jpg	34 633	83 634	0	5	0,999 958	0,999 856	1,000 000	0,999 928	0,999 814	0,002 733	0,000 186	0,999 775
008.jpg	41 762	76 074	0	20	0,999 830	0,999 521	1,000 000	0,999 761	0,999 576	0,005 644	0,000 424	0,999 537
009.jpg	22 845	93 792	0	7	0,999 940	0,999 694	1,000 000	0,999 847	0,999 726	0,003 821	0,000 274	0,999 565
010.jpg	51 017	227 828	0	5	0,999 982	0,999 902	1,000 000	0,999 951	0,999 914	0,001 339	0,000 086	0,999 856
011.jpg	56 818	150 791	0	0	0,999 923	1,000 000	1,000 000	1,000 000	0,999 750	0,003 316	0,000 250	0,999 685
012.jpg	20 859	55 162	0	11	0,999 855	0,999 473	1,000 000	0,999 736	0,999 684	0,004 205	0,000 316	0,999 604
013.jpg	30 405	57 813	0	0	1,000 000	1,000 000	1,000 000	1,000 000	0,999 864	0,001 933	0,000 136	0,999 849
014.jpg	26 904	48 371	0	1	0,999 987	0,999 963	1,000 000	0,999 981	0,999 920	0,001 269	0,000 080	0,999 913
015.jpg	40 789	80 116	0	22	0,999 818	0,999 461	1,000 000	0,999 730	0,999 554	0,005 886	0,000 446	0,999 501
016.jpg	52 357	121 987	0	6	0,999 966	0,999 885	1,000 000	0,999 943	0,999 874	0,001 926	0,000 126	0,999 850
017.jpg	53 061	95 229	0	0	0,999 906	1,000 000	1,000 000	1,000 000	0,999 757	0,003 250	0,000 243	0,999 736
018.jpg	37 016	77 783	0	0	0,999 939	1,000 000	1,000 000	1,000 000	0,999 791	0,002 837	0,000 209	0,999 761
019.jpg	15 783	148 622	0	0	0,999 970	1,000 000	1,000 000	1,000 000	0,999 903	0,001 363	0,000 097	0,999 720
020.jpg	10 591	105 535	0	11	0,999 905	0,998 962	1,000 000	0,999 481	0,999 759	0,003 234	0,000 241	0,999 273
021.jpg	19 805	62 451	0	0	1,000 000	1,000 000	1,000 000	1,000 000	0,999 878	0,001 729	0,000 122	0,999 834
022.jpg	51 133	109 117	0	40	0,999 750	0,999 218	1,000 000	0,999 609	0,999 439	0,007 099	0,000 561	0,999 354
023.jpg	49 915	140 333	0	12	0,999 937	0,999 760	1,000 000	0,999 880	0,999 758	0,003 450	0,000 242	0,999 688
024.jpg	36 639	99 086	0	3	0,999 978	0,999 918	1,000 000	0,999 959	0,999 882	0,001 801	0,000 118	0,999 850
025.jpg	7815	19 832	0	1	0,999 964	0,999 872	1,000 000	0,999 936	0,999 638	0,004 863	0,000 362	0,999 534
1.jpg	41 207	106 515	0	0	1,000 000	1,000 000	1,000 000	1,000 000	0,999 919	0,001 208	0,000 081	0,999 899

Nota: Sens. = Sensibilidad, Espec. = Especificidad, Exact. = Exactitud. Valores perfectos (1,000000) destacados en negra.

Tabla B.6: Resultados del método de segmentación PCA con el conjunto de datos Coffee diseases

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
01.jpg	30019	52010	1977	44	0.975955	0.998536	0.963380	0.980958	0.952997	0.284965	0.045245	0.949686
02.jpg	43510	37994	2539	7	0.969709	0.999839	0.937360	0.968599	0.941162	0.334283	0.056882	0.940869
03.jpg	27690	54081	2259	20	0.972885	0.999278	0.959904	0.979591	0.947150	0.306474	0.050256	0.941658
04.jpg	32010	43904	8119	17	0.903200	0.999469	0.843934	0.921702	0.825120	0.738747	0.154525	0.820479
05.jpg	23867	51806	8374	3	0.900333	0.999874	0.860851	0.930363	0.820531	0.734764	0.147579	0.800504
06.jpg	21465	58380	4198	7	0.949970	0.999674	0.932916	0.966295	0.904881	0.463052	0.083767	0.884420
07.jpg	31910	51793	293	54	0.995872	0.998311	0.994375	0.996343	0.991683	0.074522	0.008286	0.991194
08.jpg	38436	44436	1100	78	0.985985	0.997975	0.975843	0.986909	0.972200	0.195546	0.027457	0.972056
09.jpg	30601	51272	2151	26	0.974099	0.999151	0.959736	0.979444	0.949426	0.298927	0.048544	0.946305
10.jpg	27323	48955	7714	58	0.907531	0.997882	0.863876	0.930879	0.832026	0.719235	0.144616	0.820615
11.jpg	31650	49364	2924	112	0.963879	0.996474	0.944079	0.970276	0.930322	0.392064	0.066392	0.927220
12.jpg	25689	53203	5082	76	0.938632	0.997050	0.912808	0.954929	0.884732	0.551775	0.102810	0.872131
13.jpg	22096	48727	13197	30	0.842629	0.998644	0.786884	0.892764	0.734754	0.959392	0.197330	0.705981
14.jpg	21652	54088	8195	115	0.901130	0.994717	0.868423	0.931570	0.821713	0.744078	0.144246	0.794618
15.jpg	37120	44535	2326	69	0.971505	0.998145	0.950364	0.974254	0.944521	0.328533	0.053829	0.944065
16.jpg	33013	49618	1280	139	0.983117	0.995807	0.974852	0.985329	0.966666	0.228976	0.032752	0.965348
17.jpg	30345	51772	1874	59	0.977002	0.998059	0.965067	0.981563	0.954925	0.278050	0.043527	0.951914
18.jpg	33334	47551	3104	61	0.962344	0.998173	0.938723	0.968448	0.927391	0.398751	0.069137	0.925302
19.jpg	34748	47551	1608	143	0.979167	0.995902	0.967290	0.981596	0.959111	0.267609	0.040048	0.958161
20.jpg	19975	61495	2236	344	0.969304	0.983070	0.964915	0.973993	0.940425	0.357535	0.056047	0.922585
21.jpg	27829	54656	1428	137	0.981380	0.995101	0.974538	0.984820	0.963385	0.243642	0.035635	0.959383
22.jpg	30326	49861	3405	458	0.954039	0.985122	0.936076	0.960599	0.912259	0.492504	0.083681	0.907506
23.jpg	27617	56089	299	45	0.995907	0.998373	0.994697	0.996535	0.991801	0.072742	0.008155	0.990747
24.jpg	36516	43025	4478	31	0.946353	0.999152	0.905732	0.952442	0.898376	0.506326	0.095727	0.897530
25.jpg	27170	56542	304	34	0.995979	0.998750	0.994652	0.996701	0.991777	0.072085	0.008172	0.990643
26.jpg	23969	58218	1749	114	0.977835	0.995266	0.970834	0.983050	0.956492	0.272699	0.041639	0.948244
27.jpg	31640	51787	576	47	0.992588	0.998517	0.989000	0.993758	0.985145	0.117399	0.014716	0.984252
28.jpg	24465	59259	258	68	0.996121	0.997228	0.995665	0.996447	0.992273	0.070233	0.007691	0.990689
29.jpg	34698	48592	655	105	0.990958	0.996983	0.986700	0.991841	0.982032	0.141126	0.017836	0.981535
30.jpg	34491	46513	2935	111	0.963760	0.996792	0.940645	0.968718	0.930079	0.394254	0.067058	0.928627
31.jpg	34797	48668	413	172	0.993040	0.995081	0.991585	0.993333	0.986012	0.118856	0.013953	0.985625
32.jpg	25670	58073	250	57	0.996347	0.997784	0.995714	0.996749	0.992532	0.067959	0.007433	0.991239
33.jpg	27544	53723	2741	42	0.966889	0.998477	0.951456	0.974967	0.935903	0.357651	0.060377	0.929416
34.jpg	26196	57319	497	38	0.993635	0.998551	0.991404	0.994978	0.987209	0.102744	0.012647	0.985206
35.jpg	28753	52792	2478	27	0.970196	0.999062	0.955166	0.977114	0.941967	0.330733	0.055109	0.937058
36.jpg	25160	53027	5814	49	0.930244	0.998056	0.901191	0.949624	0.870157	0.595410	0.113587	0.855655
37.jpg	24648	58696	698	8	0.991600	0.999676	0.988248	0.993962	0.983178	0.123563	0.016499	0.979938
38.jpg	31279	52548	86	137	0.997347	0.995639	0.998366	0.997003	0.994660	0.053024	0.005336	0.994297
39.jpg	25265	56434	2329	22	0.972029	0.999130	0.960366	0.979748	0.945464	0.312344	0.051404	0.937254
40.jpg	28595	53337	2087	31	0.974801	0.998917	0.962345	0.980631	0.950668	0.293050	0.047205	0.946198
41.jpg	32307	47248	4396	99	0.946520	0.996945	0.914879	0.955912	0.898631	0.513103	0.094535	0.895360
42.jpg	31207	50379	2347	117	0.970684	0.996265	0.955487	0.975876	0.942997	0.338559	0.054801	0.940001
43.jpg	27660	53092	3274	24	0.960761	0.999133	0.941915	0.970524	0.924514	0.401028	0.070306	0.917394
44.jpg	24230	57268	2518	34	0.969637	0.998599	0.957883	0.978241	0.941073	0.331964	0.055123	0.930972
45.jpg	31392	52365	222	71	0.996514	0.997743	0.995778	0.996761	0.992934	0.066132	0.007050	0.992468
46.jpg	19393	63957	689	11	0.991672	0.999433	0.989342	0.994388	0.983412	0.120808	0.016161	0.977070
47.jpg	24723	58777	459	91	0.993456	0.996333	0.992251	0.994292	0.986928	0.107887	0.012953	0.984399
48.jpg	21553	62225	220	52	0.996764	0.997593	0.996477	0.997035	0.993454	0.060401	0.006512	0.991469
49.jpg	20869	61413	1744	24	0.978965	0.998851	0.972386	0.985619	0.958678	0.249893	0.038990	0.946779
50.jpg	25855	55518	2635	42	0.968150	0.998378	0.954688	0.976533	0.938172	0.346781	0.058036	0.929907
51.jpg	17833	49042	17159	16	0.795657	0.999104	0.740805	0.869954	0.674766	1.071208	0.208467	0.624932
52.jpg	21595	62018	393	44	0.994801	0.997967	0.993703	0.995835	0.989585	0.086877	0.010300	0.986492
53.jpg	19280	63948	765	57	0.990220	0.997052	0.988179	0.992615	0.980585	0.141878	0.018908	0.973144
54.jpg	18016	64462	1526	46	0.981297	0.997453	0.976875	0.987164	0.963178	0.229038	0.034670	0.947806
55.jpg	31544	48380	4063	63	0.950910	0.998007	0.922525	0.960266	0.906553	0.478715	0.087219	0.902771
56.jpg	22066	58664	3281	39	0.960500	0.998236	0.947034	0.972635	0.924031	0.398724	0.068965	0.907719
57.jpg	25708	53725	4583	34	0.945068	0.998679	0.921400	0.960040	0.896107	0.506072	0.093413	0.884233
58.jpg	24634	56704	2670	42	0.967733	0.998298	0.955031	0.976664	0.937526	0.348328	0.058339	0.927565
59.jpg	28723	50507	4713	107	0.942653	0.996289	0.914650	0.955470	0.891777	0.535465	0.098966	0.884477
60.jpg	25979	57745	285	41	0.996121	0.998424	0.995089	0.996757	0.992084	0.070277	0.007869	0.990776
61.jpg	32398	50819	800	33	0.990089	0.998982	0.984502	0.991742	0.980258	0.145438	0.019476	0.979280
62.jpg	28296	55469	243	42	0.996609	0.998518	0.995638	0.997078	0.993123	0.063054	0.006849	0.992326
63.jpg	40465	43303	155	127	0.996645	0.996871	0.996433	0.996652	0.993265	0.065150	0.006735	0.993257
64.jpg	28878	54483	677	12	0.991802	0.999585	0.987727	0.993656	0.983646	0.122362	0.016117	0.982001
65.jpg	32956	49483	1407	204	0.980833	0.993848	0.972352	0.983100	0.962354	0.255515	0.036985	0.960884
66.jpg	26737	57057	217	39	0.996954	0.998543	0.996211	0.997377	0.993903	0.056924	0.006073	0.992997
67.jpg	29455	53863	667	65	0.991291	0.997798	0.987768	0.992783	0.982570	0.134277	0.017224	0.980992
68.jpg	28385	55375	251	39	0.996550	0.998628	0.995488	0.997058	0.993005	0.063743	0.006965	0.992207
69.jpg	31309	52524	173	44	0.997418	0.998597	0.996717	0.997657	0.994755	0.050932	0.005234	0.994400
70.jpg	28429	55406	160	55	0.997442	0.998069	0.997121	0.997595	0.994779	0.051049	0.005211	0.994182
71.jpg	27919	55836	203	92	0.996490	0.996716	0.996378	0.996547	0.992958	0.066358	0.007028	0.992091
72.jpg	27372	59599	696	23	0.991446	0.999032	0.988457	0.993744	0.982991	0.126368	0.016682	0.979280
73.jpg	23662	60152	211	25	0.997192	0.998945	0.996504	0.997725	0.994258	0.053039	0.005711	0.992933
74.jpg	30859	52913	200	78	0.996692	0.997479	0.996234	0.996857	0.993265	0.063838	0.006723	0.992770
75.jpg	24056	59725	215	54	0.996800	0.997760	0.996413	0.997087	0.993525	0.060293	0.006448	0.992113
76.jpg	31838	51890	277	45	0.996169	0.998589	0.994690	0.996639	0.992249	0.069896	0.007721	0.991785
77.jpg	23392	60292	331	35	0.995645	0.998506	0.994540	0.996523	0.991140	0.076065	0.008784	0.989047

Tabla B.7: Resultados del método de segmentación Otsu con el conjunto de datos Coffee diseases

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
01.jpg	29941	52966	1021	122	0.986401	0.995942	0.981088	0.988515	0.973102	0.192741	0.026457	0.970977
02.jpg	43212	38191	2342	305	0.968507	0.992991	0.942220	0.967606	0.938907	0.374211	0.059800	0.938642
03.jpg	27653	54944	1396	57	0.982713	0.997943	0.975222	0.986582	0.965931	0.223569	0.033063	0.962054
04.jpg	31914	46142	5881	113	0.928685	0.996472	0.886954	0.941713	0.867520	0.622082	0.120864	0.863434
05.jpg	23862	53014	7166	8	0.914646	0.999665	0.880924	0.940294	0.843861	0.667523	0.131326	0.824828
06.jpg	21449	59224	3354	23	0.959822	0.998929	0.946403	0.972666	0.922805	0.399979	0.069616	0.904931
07.jpg	31829	51864	222	135	0.995753	0.995776	0.995738	0.995757	0.991447	0.079151	0.008544	0.990936
08.jpg	38355	44973	563	159	0.991410	0.995872	0.987636	0.991754	0.982850	0.139126	0.017088	0.982744
09.jpg	30557	51801	1622	70	0.979869	0.997714	0.969639	0.983676	0.960434	0.253213	0.038421	0.957829
10.jpg	27268	48606	8063	113	0.902725	0.995873	0.857718	0.926795	0.824238	0.750316	0.150850	0.812517
11.jpg	31554	50247	2041	208	0.973242	0.993451	0.960966	0.977209	0.947916	0.324640	0.050542	0.945312
12.jpg	25639	54546	3739	126	0.954015	0.995110	0.935850	0.965480	0.912194	0.459753	0.080692	0.901341
13.jpg	22068	48323	13601	58	0.837490	0.997379	0.780360	0.888869	0.727763	0.981206	0.201623	0.698616
14.jpg	21615	53725	8558	152	0.896371	0.993017	0.862595	0.927806	0.814124	0.770448	0.149540	0.786520
15.jpg	37007	44642	2219	182	0.971434	0.995106	0.952647	0.973877	0.944387	0.340296	0.054234	0.943904
16.jpg	32880	49629	1269	272	0.981666	0.991795	0.975068	0.983432	0.963957	0.251205	0.035552	0.962509
17.jpg	30290	51957	1689	114	0.978548	0.996250	0.968516	0.982383	0.957880	0.269738	0.040902	0.954993
18.jpg	33243	47644	3011	152	0.962368	0.995448	0.940559	0.968004	0.927479	0.408619	0.069378	0.925339
19.jpg	32297	47850	1309	2594	0.953563	0.925654	0.973372	0.949513	0.911352	0.533111	0.087547	0.908291
20.jpg	17582	62293	1438	2737	0.950327	0.865298	0.977436	0.921367	0.905567	0.538223	0.089828	0.872614
21.jpg	27374	54974	1110	592	0.979750	0.978831	0.980208	0.979520	0.960251	0.279932	0.039385	0.955639
22.jpg	28443	50085	3181	2341	0.934301	0.923954	0.940281	0.932117	0.877233	0.690292	0.121294	0.869057
23.jpg	27548	55915	473	114	0.993016	0.995879	0.991612	0.993745	0.986083	0.115173	0.013818	0.984321
24.jpg	36433	43626	3877	114	0.952516	0.996881	0.918384	0.957632	0.909455	0.476736	0.086164	0.908641
25.jpg	27114	56630	216	90	0.996359	0.996692	0.966200	0.996446	0.992532	0.069469	0.007448	0.991490
26.jpg	23889	58071	1896	194	0.975134	0.991945	0.968383	0.980164	0.951391	0.302822	0.046498	0.942268
27.jpg	31576	51967	396	111	0.993968	0.996497	0.992437	0.994467	0.987867	0.103825	0.012081	0.987118
28.jpg	24418	59381	136	115	0.997014	0.995312	0.997715	0.996514	0.994045	0.057847	0.005949	0.992808
29.jpg	34586	48763	484	217	0.991660	0.993765	0.990172	0.991968	0.983412	0.137334	0.016545	0.982933
30.jpg	34414	46724	2724	188	0.965354	0.994567	0.944912	0.969739	0.933041	0.389495	0.064581	0.931590
31.jpg	34638	48761	320	331	0.992255	0.990534	0.993480	0.992007	0.984559	0.131065	0.015437	0.984116
32.jpg	25609	58150	173	118	0.996538	0.995413	0.997034	0.996224	0.992958	0.066676	0.007032	0.991725
33.jpg	27492	52473	3991	94	0.951398	0.996592	0.929318	0.962955	0.907455	0.475820	0.085215	0.899131
34.jpg	26159	57383	433	75	0.993956	0.997141	0.992511	0.994826	0.987844	0.101342	0.012056	0.985924
35.jpg	28706	53266	2004	74	0.975277	0.997429	0.963742	0.980585	0.951572	0.294400	0.046519	0.947251
36.jpg	25112	53556	5285	97	0.935967	0.996152	0.910182	0.953167	0.880070	0.569470	0.106244	0.866023
37.jpg	24613	59071	323	43	0.995645	0.998256	0.994562	0.996409	0.991164	0.076685	0.008771	0.989396
38.jpg	31155	52586	48	261	0.996324	0.991692	0.999088	0.995390	0.992627	0.067395	0.007348	0.992118
39.jpg	25200	56646	2117	87	0.973778	0.996559	0.963974	0.980267	0.948772	0.306096	0.048679	0.940876
40.jpg	28561	53736	1688	65	0.979143	0.997729	0.969544	0.983637	0.958951	0.258950	0.039660	0.955063
41.jpg	32226	47716	3928	180	0.951124	0.994445	0.923941	0.959193	0.906896	0.491709	0.087692	0.903699
42.jpg	31028	50628	2098	296	0.971517	0.990550	0.960209	0.975380	0.944611	0.345705	0.053802	0.941561
43.jpg	27624	55048	1318	60	0.983605	0.997833	0.976617	0.987225	0.967655	0.215179	0.031447	0.963929
44.jpg	24182	57976	1810	82	0.977490	0.996621	0.969725	0.983173	0.955947	0.272259	0.042060	0.947863
45.jpg	31291	52467	120	172	0.996526	0.994533	0.997718	0.996126	0.992958	0.067274	0.007038	0.992484
46.jpg	19375	64071	575	29	0.992814	0.998505	0.991105	0.994805	0.985660	0.109655	0.014045	0.980117
47.jpg	24676	58860	376	138	0.993885	0.994439	0.993653	0.994046	0.987773	0.104411	0.012157	0.985384
48.jpg	21485	62270	175	120	0.996490	0.994446	0.997198	0.995822	0.992958	0.066162	0.007027	0.990808
49.jpg	20823	61439	1718	70	0.978727	0.996650	0.972798	0.984724	0.958222	0.257641	0.039549	0.946143
50.jpg	25798	55977	2176	99	0.972933	0.996177	0.962581	0.979379	0.947173	0.314922	0.050243	0.939771
51.jpg	17825	32320	33881	24	0.596609	0.998655	0.488210	0.743433	0.518643	1.365516	0.278472	0.416252
52.jpg	21513	61969	442	126	0.993242	0.994177	0.992918	0.993548	0.986505	0.111566	0.013366	0.982505
53.jpg	19186	64177	536	151	0.991826	0.992191	0.991717	0.991954	0.983739	0.129236	0.016037	0.977354
54.jpg	17949	64646	1342	113	0.982689	0.993744	0.979663	0.986703	0.965862	0.222730	0.032498	0.951336
55.jpg	31425	48633	3810	182	0.952504	0.994242	0.927350	0.960796	0.909519	0.481247	0.085162	0.905712
56.jpg	22013	59206	2739	92	0.966318	0.995838	0.955783	0.975811	0.934815	0.362766	0.060223	0.920096
57.jpg	25668	54143	4165	74	0.949566	0.997125	0.928569	0.962847	0.904153	0.483130	0.087081	0.892752
58.jpg	24593	56829	2545	83	0.968733	0.996636	0.957136	0.976886	0.939398	0.345644	0.056871	0.929602
59.jpg	28633	51674	3546	197	0.955467	0.993167	0.935784	0.964475	0.914791	0.460463	0.079843	0.908304
60.jpg	25870	57848	182	150	0.996050	0.994235	0.996864	0.995549	0.991942	0.074966	0.008048	0.990590
61.jpg	32330	50916	703	101	0.990434	0.996886	0.986381	0.991633	0.980935	0.147177	0.018885	0.979972
62.jpg	28245	55543	169	93	0.996883	0.996718	0.969677	0.996842	0.993667	0.060933	0.006324	0.992925
63.jpg	40322	43348	110	270	0.995479	0.993348	0.997469	0.995409	0.990999	0.082281	0.008993	0.990987
64.jpg	28855	54544	616	35	0.992255	0.998789	0.988832	0.993810	0.984535	0.119623	0.015275	0.982967
65.jpg	32394	49758	1132	766	0.977418	0.976900	0.977756	0.977328	0.955856	0.308730	0.043976	0.953953
66.jpg	26696	57142	132	80	0.997478	0.997012	0.997695	0.997354	0.994944	0.050270	0.005050	0.994185
67.jpg	29396	53944	586	124	0.991553	0.995799	0.989254	0.992527	0.983131	0.134646	0.016729	0.981584
68.jpg	28322	55468	158	102	0.996907	0.996411	0.997160	0.996786	0.993714	0.060733	0.006279	0.992987
69.jpg	31249	52591	106	104	0.997501	0.996683	0.997989	0.997336	0.994921	0.050971	0.005078	0.994572
70.jpg	28370	55489	77	114	0.997728	0.995998	0.98614	0.997306	0.995347	0.047042	0.004650	0.994809
71.jpg	27814	55894	145	197	0.995931	0.992967	0.997413	0.995190	0.991848	0.075902	0.008144	0.990833
72.jpg	23710	59713	582	45	0.992540	0.998106	0.990347	0.994227	0.985145	0.115673	0.014632	0.981865
73.jpg	23612	60234	129	75	0.997573	0.996834	0.997863	0.997348	0.995015	0.049339	0.004977	0.993854
74.jpg	30795	53007	106	142	0.997049	0.995410	0.998004	0.996707	0.994069	0.058126	0.005928	0.993626
75.jpg	23999	59797	143	111	0.996978	0.995396	0.997614	0.996505	0.993880	0.059116	0.006113	0.992533
76.jpg	31786	51993	174	97	0.996776	0.996665	0.996665	0.996811	0.993454	0.062938	0.006539	0.993055
77.jpg	23362	60355	268	65	0.996038	0.997225	0.995579	0.996402	0.991919	0.072485	0.008036	0.989995

Tabla B.8: Resultados del método de segmentación GA con el conjunto de datos

Coffee diseases

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
01.jpg	29907	53627	360	156	0.993861	0.994811	0.993332	0.994071	0.987726	0.106209	0.012235	0.986678
02.jpg	42808	40086	447	709	0.986246	0.983708	0.988972	0.986340	0.972778	0.209103	0.027198	0.972745
03.jpg	27351	56122	218	359	0.993135	0.987044	0.996131	0.991588	0.986317	0.116946	0.013647	0.984528
04.jpg	31793	50167	1856	234	0.975134	0.992694	0.964323	0.978509	0.951526	0.309914	0.047247	0.949185
05.jpg	23578	59555	625	292	0.989090	0.987767	0.989614	0.988691	0.978417	0.168216	0.021402	0.973700
06.jpg	20975	62452	126	497	0.992588	0.976854	0.997987	0.987420	0.985215	0.119756	0.014616	0.980546
07.jpg	31470	51884	202	494	0.991719	0.984545	0.996122	0.990333	0.983482	0.136176	0.016456	0.982463
08.jpg	37946	45363	173	568	0.991184	0.985252	0.996201	0.990726	0.982359	0.143112	0.017583	0.982222
09.jpg	29902	53310	113	725	0.990030	0.976328	0.997885	0.987106	0.980235	0.151864	0.019552	0.978606
10.jpg	26697	56283	386	684	0.987269	0.975019	0.993189	0.984104	0.974886	0.192821	0.024972	0.971429
11.jpg	30894	51950	338	868	0.985651	0.972672	0.993536	0.983104	0.971668	0.211703	0.028128	0.969821
12.jpg	24709	58053	232	1056	0.984676	0.959014	0.996020	0.977517	0.969798	0.215368	0.029622	0.964347
13.jpg	20903	61802	122	1223	0.983998	0.944726	0.998030	0.971378	0.968507	0.212814	0.030393	0.959122
14.jpg	20392	62078	205	1375	0.981202	0.936831	0.996709	0.966770	0.963178	0.244882	0.035454	0.951721
15.jpg	36988	46651	210	201	0.995110	0.994595	0.995519	0.995057	0.990150	0.089953	0.009849	0.990019
16.jpg	31862	50812	86	1290	0.983629	0.961088	0.998310	0.979699	0.967747	0.219060	0.031595	0.966068
17.jpg	30137	53494	152	267	0.995015	0.991218	0.997167	0.994192	0.989985	0.090257	0.009999	0.989155
18.jpg	33040	50459	196	355	0.993444	0.989370	0.996131	0.992750	0.986928	0.113144	0.013051	0.986348
19.jpg	34259	49008	151	632	0.990684	0.981886	0.996928	0.989407	0.981495	0.147098	0.018397	0.980916
20.jpg	19238	63404	327	1081	0.983248	0.946799	0.994869	0.970834	0.967034	0.230670	0.032121	0.955006
21.jpg	25669	55903	181	2297	0.970518	0.917865	0.996773	0.957319	0.942795	0.341890	0.054448	0.934781
22.jpg	30075	52696	570	709	0.984783	0.976969	0.989299	0.983134	0.969982	0.225697	0.029947	0.967706
23.jpg	27235	56310	78	427	0.993992	0.984564	0.998617	0.991590	0.988008	0.100739	0.011905	0.986400
24.jpg	36286	47317	186	261	0.994682	0.992859	0.996084	0.994471	0.989326	0.096052	0.010669	0.989140
25.jpg	26856	56651	195	348	0.993540	0.987208	0.996570	0.991889	0.986998	0.111850	0.012965	0.985155
26.jpg	23811	59797	170	272	0.994741	0.988706	0.997165	0.992935	0.989420	0.093751	0.010552	0.987074
27.jpg	31296	52164	199	391	0.992980	0.987661	0.996200	0.991930	0.985918	0.119983	0.014048	0.985006
28.jpg	24278	59426	91	255	0.995883	0.989606	0.998471	0.994038	0.991801	0.074567	0.008167	0.990080
29.jpg	33687	49098	149	1116	0.984949	0.967934	0.996974	0.982454	0.970352	0.210484	0.029240	0.969345
30.jpg	34391	49253	195	211	0.995170	0.993902	0.996056	0.994979	0.990315	0.088549	0.009683	0.990005
31.jpg	34761	48947	134	208	0.995931	0.994052	0.997270	0.995661	0.991730	0.077266	0.008265	0.991489
32.jpg	25354	58232	91	373	0.994479	0.985502	0.998440	0.991971	0.988926	0.095154	0.011005	0.986950
33.jpg	27125	56384	80	461	0.993563	0.983289	0.998583	0.990936	0.987186	0.106229	0.012712	0.985444
34.jpg	25615	57770	46	619	0.992088	0.976405	0.999204	0.987805	0.984207	0.122393	0.015572	0.981547
35.jpg	28526	55140	130	254	0.995431	0.991174	0.997648	0.994411	0.990739	0.084179	0.009243	0.989717
36.jpg	24977	58673	168	232	0.995241	0.990797	0.997145	0.993971	0.990456	0.086306	0.009527	0.988649
37.jpg	24308	59304	90	348	0.994789	0.985886	0.998485	0.992185	0.989561	0.090543	0.010375	0.987398
38.jpg	31227	52491	143	189	0.996050	0.993984	0.997283	0.995634	0.992084	0.074348	0.007911	0.991547
39.jpg	25154	58667	96	133	0.997275	0.994740	0.998366	0.996553	0.994447	0.054524	0.005548	0.993404
40.jpg	28379	55210	214	247	0.994515	0.991371	0.996139	0.993755	0.988879	0.098878	0.011112	0.987631
41.jpg	32032	51434	210	374	0.993052	0.988459	0.995934	0.992196	0.986059	0.119337	0.013915	0.985285
42.jpg	31155	52483	243	169	0.995098	0.994605	0.995391	0.994998	0.990150	0.089327	0.009840	0.989479
43.jpg	26760	56328	38	924	0.988554	0.966623	0.999326	0.982975	0.977417	0.160889	0.022114	0.974305
44.jpg	24125	59583	203	139	0.995931	0.994271	0.996605	0.995438	0.991848	0.075339	0.008137	0.990100
45.jpg	30986	52478	109	477	0.993028	0.984839	0.997927	0.991383	0.986036	0.116019	0.013885	0.985072
46.jpg	18959	64581	65	445	0.993932	0.977067	0.998995	0.988031	0.987867	0.098981	0.011962	0.982885
47.jpg	24444	59042	194	370	0.993290	0.985089	0.996725	0.990907	0.986552	0.114326	0.013392	0.983851
48.jpg	21345	62378	67	260	0.996109	0.987966	0.998927	0.993446	0.992155	0.070745	0.007800	0.989724
49.jpg	19837	63091	66	1056	0.986651	0.949457	0.998955	0.974206	0.973565	0.181631	0.025524	0.964370
50.jpg	25762	57921	232	135	0.995634	0.994787	0.996011	0.995399	0.991140	0.080812	0.008840	0.989635
51.jpg	16797	66103	98	1052	0.986318	0.941061	0.998520	0.969790	0.973102	0.184108	0.025802	0.959545
52.jpg	21394	62393	18	245	0.996871	0.988678	0.999712	0.994195	0.993690	0.056636	0.006263	0.991736
53.jpg	18954	64580	133	383	0.993861	0.980193	0.997945	0.989069	0.987750	0.102993	0.012139	0.982719
54.jpg	17586	65881	107	476	0.993064	0.973646	0.998378	0.986012	0.986200	0.111396	0.013596	0.979534
55.jpg	31439	52200	243	168	0.995110	0.994685	0.995366	0.995026	0.990174	0.089165	0.009816	0.989541
56.jpg	21779	61779	166	326	0.994146	0.985252	0.997320	0.991286	0.988267	0.101320	0.011677	0.984882
57.jpg	25621	58063	245	121	0.995645	0.995300	0.995798	0.995549	0.991258	0.079615	0.008718	0.989743
58.jpg	24507	59217	157	169	0.996121	0.993151	0.997356	0.995253	0.992249	0.072437	0.007744	0.990670
59.jpg	28472	54989	231	358	0.992992	0.987582	0.995817	0.991700	0.986012	0.119460	0.013958	0.984489
60.jpg	25561	57887	143	459	0.992838	0.982360	0.997536	0.989948	0.985684	0.118865	0.014220	0.983238
61.jpg	32057	51461	158	374	0.993670	0.988468	0.996939	0.992703	0.987303	0.109570	0.012663	0.986597
62.jpg	28003	55569	143	335	0.994313	0.988178	0.997433	0.992806	0.988573	0.099860	0.011390	0.987212
63.jpg	40039	43389	69	553	0.992600	0.986377	0.998412	0.992394	0.985309	0.118777	0.014619	0.985285
64.jpg	28162	54993	167	728	0.989352	0.974801	0.996972	0.985887	0.978837	0.162761	0.020943	0.976488
65.jpg	32454	50541	349	706	0.987448	0.978709	0.993142	0.985926	0.975164	0.191937	0.024737	0.973985
66.jpg	26490	57099	175	286	0.994515	0.989319	0.996945	0.993132	0.989067	0.096761	0.010908	0.987416
67.jpg	28940	54389	141	580	0.991422	0.980352	0.997414	0.988883	0.982874	0.137354	0.016996	0.981175
68.jpg	28160	55497	129	264	0.995324	0.990712	0.997681	0.994197	0.990574	0.085259	0.009405	0.989470
69.jpg	31090	52584	113	263	0.995526	0.991612	0.997856	0.994734	0.990999	0.082020	0.008984	0.990374
70.jpg	28405	55354	212	79	0.996538	0.997227	0.996185	0.996706	0.992981	0.065847	0.007002	0.992183
71.jpg	27779	55865	174	232	0.995170	0.991718	0.996895	0.994306	0.990291	0.088003	0.009697	0.989084
72.jpg	23193	60206	89	562	0.992255	0.976342	0.998524	0.987433	0.984582	0.122355	0.015219	0.980942
73.jpg	23410	60238	125	277	0.995217	0.988306	0.997929	0.993118	0.990386	0.085821	0.009578	0.988127
74.jpg	30736	52948	165	201	0.995645	0.993503	0.996893	0.995198	0.991187	0.081456	0.008808	0.990531
75.jpg	23927	59732	208	183	0.995348	0.992410	0.996530	0.994470	0.990645	0.084813	0.009342	0.988594
76.jpg	31672	51969	198	211	0.995134	0.993382	0.996204	0.994793	0.990197	0.089260	0.009799	0.989596
77.jpg	22961	60562	61	466	0.993730	0.980108	0.998994	0.989551	0.987350	0.103488	0.012508	0.984233

Tabla B.9: Resultados del método de segmentación PSO con el conjunto de datos Coffee diseases

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
01.jpg	29746	53899	88	317	0.995181	0.989455	0.998370	0.993913	0.990339	0.085727	0.009625	0.989478
02.jpg	40797	40304	229	2720	0.964914	0.937496	0.994350	0.965923	0.932245	0.397747	0.066100	0.932160
03.jpg	27146	56307	33	564	0.992897	0.979646	0.999414	0.989530	0.985848	0.111308	0.013981	0.983941
04.jpg	30743	50909	1114	1284	0.971469	0.959909	0.978586	0.969248	0.944544	0.371344	0.055291	0.941305
05.jpg	22204	60068	112	1666	0.978846	0.930205	0.998139	0.964172	0.958586	0.261315	0.039536	0.948556
06.jpg	20877	62556	22	595	0.992659	0.972289	0.999648	0.985969	0.985356	0.111601	0.014360	0.980670
07.jpg	31505	52017	69	459	0.993718	0.985640	0.998675	0.992158	0.987421	0.104775	0.012500	0.986634
08.jpg	37794	45468	68	720	0.990625	0.981305	0.998507	0.989906	0.981261	0.143946	0.018594	0.981105
09.jpg	29703	53372	51	924	0.988400	0.969831	0.999045	0.984438	0.977092	0.164953	0.022516	0.975163
10.jpg	26420	56474	195	961	0.986246	0.964903	0.996559	0.980731	0.972940	0.197126	0.026625	0.969087
11.jpg	30758	52138	150	1004	0.986270	0.968390	0.997131	0.982761	0.972871	0.196267	0.026742	0.971042
12.jpg	24379	58255	30	1386	0.983153	0.946206	0.999485	0.972846	0.966896	0.214071	0.031901	0.960714
13.jpg	20279	61883	41	1847	0.977537	0.916524	0.999338	0.957931	0.956083	0.263623	0.041260	0.942610
14.jpg	19620	62231	52	2147	0.973837	0.901364	0.999165	0.950265	0.949110	0.295130	0.047232	0.932626
15.jpg	37048	46691	170	141	0.996300	0.996209	0.996372	0.996290	0.992509	0.071262	0.007490	0.992410
16.jpg	32771	50850	48	381	0.994896	0.988507	0.999057	0.993782	0.989750	0.088030	0.010198	0.989257
17.jpg	30185	53505	141	219	0.995717	0.992797	0.997372	0.995084	0.991329	0.080096	0.008662	0.990612
18.jpg	33270	50481	174	125	0.996443	0.996257	0.996565	0.996411	0.992769	0.068888	0.007227	0.992454
19.jpg	34718	48909	250	173	0.994967	0.995042	0.994914	0.994978	0.989844	0.091926	0.010149	0.989548
20.jpg	19834	63505	226	485	0.991541	0.976131	0.996454	0.986292	0.983154	0.135795	0.016695	0.977064
21.jpg	25727	55945	139	2239	0.971707	0.919938	0.997522	0.958730	0.944948	0.328978	0.052386	0.937229
22.jpg	30444	52782	484	340	0.990196	0.988955	0.990914	0.989934	0.980538	0.157804	0.019421	0.979088
23.jpg	27268	56368	20	394	0.995074	0.985757	0.999645	0.992701	0.990197	0.081812	0.009718	0.988876
24.jpg	36239	47386	117	308	0.994943	0.991572	0.997537	0.994555	0.989844	0.090845	0.010139	0.989664
25.jpg	26814	56813	33	390	0.994967	0.985664	0.999419	0.992542	0.989773	0.086421	0.010146	0.988296
26.jpg	23754	59948	19	329	0.995860	0.986339	0.999683	0.993011	0.991588	0.072300	0.008340	0.989692
27.jpg	31453	52251	112	234	0.995883	0.992615	0.997861	0.995238	0.991659	0.077176	0.008329	0.991119
28.jpg	24328	59473	44	205	0.997037	0.991644	0.999261	0.995452	0.994092	0.055543	0.005884	0.992850
29.jpg	34270	49158	89	533	0.992600	0.984685	0.998193	0.991439	0.985309	0.119761	0.014603	0.984839
30.jpg	34383	49390	58	219	0.996704	0.993671	0.998827	0.996249	0.993359	0.062653	0.006628	0.993142
31.jpg	34690	49038	43	279	0.996169	0.992022	0.999124	0.995573	0.992202	0.070706	0.007774	0.991969
32.jpg	25367	58284	39	360	0.995253	0.986007	0.999331	0.992669	0.990362	0.082583	0.009564	0.988635
33.jpg	27241	56423	41	345	0.995407	0.987494	0.999274	0.993384	0.990786	0.079726	0.009153	0.989536
34.jpg	25629	57802	14	605	0.992635	0.976938	0.999758	0.988348	0.985285	0.112671	0.014493	0.982800
35.jpg	28421	55218	52	359	0.995110	0.987526	0.999059	0.993293	0.990103	0.085623	0.009839	0.988994
36.jpg	24870	58793	48	339	0.995396	0.986552	0.999184	0.992868	0.990810	0.079640	0.009124	0.989041
37.jpg	24186	59364	30	470	0.994051	0.980938	0.999495	0.990216	0.988149	0.095975	0.011710	0.985667
38.jpg	31217	52585	49	199	0.997049	0.993666	0.999069	0.996367	0.994069	0.056609	0.005918	0.993661
39.jpg	25096	58715	48	191	0.997156	0.992447	0.999183	0.995815	0.994163	0.055599	0.005818	0.993059
40.jpg	28333	55371	53	293	0.995883	0.989765	0.999044	0.994404	0.991588	0.075328	0.008374	0.990626
41.jpg	31972	51573	71	434	0.993992	0.986607	0.998625	0.992616	0.987914	0.101814	0.012018	0.987227
42.jpg	31102	52631	95	222	0.996228	0.992913	0.998198	0.995556	0.992391	0.071151	0.007596	0.991861
43.jpg	24829	56332	34	2855	0.965628	0.896872	0.999397	0.948134	0.933573	0.364049	0.061790	0.923441
44.jpg	23996	59712	74	268	0.995931	0.988955	0.998762	0.993859	0.991848	0.073598	0.008113	0.990069
45.jpg	31092	52535	52	371	0.994967	0.988208	0.999011	0.993610	0.989867	0.087340	0.010079	0.989169
46.jpg	18983	64610	36	421	0.994563	0.978303	0.999443	0.988873	0.989114	0.088895	0.010728	0.984634
47.jpg	24378	59206	30	436	0.994456	0.982429	0.999494	0.990961	0.988855	0.091528	0.011025	0.986575
48.jpg	21257	62426	19	348	0.995634	0.983893	0.999696	0.991794	0.991211	0.074180	0.008694	0.988468
49.jpg	19814	63134	23	1079	0.986889	0.948356	0.999636	0.973996	0.974028	0.174662	0.024993	0.964951
50.jpg	25710	58022	131	187	0.996217	0.992779	0.997747	0.995263	0.992296	0.072094	0.007694	0.990972
51.jpg	16995	66114	87	854	0.988804	0.952154	0.998686	0.975420	0.977859	0.158295	0.021421	0.966748
52.jpg	21405	62410	1	234	0.997204	0.989186	0.999984	0.994585	0.994353	0.050028	0.005603	0.992601
53.jpg	18969	64620	93	368	0.994515	0.980969	0.998563	0.989766	0.989043	0.092799	0.010847	0.984528
54.jpg	17567	65967	21	495	0.993861	0.972594	0.999682	0.986138	0.987726	0.095942	0.012028	0.981746
55.jpg	31423	52286	157	184	0.995943	0.994179	0.997006	0.995592	0.991824	0.076503	0.008172	0.991292
56.jpg	21794	61880	65	311	0.995526	0.985931	0.998951	0.992441	0.990999	0.078908	0.008936	0.988378
57.jpg	25473	58266	42	269	0.996300	0.989550	0.999280	0.994415	0.992603	0.066687	0.007357	0.991286
58.jpg	24475	59322	52	201	0.996990	0.991854	0.999124	0.995489	0.993974	0.056899	0.006004	0.992732
59.jpg	28399	55094	126	431	0.993373	0.985050	0.997718	0.991384	0.986763	0.111581	0.013165	0.985299
60.jpg	25589	57953	77	431	0.993956	0.983436	0.998673	0.991054	0.987891	0.101393	0.012012	0.985811
61.jpg	32238	51381	238	193	0.994872	0.994049	0.995389	0.994719	0.989679	0.093182	0.010314	0.989120
62.jpg	28089	55634	78	249	0.996109	0.991213	0.998600	0.994907	0.992131	0.072313	0.007845	0.991192
63.jpg	40114	43447	11	478	0.994182	0.988224	0.999747	0.993986	0.988432	0.093467	0.011502	0.988413
64.jpg	28409	55109	51	481	0.993670	0.983351	0.999075	0.991213	0.987327	0.103745	0.012562	0.985924
65.jpg	32869	50663	227	291	0.993837	0.991224	0.995539	0.993382	0.987703	0.108046	0.012288	0.987133
66.jpg	26518	57181	93	258	0.995824	0.990365	0.998376	0.994370	0.991659	0.076001	0.008313	0.990391
67.jpg	29047	54454	76	473	0.993468	0.983977	0.998606	0.991292	0.986857	0.108738	0.013047	0.985553
68.jpg	28201	55550	76	223	0.996443	0.992155	0.998634	0.995394	0.992792	0.067325	0.007189	0.991945
69.jpg	31095	52662	35	258	0.996514	0.991771	0.999336	0.995553	0.992958	0.064297	0.007016	0.992465
70.jpg	28389	55449	117	95	0.997478	0.996665	0.997894	0.997280	0.994850	0.051367	0.005147	0.994257
71.jpg	27810	55910	129	201	0.996074	0.992824	0.997698	0.995261	0.992084	0.073907	0.007906	0.991096
72.jpg	23489	60226	69	266	0.996014	0.988802	0.998856	0.993829	0.992013	0.072111	0.007946	0.990145
73.jpg	23413	60281	82	274	0.995764	0.988432	0.998642	0.993537	0.991470	0.076674	0.008488	0.989459
74.jpg	30743	52977	136	194	0.996074	0.993729	0.997439	0.995584	0.992037	0.074648	0.007957	0.991442
75.jpg	23915	59806	134	195	0.996086	0.991912	0.997764	0.994838	0.992107	0.073315	0.007879	0.990364
76.jpg	31661	52050	117	222	0.995967	0.993037	0.997757	0.995397	0.991848	0.075855	0.008142	0.991342
77.jpg	23030	60596	27	397	0.994955	0.983054	0.999555	0.991304	0.989773	0.085221	0.010119	0.987251

Tabla B.10: Resultados del método de segmentación CP con el conjunto de datos Coffee diseases

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
01.jpg	29865	53919	68	198	0.996835	0.993414	0.998740	0.996077	0.993619	0.060804	0.006368	0.993054
02.jpg	42792	40518	15	725	0.991196	0.983340	0.999630	0.991485	0.982500	0.131185	0.017354	0.982486
03.jpg	27489	56314	26	221	0.997061	0.992025	0.999539	0.995782	0.994045	0.055347	0.005930	0.993257
04.jpg	31308	52020	3	719	0.991410	0.977550	0.999942	0.988746	0.982991	0.125266	0.016773	0.981909
05.jpg	23249	60154	26	621	0.992302	0.973984	0.999568	0.986776	0.984723	0.116261	0.015011	0.981142
06.jpg	21301	62507	71	171	0.997121	0.992036	0.998865	0.995451	0.994187	0.055664	0.005795	0.992362
07.jpg	31684	52000	86	280	0.995645	0.991240	0.998349	0.994795	0.991234	0.079456	0.008742	0.990695
08.jpg	38089	45524	12	425	0.994801	0.988965	0.999736	0.994351	0.989491	0.087225	0.010453	0.989408
09.jpg	30221	53415	8	406	0.995074	0.986744	0.999850	0.993297	0.990126	0.081647	0.009795	0.989321
10.jpg	27020	56644	25	361	0.995407	0.986816	0.999559	0.993187	0.990739	0.079023	0.009191	0.989439
11.jpg	31357	52225	63	405	0.994432	0.987249	0.998795	0.993022	0.988879	0.094581	0.011059	0.988156
12.jpg	25366	58120	165	399	0.993290	0.984514	0.997169	0.990841	0.986646	0.113091	0.013287	0.984291
13.jpg	21603	61865	59	523	0.993076	0.976363	0.999047	0.987705	0.986247	0.109476	0.013554	0.982223
14.jpg	21426	62155	128	341	0.994420	0.984334	0.997945	0.991139	0.988784	0.096681	0.011145	0.985391
15.jpg	36967	46715	146	222	0.995622	0.994030	0.996884	0.995457	0.991164	0.081828	0.008833	0.991044
16.jpg	32927	50886	12	225	0.997180	0.993213	0.999764	0.996489	0.994234	0.053333	0.005746	0.993960
17.jpg	30202	53543	103	202	0.996371	0.993356	0.998080	0.995718	0.992627	0.069525	0.007363	0.992015
18.jpg	33197	50608	47	198	0.997085	0.994071	0.999072	0.996572	0.994045	0.056983	0.005943	0.993780
19.jpg	34684	49125	34	207	0.997133	0.994067	0.999308	0.996688	0.994140	0.055656	0.005847	0.993962
20.jpg	20165	63656	75	154	0.997275	0.992421	0.998823	0.995622	0.994495	0.053204	0.005490	0.992496
21.jpg	27839	56016	68	127	0.997680	0.995459	0.998788	0.997123	0.995252	0.047670	0.004743	0.994654
22.jpg	30548	52951	315	236	0.993444	0.992334	0.994086	0.993210	0.986928	0.113387	0.013056	0.985941
23.jpg	27409	56359	29	253	0.996645	0.990854	0.999486	0.995170	0.993265	0.061207	0.006702	0.992365
24.jpg	36344	47469	34	203	0.997180	0.994446	0.999284	0.996865	0.994282	0.054500	0.005707	0.994180
25.jpg	26917	56795	51	287	0.995979	0.989450	0.999103	0.994276	0.991777	0.073716	0.008183	0.990598
26.jpg	23868	59869	98	215	0.996276	0.991073	0.998366	0.994719	0.992414	0.070385	0.007564	0.990725
27.jpg	31559	52316	47	128	0.997918	0.995960	0.999102	0.997531	0.995702	0.043618	0.004293	0.995424
28.jpg	24330	59489	28	203	0.997252	0.991725	0.999530	0.995627	0.994518	0.051356	0.005456	0.993363
29.jpg	34593	49160	87	210	0.996466	0.993966	0.998233	0.996100	0.992910	0.067106	0.007081	0.992692
30.jpg	34453	49384	64	149	0.997466	0.995694	0.998706	0.997200	0.994873	0.050963	0.005122	0.994707
31.jpg	34779	49022	59	190	0.997037	0.994567	0.998798	0.996682	0.993927	0.058494	0.006065	0.993748
32.jpg	25505	58231	92	222	0.996264	0.991371	0.998423	0.994897	0.992367	0.070822	0.007612	0.991015
33.jpg	27412	56429	35	174	0.997513	0.993692	0.999380	0.996536	0.994968	0.048728	0.005018	0.994291
34.jpg	25943	57780	36	291	0.996109	0.988908	0.999377	0.994142	0.992155	0.069769	0.007799	0.990852
35.jpg	28616	55234	36	164	0.997620	0.994302	0.999349	0.996825	0.995039	0.048579	0.004950	0.994489
36.jpg	25067	58737	104	142	0.997073	0.994367	0.998233	0.996300	0.994092	0.057465	0.005901	0.992971
37.jpg	24366	59371	23	290	0.996276	0.988238	0.999613	0.993925	0.992462	0.066518	0.007486	0.990893
38.jpg	31283	52587	47	133	0.997858	0.995766	0.999107	0.997437	0.995679	0.043676	0.004316	0.995383
39.jpg	25137	58714	49	150	0.997632	0.994068	0.999166	0.996617	0.995110	0.048195	0.004879	0.994187
40.jpg	28470	55349	75	156	0.997252	0.994550	0.998647	0.996599	0.994305	0.055675	0.005688	0.993660
41.jpg	32232	51574	70	174	0.997097	0.994631	0.998645	0.996638	0.994069	0.057590	0.005924	0.993739
42.jpg	31168	52605	121	156	0.996704	0.995020	0.997705	0.996362	0.993336	0.064239	0.006661	0.992875
43.jpg	27485	56343	23	199	0.997359	0.992812	0.999592	0.996202	0.994637	0.050647	0.005343	0.993924
44.jpg	24151	59687	99	113	0.997478	0.995343	0.998344	0.996843	0.994921	0.050482	0.005076	0.993822
45.jpg	31203	52534	53	260	0.996276	0.991736	0.998992	0.995364	0.992462	0.069043	0.007514	0.991947
46.jpg	19109	64619	27	295	0.996169	0.984797	0.999582	0.992190	0.992296	0.066893	0.007626	0.989135
47.jpg	24507	59169	67	307	0.995550	0.987628	0.998869	0.993248	0.991022	0.079363	0.008926	0.989201
48.jpg	21375	62435	10	230	0.997145	0.989354	0.999840	0.994597	0.994258	0.051754	0.005701	0.992471
49.jpg	20608	63110	47	285	0.996050	0.986359	0.999256	0.992807	0.991989	0.070875	0.007949	0.989266
50.jpg	25733	58128	25	164	0.997751	0.993667	0.999570	0.996619	0.995347	0.045295	0.004639	0.994539
51.jpg	17634	66145	56	215	0.996776	0.987955	0.999154	0.993554	0.993525	0.059551	0.006434	0.990321
52.jpg	21457	62397	14	182	0.997668	0.991589	0.999776	0.995682	0.995276	0.044464	0.004699	0.993815
53.jpg	19128	64632	81	209	0.996550	0.989192	0.998748	0.993970	0.993076	0.064062	0.006892	0.990232
54.jpg	17862	65934	54	200	0.996978	0.988927	0.999182	0.994054	0.993856	0.057136	0.006109	0.990895
55.jpg	31410	52374	69	197	0.996835	0.993767	0.998684	0.996226	0.993596	0.061154	0.006393	0.993174
56.jpg	21911	61899	46	194	0.997145	0.991224	0.999257	0.995241	0.994211	0.054668	0.005765	0.992529
57.jpg	25654	58138	170	88	0.996930	0.996581	0.997084	0.996833	0.993809	0.059477	0.006180	0.992729
58.jpg	24513	59294	80	163	0.997109	0.993394	0.998653	0.996024	0.994211	0.056029	0.005778	0.993023
59.jpg	28675	55111	109	155	0.996859	0.994624	0.998026	0.996325	0.993619	0.061678	0.006376	0.992924
60.jpg	25751	57971	59	269	0.996098	0.989662	0.998983	0.994323	0.992084	0.071674	0.007880	0.990732
61.jpg	32106	51566	53	325	0.995503	0.989979	0.998973	0.994476	0.990928	0.080130	0.009035	0.990418
62.jpg	28062	55663	49	276	0.996133	0.990260	0.999120	0.994690	0.992178	0.070627	0.007787	0.991241
63.jpg	40348	43419	39	244	0.996633	0.993989	0.999103	0.996546	0.993289	0.062240	0.006699	0.993280
64.jpg	28509	55057	103	381	0.994242	0.986812	0.998133	0.992472	0.988455	0.099292	0.011488	0.987191
65.jpg	32944	50825	65	216	0.996657	0.993486	0.998723	0.996104	0.993241	0.063816	0.006746	0.992923
66.jpg	26549	57236	38	227	0.996847	0.991522	0.999337	0.995429	0.993690	0.058505	0.006283	0.992727
67.jpg	29159	54488	42	361	0.995205	0.987771	0.999230	0.993500	0.990291	0.083705	0.009650	0.989331
68.jpg	28159	55617	9	265	0.996740	0.990677	0.999838	0.995258	0.993383	0.058972	0.006580	0.992598
69.jpg	31109	52668	29	244	0.996752	0.992218	0.999450	0.995834	0.993430	0.060407	0.006546	0.992971
70.jpg	28233	55544	22	251	0.996752	0.991188	0.999604	0.995396	0.993407	0.059940	0.006562	0.992634
71.jpg	27762	55998	41	249	0.996550	0.991111	0.999268	0.995190	0.993076	0.063476	0.006895	0.992202
72.jpg	23453	60265	30	302	0.996050	0.987287	0.999502	0.993395	0.992084	0.069293	0.007856	0.990223
73.jpg	23450	60336	27	237	0.996859	0.989995	0.999553	0.994774	0.993596	0.058497	0.006368	0.992080
74.jpg	30643	53086	27	294	0.996181	0.990497	0.999492	0.994994	0.992249	0.068986	0.007714	0.991659
75.jpg	23780	59914	26	330	0.995764	0.986313	0.999566	0.992939	0.991517	0.073048	0.008412	0.989614
76.jpg	31544	52152	15	339	0.995788	0.989367	0.999712	0.994540	0.991494	0.073283	0.008457	0.990954
77.jpg	23136	60607	16	291	0.996347	0.987578	0.999736	0.993657	0.992532	0.065437	0.007409	0.990696

Tabla B.11: Resultados del método de segmentación PCA con el conjunto de datos Plant Village

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
008.JPG	15532	43605	4293	2106	0.902359	0.880599	0.910372	0.895485	0.823783	0.876050	0.163941	0.790120
009.JPG	20266	39936	5009	325	0.918610	0.984216	0.888553	0.936385	0.850287	0.701778	0.132535	0.836702
010.JPG	25323	36378	3502	333	0.941483	0.987021	0.912187	0.949604	0.889758	0.576938	0.104005	0.886497
011.JPG	21647	38759	4084	1046	0.921722	0.953906	0.904675	0.929291	0.855569	0.744925	0.136039	0.845622
012.JPG	28355	18645	16723	1813	0.717163	0.939903	0.527171	0.735537	0.594300	1.478687	0.321051	0.553066
013.JPG	11223	11829	22098	20386	0.351746	0.355057	0.348660	0.351859	0.543997	1.870332	0.455530	0.215324
014.JPG	28615	32736	3507	678	0.936142	0.976855	0.903236	0.940406	0.880438	0.641293	0.115606	0.879529
015.JPG	29312	32408	3141	675	0.941772	0.977490	0.911643	0.944567	0.890216	0.604938	0.106856	0.889623
016.JPG	14774	40691	8872	1199	0.846329	0.924936	0.820996	0.872966	0.739904	1.070496	0.204680	0.698147
017.JPG	26954	26286	11116	1180	0.812378	0.958058	0.702797	0.830427	0.695136	1.222056	0.272917	0.683993
018.JPG	20926	35345	8131	1134	0.858627	0.948505	0.812977	0.880786	0.757136	1.050222	0.212287	0.742614
019.JPG	31648	29848	3866	174	0.938354	0.994532	0.885330	0.939931	0.884227	0.580975	0.109805	0.883709
020.JPG	12628	51455	1431	22	0.977829	0.98261	0.972942	0.985601	0.956524	0.254916	0.039994	0.934495
021.JPG	19805	40133	4775	823	0.914581	0.960103	0.893672	0.926887	0.843804	0.765547	0.141923	0.828851
022.JPG	20620	42536	1903	477	0.963684	0.977390	0.957177	0.967284	0.929863	0.425053	0.067690	0.921606
023.JPG	22983	35326	6083	1144	0.889725	0.952584	0.853100	0.902842	0.803696	0.922178	0.180658	0.795391
024.JPG	28181	32944	3796	615	0.932693	0.978643	0.896679	0.937661	0.874419	0.658296	0.120534	0.873261
025.JPG	9224	45156	5799	5357	0.829773	0.632604	0.7886194	0.759399	0.717496	1.254804	0.627245	
026.JPG	28148	30596	6766	26	0.896362	0.999077	0.8189092	0.908992	0.814179	0.776226	0.167279	0.811944
027.JPG	24241	36105	4297	893	0.920807	0.964704	0.929057	0.854155	0.745008	1.373776	0.848986	
028.JPG	29386	32229	3031	890	0.940170	0.970604	0.914039	0.942321	0.887417	0.628020	0.110355	0.886824
029.JPG	25269	23373	15717	1177	0.742218	0.955494	0.507928	0.776711	0.617348	1.390160	0.321066	0.589904
030.JPG	15217	44380	5417	522	0.909378	0.966834	0.891218	0.929026	0.835153	0.751575	0.137658	0.800587
031.JPG	17904	43411	3801	420	0.935393	0.977079	0.919491	0.948285	0.879453	0.608994	0.108398	0.860270
032.JPG	18882	38600	6827	1227	0.877106	0.938983	0.849715	0.894349	0.784345	0.973258	0.189352	0.764099
033.JPG	24589	31586	5929	3432	0.857162	0.877520	0.841957	0.859738	0.755126	1.167740	0.240252	0.747828
034.JPG	33607	26417	5366	146	0.915894	0.995674	0.831168	0.913421	0.845856	0.702198	0.140601	0.843155
035.JPG	24070	33441	7837	188	0.877548	0.992250	0.810141	0.901195	0.785036	0.883153	0.186159	0.778164
036.JPG	29720	25387	9573	856	0.840866	0.972004	0.726173	0.849088	0.732354	1.101088	0.237555	0.724508
037.JPG	18703	40058	5852	923	0.896622	0.952971	0.827533	0.912752	0.814542	0.861684	0.163595	0.794621
038.JPG	21578	28179	6009	9770	0.759232	0.688337	0.824237	0.756287	0.634428	1.572746	0.356354	0.609368
039.JPG	23956	28552	9248	3780	0.801208	0.863715	0.755344	0.809529	0.681448	1.395194	0.305481	0.667207
040.JPG	34892	25707	4814	123	0.924667	0.996487	0.842273	0.919380	0.860631	0.651091	0.127523	0.857416
041.JPG	21220	38333	4703	1280	0.908707	0.943111	0.890719	0.916915	0.834030	0.829818	0.155325	0.822470
042.JPG	30547	27543	7005	441	0.886383	0.985769	0.797239	0.891504	0.798581	0.881848	0.183679	0.795604
043.JPG	31569	23649	9987	331	0.842560	0.989624	0.703086	0.846355	0.734628	1.033838	0.224316	0.724888
044.JPG	12365	46295	5931	945	0.895081	0.929001	0.886436	0.907718	0.812174	0.837495	0.150550	0.756644
045.JPG	30712	30765	3843	216	0.938065	0.993016	0.888956	0.940986	0.883666	0.588330	0.110900	0.883218
046.JPG	20989	34411	8783	1353	0.845337	0.939441	0.796662	0.868051	0.738490	1.116838	0.228688	0.723378
047.JPG	27252	29126	9158	0	0.860260	1.000000	0.760788	0.880394	0.759505	0.915898	0.209229	0.754561
048.JPG	32013	28494	4961	68	0.923264	0.997880	0.851711	0.924796	0.858392	0.651260	0.131018	0.857109
049.JPG	29537	32316	3233	450	0.943802	0.984993	0.909055	0.947024	0.893837	0.574888	0.102547	0.893330
050.JPG	25202	31810	5014	3510	0.869934	0.877751	0.863839	0.870795	0.773564	1.108366	0.224218	0.767813
051.JPG	30374	29790	5294	78	0.918030	0.997439	0.849105	0.923272	0.849342	0.681789	0.139688	0.848313
052.JPG	27840	32543	4588	565	0.921371	0.980109	0.876437	0.928273	0.855003	0.720932	0.137233	0.853451
053.JPG	25509	32627	6835	565	0.887085	0.978331	0.826795	0.902563	0.799548	0.890893	0.181545	0.795004
054.JPG	32356	30765	2262	153	0.963150	0.995294	0.931511	0.963402	0.929043	0.407949	0.068920	0.928909
055.JPG	13268	44600	7158	510	0.882996	0.962894	0.861703	0.912344	0.793368	0.858637	0.157282	0.743518
056.JPG	11843	31402	7412	14879	0.659866	0.443193	0.809038	0.626115	0.551108	1.742875	0.384248	0.465894
057.JPG	29716	30656	5011	153	0.921204	0.994878	0.859506	0.927192	0.854694	0.676918	0.135612	0.853761
058.JPG	22391	25761	14237	3147	0.876772	0.644057	0.760415	0.610187	0.540845	0.351174	0.579990	
059.JPG	27728	34279	2668	861	0.946152	0.969884	0.927788	0.948836	0.898101	0.583917	0.099902	0.896878
060.JPG	29370	29168	4771	2227	0.893219	0.929519	0.859424	0.894471	0.809359	0.961040	0.188175	0.807165
061.JPG	18665	28278	17734	859	0.716293	0.956003	0.614579	0.785291	0.593533	1.393147	0.302936	0.552099
062.JPG	32362	21259	11861	54	0.818192	0.998334	0.641878	0.820106	0.702448	1.058856	0.234010	0.685824
063.JPG	16985	23969	19384	5198	0.624908	0.765676	0.552880	0.659278	0.531189	1.776133	0.406603	0.451133
064.JPG	22836	36213	6012	475	0.901016	0.979623	0.857620	0.918622	0.821552	0.812474	0.159585	0.813344
065.JPG	18427	36819	9205	1085	0.842987	0.944393	0.799996	0.872194	0.735255	1.094690	0.219489	0.711593
066.JPG	18612	34997	10894	1033	0.818908	0.947417	0.762611	0.855014	0.702254	1.173601	0.240311	0.677633
067.JPG	32202	29864	2849	621	0.947052	0.981080	0.912909	0.946995	0.899600	0.565455	0.098067	0.899204
068.JPG	18911	44804	1245	576	0.972214	0.970442	0.972964	0.971703	0.945913	0.354231	0.053056	0.936485
069.JPG	32399	24136	7998	1003	0.862656	0.969972	0.751105	0.860538	0.763012	1.031328	0.213027	0.755456
070.JPG	19924	40635	2903	2074	0.924057	0.905719	0.933323	0.919521	0.859698	0.759862	0.137486	0.845557
071.JPG	21442	22728	20351	1015	0.673981	0.954803	0.527589	0.741196	0.560532	1.476501	0.348262	0.508166
072.JPG	29451	10234	24927	924	0.605455	0.969580	0.291061	0.630321	0.522253	1.453660	0.248762	0.408039
073.JPG	27056	32628	5458	394	0.910706	0.985647	0.856693	0.921170	0.837331	0.760923	0.150505	0.835020
074.JPG	26583	21135	12812	5006	0.728119	0.841527	0.622588	0.732058	0.604112	1.627139	0.371936	0.570699
075.JPG	27324	31426	4805	1981	0.896454	0.932401	0.867379	0.899890	0.814324	0.935680	0.181593	0.811708
076.JPG	22184	38907	3121	1324	0.932175	0.943679	0.925740	0.934709	0.873469	0.693349	0.122632	0.865181
077.JPG	17398	37504	9233	1401	0.837738	0.925475	0.802448	0.863961	0.728171	1.131976	0.225238	0.699914
078.JPG	31689	18605	14858	384	0.767426	0.988027	0.555987	0.772007	0.643060	1.234375	0.263618	0.612500
079.JPG	17408	33762	8866	5500	0.780792	0.759909	0.792015	0.775962	0.657666	1.477708	0.323611	0.624664
080.JPG	29778	31069	4500	189	0.928452	0.993693	0.873485	0.933589	0.867139	0.641334	0.125034	0.866413
081.JPG	25957	37753	1741	85	0.972137	0.996736	0.955917	0.976327	0.945625	0.327007	0.052566	0.943860
082.JPG	32088	25420	7928	100	0.877502	0.996893	0.762265	0.879579	0.784944	0.870624	0.187520	0.779862
083.JPG	31123	18430	14300	1683	0.756119	0.948698	0.563092	0.755895	0.631219	1.389731	0.294405	0.598174
084.JPG	20752	25794	15551	3439	0.710236	0.857840	0.623872	0.740856	0.588392	1.599076	0.363851	0.549064
085.JPG	7369	14172	29944	14051	0.328690	0.344024	0.321244	0.332634	0.558709	1.751920	0.395755	0.193531
086.JPG	28917	30030										

Tabla B.12: Resultados del método de segmentación OTSU con el conjunto de datos Plant Village

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
008.JPG	7636	44085	3813	10002	0.789199829	0.432928904	0.920393336	0.67666112	0.66730331	1.288904703	0.239234999	0.55872039
009.JPG	19872	41031	3914	719	0.929300603	0.965081832	0.912915786	0.938998809	0.868474326	0.675737318	0.121448518	0.854585074
010.JPG	24910	36695	3185	746	0.9400177	0.970922981	0.920135406	0.945529194	0.88717572	0.618116281	0.108539242	0.883413238
011.JPG	18943	38831	4012	3750	0.881561279	0.834750804	0.906355764	0.870553284	0.7910584	1.03109577	0.205499083	0.771237308
012.JPG	21785	18607	16761	8383	0.616333008	0.722122779	0.526097037	0.624109908	0.527052421	1.880135589	0.453847267	0.444737001
013.JPG	31139	13851	20076	470	0.68649292	0.985130817	0.408258909	0.696694863	0.569495748	1.359451006	0.264283737	0.502500871
014.JPG	24045	35231	3012	5248	0.873962402	0.82084457	0.916894297	0.868869434	0.779692395	1.077046505	0.215747069	0.772614795
015.JPG	27642	32537	3012	2345	0.918258667	0.921799446	0.915271878	0.918535662	0.849776226	0.81457464	0.149737973	0.84804094
016.JPG	13935	40836	8727	2038	0.835739136	0.872409691	0.82392107	0.848165381	0.725457838	1.15486754	0.222976813	0.677796137
017.JPG	25520	27734	9608	2614	0.812591553	0.90708751	0.741511149	0.824299329	0.695403231	1.313542463	0.286883811	0.684062314
018.JPG	20625	35524	7952	1435	0.856765747	0.934950136	0.817094489	0.876022312	0.754516302	1.077829204	0.217252261	0.739052048
019.JPG	31346	30558	3156	476	0.944580078	0.985041795	0.906389037	0.945715416	0.895219892	0.571953035	0.101681745	0.894884613
020.JPG	11185	51857	1029	1465	0.96194458	0.884189723	0.980543055	0.932366389	0.926728084	0.437222346	0.070385111	0.885797999
021.JPG	18750	40426	4482	1878	0.902954102	0.908958697	0.900195956	0.904577327	0.824839728	0.878836361	0.165021028	0.805479638
022.JPG	11786	37465	6794	9311	0.754257202	0.558657629	0.847116272	0.70288695	0.629303312	1.520403912	0.334385637	0.561474589
023.JPG	21823	35642	5947	2304	0.874099731	0.904505326	0.856383878	0.88044602	0.779875027	1.050974045	0.208653304	0.76841644
024.JPG	22449	30618	6122	6347	0.809738159	0.779587443	0.833369624	0.804478534	0.691851848	1.398707951	0.306721256	0.676733182
025.JPG	1008	45664	5291	13573	0.712158203	0.069131061	0.896163281	0.482647171	0.59001595	1.217537727	0.173337275	0.379193214
026.JPG	28116	31588	5774	58	0.911017412	0.97941364	0.845457352	0.921699658	0.8378321	0.713200976	0.147973912	0.836151674
027.JPG	21448	19112	21290	3686	0.6118896484	0.853346095	0.473045889	0.663195977	0.528251038	1.720035349	0.403350375	0.447718155
028.JPG	25444	10943	24317	4832	0.555221558	0.840401638	0.310351673	0.575376656	0.506081197	1.74634272	0.354246318	0.369469462
029.JPG	24053	23917	15173	2393	0.731964111	0.909513726	0.611844461	0.760679094	0.607622869	1.504039423	0.349691308	0.577260354
030.JPG	14863	46368	3429	876	0.934310913	0.94434208	0.93114043	0.937741255	0.877223557	0.639571326	0.111274636	0.845185086
031.JPG	18324	0	47212	0	0.279602051	1	0	0	0.5	0.597157818	0.854887175	0
032.JPG	11191	38725	6692	8918	0.761810303	0.556516982	0.852686728	0.704601855	0.637035796	1.488122701	0.325646637	0.565084416
033.JPG	13861	32000	5515	14160	0.699783325	0.494664716	0.852992136	0.673828426	0.579844732	1.653453398	0.357307672	0.516317673
034.JPG	33324	27173	4610	429	0.923110962	0.987290019	0.854953906	0.921121962	0.857966135	0.698078751	0.133265808	0.856030885
035.JPG	18078	34260	7018	6180	0.798614502	0.745238684	0.829982073	0.787610378	0.678299883	1.425477077	0.314083958	0.649911097
036.JPG	24673	17251	17709	5903	0.639709473	0.806940084	0.493449657	0.65019487	0.539021913	1.787046117	0.412050583	0.466656128
037.JPG	2440	40825	5085	17186	0.660171509	0.124324875	0.889239817	0.506782346	0.551312755	1.394307127	0.203200377	0.372864727
038.JPG	12341	28315	5873	19007	0.620361328	0.393677428	0.828214578	0.610946003	0.528981206	1.761679422	0.376784672	0.431951212
039.JPG	17551	26986	10814	10185	0.679580688	0.632787713	0.713915344	0.673351528	0.564491802	1.795057043	0.429848557	0.508828963
040.JPG	37666	27042	3479	249	0.943115234	0.922888762	0.886012909	0.939450836	0.892646494	0.558925273	0.101661317	0.890049346
041.JPG	11515	40270	2766	10985	0.790176392	0.511777778	0.935728228	0.723753003	0.668364196	1.310175748	0.250591354	0.600545475
042.JPG	29643	28669	6679	1345	0.877563477	0.965696102	0.806674771	0.881635437	0.78515117	0.996592286	0.203626039	0.781760039
043.JPG	27495	26711	6925	4405	0.82711792	0.861912226	0.794119396	0.828015811	0.713948009	1.318328623	0.283744679	0.705104414
044.JPG	10775	46511	5715	2535	0.87411499	0.809541698	0.890571746	0.850056722	0.779966365	0.993344441	0.187253531	0.707890707
045.JPG	30566	31333	3275	362	0.944503784	0.988295396	0.905368701	0.946832048	0.895029986	0.562843494	0.101316993	0.940693295
046.JPG	16563	34597	8597	5779	0.780639648	0.741339182	0.800967727	0.771153454	0.65749487	1.476338032	0.323829264	0.620869231
047.JPG	27248	28751	9533	4	0.854476929	0.990853222	0.750992582	0.875422902	0.751239088	0.938201704	0.215687718	0.745755225
048.JPG	31917	29154	4301	164	0.931869507	0.99488794	0.871439247	0.933163593	0.873020604	0.618074299	0.119361018	0.872231337
049.JPG	24498	32606	2943	5489	0.871337891	0.816954013	0.917212861	0.867803437	0.775712246	1.089538035	0.219300572	0.769162206
050.JPG	22938	37914	4033	5774	0.850357056	0.798899415	0.890479035	0.844689225	0.745410719	1.206615829	0.250466399	0.73504182
051.JPG	28785	29819	5265	1667	0.894226074	0.945258111	0.849931593	0.897594852	0.810729268	0.935455828	0.184088056	0.808548718
052.JPG	20195	36247	4484	8210	0.806304932	0.710966379	0.879238372	0.795102376	0.687603267	1.38605916	0.29850204	0.666982017
053.JPG	23096	35303	4159	2978	0.891098022	0.885786607	0.89460747	0.890197039	0.805888494	0.983167952	0.191378204	0.797853182
054.JPG	32099	31185	1842	410	0.965637207	0.987388108	0.94422745	0.965807779	0.933663426	0.410449864	0.065409621	0.933574036
055.JPG	12932	44927	6831	846	0.882858276	0.938597765	0.868020403	0.903309084	0.739315763	0.887780876	0.161750996	0.740776685
056.JPG	9842	31479	7335	16880	0.630508423	0.368310755	0.811021796	0.589666276	0.534057787	1.747965297	0.371279577	0.427093264
057.JPG	29167	31276	4391	702	0.922286987	0.976497372	0.876889001	0.926693186	0.85652155	0.727521686	0.137534366	0.855520838
058.JPG	9784	28328	16250	15754	0.511657715	0.383115358	0.593729686	0.488422522	0.500266318	1.933115533	0.475386692	0.330081317
059.JPG	25567	32671	4276	3022	0.888641357	0.894295009	0.884266652	0.88928083	0.802081189	1.001615171	0.19620794	0.797767318
060.JPG	19313	10811	23128	12284	0.459655762	0.611228914	0.31854209	0.464885502	0.503255127	1.927468964	0.453963302	0.293366322
061.JPG	12429	24818	21194	7095	0.568344116	0.636601106	0.539381031	0.587991069	0.509342694	1.840320738	0.407116219	0.386360765
062.JPG	33329	22706	10414	87	0.839767456	0.99731614	0.685567633	0.841441887	0.730838268	1.002311154	0.220556878	0.719252502
063.JPG	13355	25022	18331	8828	0.585586548	0.602037596	0.577168823	0.58960321	0.514637485	1.880814269	0.434906826	0.404560043
064.JPG	22324	37058	5167	987	0.906097412	0.957659474	0.877631735	0.917645604	0.829753264	0.828794373	0.157433832	0.820666925
065.JPG	17307	36954	9070	2205	0.827957153	0.88699262	0.802928907	0.844960763	0.715087425	1.214840855	0.245109987	0.685840311
066.JPG	18086	35295	10596	1559	0.814529419	0.920641385	0.769105053	0.844873219	0.6978529	1.223692424	0.249448037	0.670994478
067.JPG	31382	16733	15980	19641	0.456466675	0.401608628	0.511509186	0.456558907	0.503793365	1.980172142	0.490158869	0.294806302
068.JPG	15661	44991	1058	3826	0.907089233	0.977024474	0.803642228	0.862005738	0.705761602	0.125437691	0.832109798	0.620869231
069.JPG	26009	27025	5109	7393	0.809234619	0.778665948	0.841009523	0.809837735	0.691228514	1.398657262	0.306827713	0.6795115829
070.JPG	18393	41054	2484	3605	0.907089233	0.836214466	0.942946392	0.889533929	0.831490412	0.689009501	0.163429908	0.81111141
071.JPG	14329	25818	17261	8128	0.612594604	0.638063855	0.599317533	0.618690694	0.525347847	1.852394694	0.427572392	0.432475443
072.JPG	14791	10947	24214	15584	0.392730713	0.486946502	0.311339268	0.399142885	0.523025768	1.908956227	0.461634654	0.243305001
073.JPG	25737	32931	5155	1713	0.9585202637	0.937595628	0.864648427	0.901122028	0.812343264	0.930352192	0.180778261	0.808369127
074.JPG	31263	13898	20049	326	0.689102173	0.898679952	0.409402893	0.699541422	0.571478103	1.335635513	0.260353294	0.505420679
075.JPG	17426	31458	4773	11879	0.745910645	0.594642552	0.868261986	0.731452269	0.620923297	1.561077943	0.342065923	0.582591376
076.JPG	14726	39315										

Tabla B.13: Resultados del método de segmentación GA con el conjunto de datos Plant Village

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
008.JPG	17552	47560	338	86	0.993530273	0.995124164	0.992943338	0.994033751	0.987083821	0.107384783	0.012798555	0.983712073
009.JPG	20510	44806	139	81	0.996643066	0.996066243	0.996907331	0.996486787	0.993157015	0.064946316	0.006831031	0.992074597
010.JPG	25620	39786	94	36	0.998016357	0.998596819	0.997642929	0.998119874	0.995979732	0.041120751	0.00401639	0.995783504
011.JPG	22641	42740	103	52	0.997634888	0.997708544	0.997595873	0.997652209	0.995129034	0.048620627	0.004865211	0.994625811
012.JPG	30077	35271	97	91	0.997131348	0.996983559	0.997257408	0.997120483	0.994248724	0.05694469	0.005750962	0.994212607
013.JPG	31577	33846	81	32	0.998275757	0.99898763	0.997612521	0.998300076	0.996405355	0.037455911	0.037453922	0.996401081
014.JPG	29213	36170	73	80	0.997665405	0.997268972	0.997985818	0.997027395	0.995280891	0.048039759	0.00471882	0.995227377
015.JPG	29913	35454	95	74	0.997421265	0.997532264	0.997327632	0.997429948	0.994734315	0.052720905	0.00526487	0.994696592
016.JPG	15956	49444	119	17	0.997924805	0.998035704	0.997599015	0.99826736	0.995827768	0.040414641	0.004153392	0.994361649
017.JPG	28079	37317	85	55	0.99786377	0.99804567	0.997727394	0.990416473	0.995645442	0.040635358	0.00353152	0.995557544
018.JPG	22002	42757	719	58	0.988143921	0.997370807	0.98346214	0.997449447	0.976494447	0.170086044	0.02312973	0.973887841
019.JPG	31770	35543	171	52	0.996549729	0.99836591	0.994927923	0.996646917	0.993126707	0.064820403	0.00686559	0.993121713
020.JPG	12904	52840	46	46	0.998596191	0.996363636	0.999130205	0.99774692	0.997074559	0.030801041	0.0029222435	0.995312526
021.JPG	20487	44760	148	141	0.99559021	0.993164631	0.996704373	0.994934502	0.990916736	0.083090231	0.009074324	0.989486325
022.JPG	21009	44413	26	88	0.998260498	0.995828791	0.999414928	0.99762186	0.996374947	0.037317392	0.003619727	0.995487047
023.JPG	24077	41268	141	50	0.997085571	0.997927633	0.996594943	0.997261288	0.994097025	0.056912225	0.005892434	0.993663475
024.JPG	28718	36708	32	78	0.998321533	0.997291229	0.999129015	0.998210153	0.996557407	0.036221147	0.00344102	0.996505649
025.JPG	14554	50930	25	47	0.998901367	0.996770627	0.999506871	0.998142999	0.997805115	0.024032095	0.00292268	0.996829405
026.JPG	28073	37285	77	101	0.997283936	0.996415135	0.997938082	0.997177109	0.994622895	0.053727238	0.005385396	0.994504926
027.JPG	25084	40312	90	50	0.99786377	0.998010663	0.997772388	0.997891525	0.995675828	0.04414676	0.004321345	0.995430141
028.JPG	30243	35166	94	33	0.998062134	0.998010028	0.997334009	0.998122059	0.996040524	0.040595112	0.003956876	0.996018162
029.JPG	26409	38991	99	37	0.997924805	0.998600923	0.997467383	0.998034153	0.995706214	0.043482226	0.004289456	0.995542538
030.JPG	15688	49704	93	51	0.997802734	0.996759642	0.998132418	0.99744603	0.99552391	0.044617879	0.004466667	0.993883111
031.JPG	18274	47164	48	50	0.998504639	0.997271338	0.998983309	0.998127324	0.996983278	0.032194432	0.003015524	0.996258352
032.JPG	20038	45315	112	71	0.997207642	0.996469243	0.997534506	0.997001874	0.994339754	0.055311263	0.00565285	0.99335691
033.JPG	27957	37428	87	64	0.997695923	0.997715099	0.997680928	0.997698463	0.995341664	0.047387858	0.004657236	0.995242752
034.JPG	33686	31698	85	67	0.997680664	0.998014991	0.997325614	0.997670303	0.995220145	0.048536337	0.004779424	0.995215832
035.JPG	24188	41205	73	70	0.997817993	0.997114354	0.998231503	0.997672929	0.995584674	0.045179096	0.00441435	0.995266937
036.JPG	30436	34925	35	140	0.997329712	0.995421245	0.998998856	0.997210051	0.994643249	0.052150782	0.005350399	0.994618165
037.JPG	19602	45849	61	24	0.998703003	0.998777132	0.998671313	0.998724223	0.997257148	0.029279649	0.002739438	0.996734668
038.JPG	31274	34121	67	74	0.997848511	0.997639403	0.998040248	0.997830825	0.995645442	0.044857701	0.004354433	0.995637279
039.JPG	27702	37714	86	34	0.998168945	0.998774156	0.997724868	0.998249512	0.99634454	0.037962273	0.003652916	0.996257523
040.JPG	34937	30448	73	78	0.997695923	0.99772383	0.997608204	0.997690294	0.995341664	0.047532105	0.004658204	0.995319849
041.JPG	22453	42974	62	47	0.998336792	0.997911111	0.998559346	0.998235228	0.996618235	0.035756292	0.003380332	0.99625196
042.JPG	30934	34462	86	54	0.99786377	0.99825739	0.99751071	0.99788405	0.995615058	0.044927993	0.004383984	0.995602386
043.JPG	31772	33551	85	128	0.996749878	0.995987461	0.997472946	0.996730203	0.993369197	0.064197266	0.006629808	0.99336434
044.JPG	13229	52132	94	81	0.997329712	0.99391435	0.99820013	0.99605724	0.994612895	0.052136201	0.005376318	0.991705164
045.JPG	30840	34288	320	88	0.993774414	0.997154682	0.990753583	0.993954132	0.987475477	0.107068238	0.012493438	0.98740474
046.JPG	22305	43140	54	37	0.99861115	0.99843926	0.998749826	0.998546876	0.997196281	0.030341389	0.002802421	0.996881968
047.JPG	27212	38182	102	40	0.997833252	0.998532218	0.997335702	0.99793396	0.995584674	0.044651895	0.004411453	0.995457948
048.JPG	32007	33397	58	74	0.99798584	0.997633339	0.998266328	0.997979833	0.995979732	0.041837807	0.004020043	0.995077975
049.JPG	29907	35480	69	80	0.99772644	0.997332177	0.998059017	0.997695597	0.995372016	0.047248795	0.004627708	0.995338473
050.JPG	28612	36711	113	100	0.996749878	0.996517136	0.996931349	0.996724242	0.993338882	0.064487113	0.006660015	0.993236145
051.JPG	30351	35028	56	101	0.99760437	0.996683305	0.998403831	0.997543568	0.995037931	0.049976822	0.004960797	0.995012637
052.JPG	28260	37013	118	145	0.995986938	0.994895265	0.996822062	0.995858664	0.991824342	0.076763097	0.008174	0.997676868
053.JPG	25954	39410	52	120	0.997375488	0.995397714	0.998682277	0.997039995	0.994612895	0.053199376	0.005382242	0.994376987
054.JPG	32430	32959	68	79	0.997576958	0.997569904	0.997941079	0.99755491	0.99549353	0.046197047	0.004506344	0.995493282
055.JPG	13568	51605	153	210	0.99446106	0.98475831	0.997043935	0.990901123	0.988983311	0.095478042	0.010966399	0.983478969
056.JPG	26645	38574	240	77	0.995162964	0.997118479	0.993816664	0.995467572	0.990312131	0.086599911	0.009663949	0.989683272
057.JPG	29778	35530	137	91	0.996520996	0.996953363	0.996158914	0.996556139	0.99297518	0.067305433	0.00702234	0.992920956
058.JPG	25464	39970	28	74	0.998443604	0.997102357	0.999299965	0.998201161	0.996800739	0.03858968	0.003197068	0.996366448
059.JPG	28486	36899	48	103	0.997695923	0.996397216	0.998700842	0.997549029	0.995341664	0.047081434	0.004655799	0.995263965
060.JPG	31558	33879	60	39	0.99848938	0.998765706	0.998232122	0.998498914	0.99683116	0.033943155	0.003168382	0.996827216
061.JPG	19477	45974	38	47	0.998703003	0.997592706	0.999174128	0.998383417	0.997287583	0.029427795	0.002711441	0.996759113
062.JPG	32250	33052	68	166	0.996429443	0.994879072	0.99794686	0.996412966	0.992884276	0.067447871	0.007111094	0.99288323
063.JPG	22108	43317	36	75	0.998306274	0.996619033	0.999169608	0.99789432	0.996526995	0.03637484	0.003470194	0.996122201
064.JPG	23273	42096	129	38	0.997451782	0.998369868	0.996944938	0.997657403	0.99482539	0.050456411	0.003169329	0.99436254
065.JPG	19426	45860	164	86	0.996185303	0.995592456	0.996436642	0.996014549	0.992369309	0.070958282	0.007612126	0.990899713
066.JPG	19596	45761	130	49	0.997268677	0.997055727	0.997167201	0.997336464	0.99449149	0.053348793	0.005495128	0.993453639
067.JPG	32672	32687	26	151	0.997299194	0.995399567	0.999205209	0.997302388	0.99449149	0.052970037	0.005501674	0.994491531
068.JPG	19399	45972	77	88	0.9974823	0.995481169	0.998327868	0.996906019	0.994916474	0.050582699	0.005080161	0.993922208
069.JPG	33340	32014	120	62	0.99722229	0.998143824	0.996265638	0.997204731	0.994370099	0.055493728	0.005627873	0.994367858
070.JPG	21979	43443	95	19	0.998260498	0.999136285	0.997817998	0.998477142	0.996405355	0.036421699	0.003587271	0.995975048
071.JPG	22331	42918	161	126	0.995620728	0.994389277	0.99626268	0.995325979	0.991219178	0.080941486	0.008772275	0.990266918
072.JPG	30306	35126	35	69	0.998413086	0.997728395	0.999004579	0.998366487	0.996739901	0.034726673	0.003259344	0.99722247
073.JPG	27399	37988	98	51	0.99772644	0.998142077	0.997426876	0.997784476	0.995432771	0.046316923	0.004564672	0.995310937
074.JPG	31507	33906	41	82	0.998123169	0.99740416	0.998792235	0.998098197	0.996162119	0.039976928	0.003836978	0.996157021
075.JPG	29242	36160	71	63	0.997955322	0.997850196	0.998040352	0.997945274	0.995827768	0.043190459	0.004171863	0.995780956
076.JPG	23432	41952	76	76	0.997680664	0.996767058	0.998191682	0.99747937	0.995220145	0.048318321	0.004077846	0.994807166
077.JPG	18705	46641										

Tabla B.14: Resultados del método de segmentación PSO con el conjunto de datos Plant Village

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
008.JPG	17603	47609	289	35	0.995056152	0.998015648	0.993966345	0.995990997	0.990100607	0.083712552	0.009803509	0.987507131
009.JPG	20564	44908	37	27	0.999023438	0.998688748	0.999176772	0.99893276	0.997835566	0.024144424	0.002163385	0.997489781
010.JPG	25623	39863	17	33	0.999237061	0.998713751	0.999573721	0.999143736	0.998414321	0.018493679	0.001585342	0.998335879
011.JPG	22666	42829	14	27	0.99937439	0.998810206	0.999673225	0.999241716	0.998597155	0.016599886	0.001502571	0.998450653
012.JPG	30111	35358	10	57	0.998977661	0.998110581	0.999717259	0.99891392	0.997926926	0.022803693	0.002071804	0.997913604
013.JPG	31595	33912	15	14	0.999557495	0.999557088	0.999557874	0.999557481	0.998962925	0.012795012	0.001037035	0.998961638
014.JPG	29257	36209	34	36	0.998931885	0.998771037	0.999061888	0.998916465	0.997805115	0.024749561	0.002194824	0.997780207
015.JPG	29963	35519	30	24	0.999176025	0.999199653	0.999156094	0.999177874	0.99823152	0.020456697	0.001768349	0.998218747
016.JPG	15964	49522	41	9	0.999237061	0.999436549	0.999172777	0.99930466	0.998444791	0.017467061	0.001552983	0.997893325
017.JPG	28093	37385	17	41	0.99911499	0.998542689	0.999545479	0.999044084	0.998140132	0.021239477	0.001459422	0.998102051
018.JPG	22050	43442	34	10	0.999328613	0.999546091	0.999217959	0.998882325	0.998536207	0.016866689	0.001462778	0.998361881
019.JPG	31790	33673	41	32	0.998886108	0.998994406	0.998783888	0.9988589147	0.997683318	0.025919261	0.002316553	0.997681435
020.JPG	12642	52872	14	8	0.999664307	0.999367589	0.99973528	0.999551434	0.999206846	0.009752478	0.000792639	0.998727712
021.JPG	20598	44864	44	30	0.99887085	0.998545666	0.999020219	0.998782943	0.99743977	0.027898872	0.002558554	0.997033853
022.JPG	21087	44427	12	10	0.999664307	0.999525999	0.999729967	0.999627983	0.999176352	0.010333106	0.000823488	0.999056778
023.JPG	24113	41384	25	14	0.999404907	0.999419737	0.999396267	0.999408902	0.998719064	0.015273566	0.001280603	0.998623579
024.JPG	28781	36730	10	15	0.99961853	0.999479094	0.999727817	0.999603456	0.99914586	0.010784244	0.000854113	0.999133121
025.JPG	14565	50944	11	16	0.999588013	0.998902682	0.999784123	0.999343402	0.999176352	0.010214642	0.000823606	0.998809979
026.JPG	28118	37340	22	56	0.998808914	0.998012352	0.999411166	0.998711759	0.997591981	0.026505165	0.002404706	0.997543479
027.JPG	25118	40384	18	16	0.999481201	0.999363412	0.999554478	0.999458945	0.998901954	0.013428785	0.001097954	0.998839027
028.JPG	30261	35239	21	15	0.999450684	0.999504558	0.999404424	0.99945491	0.998810505	0.014417838	0.00118941	0.998803619
029.JPG	26427	39050	40	19	0.999099731	0.999281555	0.998976772	0.999129138	0.998048753	0.022055134	0.001950508	0.997973693
030.JPG	15711	47362	35	28	0.999038696	0.9982927146	0.998759063	0.998048753	0.9981869231	0.001950025	0.001950025	0.997328995
031.JPG	18315	47204	8	9	0.999740601	0.999508841	0.999830552	0.999669696	0.999450826	0.007205195	0.000549129	0.999318466
032.JPG	20075	45414	13	34	0.999282837	0.998309215	0.999713827	0.999011521	0.998475262	0.017678566	0.001523952	0.998207693
033.JPG	27987	37500	15	34	0.999252319	0.998786624	0.999600016	0.999193392	0.998444791	0.01816733	0.001554902	0.998411379
034.JPG	33732	31743	40	21	0.999069214	0.999377833	0.998741466	0.999059649	0.997987838	0.022758804	0.002011817	0.997986024
035.JPG	24213	41261	17	45	0.999053955	0.998144942	0.999588158	0.99886655	0.998048753	0.022009661	0.00195031	0.997907422
036.JPG	30545	34955	5	31	0.999450684	0.998966133	0.998566979	0.999421556	0.99887147	0.013402113	0.00112816	0.998866357
037.JPG	19619	45887	23	7	0.999542236	0.99964333	0.99949902	0.999571175	0.998932439	0.012719823	0.001066865	0.998792888
038.JPG	31327	34152	36	21	0.999130249	0.999330101	0.998946999	0.99913855	0.998201056	0.020701435	0.001798738	0.998197735
039.JPG	27728	37783	17	8	0.99961853	0.999711566	0.999550265	0.999630915	0.999176352	0.010338475	0.000823526	0.999156518
040.JPG	34992	30495	26	23	0.999252319	0.999343139	0.999148128	0.999245633	0.998444791	0.018302234	0.001555152	0.999437475
041.JPG	22474	43012	24	26	0.999237061	0.998844444	0.999442327	0.999143386	0.998414321	0.018514585	0.001585518	0.998241923
042.JPG	30967	34506	42	21	0.999038696	0.999322318	0.9987843	0.999053309	0.997957381	0.023045469	0.002042205	0.997951435
043.JPG	31835	33615	21	65	0.998687744	0.997962382	0.999375669	0.998669026	0.997226714	0.029983419	0.002772444	0.997224698
044.JPG	13283	52201	25	27	0.999206543	0.997971745	0.999521311	0.998746381	0.99829245	0.019345188	0.00170663	0.997364449
045.JPG	30889	34324	284	39	0.995071411	0.998739007	0.991793805	0.995266406	0.99004018	0.086162334	0.00992665	0.990012839
046.JPG	22308	43185	9	34	0.999343872	0.998478202	0.999791638	0.99913492	0.998597155	0.016381804	0.001402128	0.998483938
047.JPG	27238	38241	43	14	0.999130249	0.999486276	0.998876815	0.999181546	0.998170594	0.020688095	0.001828531	0.998117628
048.JPG	32032	33437	18	49	0.998977661	0.998472616	0.999461964	0.99896729	0.997957381	0.02293152	0.002042098	0.997956473
049.JPG	29908	35538	11	79	0.998626709	0.997365525	0.999690568	0.998528047	0.997165849	0.029825514	0.002831614	0.997144853
050.JPG	28689	36796	28	23	0.999221802	0.999198941	0.999239626	0.999219284	0.998261984	0.020127807	0.001737835	0.998235008
051.JPG	30436	35052	32	16	0.999267578	0.999474583	0.999087903	0.999281243	0.998353384	0.019050999	0.00146628	0.99834519
052.JPG	28367	37089	42	38	0.998779297	0.998662207	0.99886887	0.998765539	0.997378892	0.028850719	0.00262085	0.997331726
053.JPG	26048	38997	465	26	0.992507935	0.999002838	0.988216512	0.993609675	0.984977612	0.117388165	0.014881925	0.984375668
054.JPG	32456	33016	11	53	0.999023438	0.998369682	0.999666939	0.999018311	0.998018295	0.022038098	0.001980873	0.998018178
055.JPG	13733	51721	37	45	0.998748779	0.996733924	0.999285135	0.998009529	0.997500652	0.026980201	0.002497149	0.996240751
056.JPG	26696	38695	119	26	0.997787476	0.999027019	0.996934096	0.997980557	0.99552391	0.044370178	0.004468889	0.99536937
057.JPG	29824	35663	34	45	0.998794556	0.998493421	0.999046738	0.99877008	0.99743977	0.028300953	0.002560106	0.997419548
058.JPG	25526	39699	29	12	0.99937439	0.999530112	0.999027464	0.999402538	0.998658108	0.015834362	0.001341441	0.998589694
059.JPG	28550	36937	10	39	0.999252319	0.998635839	0.999729342	0.99918259	0.998444791	0.017968703	0.001554654	0.998418948
060.JPG	31583	33927	12	14	0.999603271	0.99955692	0.999646424	0.999601672	0.999054388	0.011807082	0.00094559	0.999053187
061.JPG	19510	46003	9	14	0.999649048	0.999282934	0.999804399	0.999543666	0.999176352	0.010347233	0.000823556	0.999015575
062.JPG	32377	33103	17	39	0.999145508	0.99879689	0.999486715	0.999141803	0.99823152	0.020349036	0.001768255	0.998231328
063.JPG	22152	43333	20	31	0.999221802	0.998602533	0.999538671	0.999070602	0.998353384	0.01910247	0.001646255	0.998161642
064.JPG	23296	42192	33	15	0.999267578	0.999356527	0.999218472	0.9992875	0.998383852	0.018618492	0.00161538	0.998237511
065.JPG	19426	46012	12	86	0.998504639	0.995592456	0.999739266	0.997665861	0.996983278	0.031030854	0.003009808	0.996391056
066.JPG	19615	45865	26	30	0.999145508	0.998472894	0.99943344	0.998953167	0.99823152	0.020278994	0.0017681	0.997894257
067.JPG	32801	32697	16	22	0.999420166	0.999329738	0.999510898	0.999420318	0.998719064	0.015442739	0.001280915	0.99871908
068.JPG	19473	46018	31	14	0.999313354	0.999281572	0.999326804	0.999304188	0.998566681	0.016691219	0.001432492	0.998285834
069.JPG	33376	32082	52	26	0.998809814	0.999221603	0.998381776	0.99880169	0.997531094	0.027209057	0.002468434	0.997530172
070.JPG	21982	43519	19	16	0.999465942	0.999272661	0.9995636	0.99941813	0.998810505	0.014342914	0.001189272	0.998666642
071.JPG	22412	43040	39	45	0.998718262	0.99799671	0.999094687	0.998545428	0.997378892	0.028708342	0.002620569	0.997901471
072.JPG	30360	35149	12	15	0.999588013	0.999506173	0.999658713	0.999582443	0.999084878	0.011473027	0.000915106	0.999079977
073.JPG	27442	38056	30	8	0.999420166	0.999708561	0.999212309	0.999460435	0.998810505	0.014127377	0.001189067	0.998778515
074.JPG	31570	33918	29	19	0.999267578	0.999398525	0.999145727	0.999272126	0.998444791	0.018243184	0.001555083	0.998442813
075.JPG	29487	36210	21	18	0.999404907	0.99938577	0.999420386	0.999403078	0.998719064	0.015415525	0.001280866	0.998704634
076.JPG	23276	42000	28	32	0.999084473	0.998638761	0.999333777	0.998986269	0.997957381	0.023133991	0.002042344	0.9977803
077.JPG	18641	46719	18	35	0.999191284	0.998138199	0.999614866					

Tabla B.15: Resultados del método de segmentación CP con el conjunto de datos Plant Village

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
008.JPG	17632	47585	313	6	0.995132446	0.999659825	0.99346528	0.996562553	0.990251691	0.079161696	0.009627376	0.987708683
009.JPG	20584	44851	94	7	0.998458862	0.999600046	0.997908555	0.9987843	0.996709483	0.032719367	0.00328081	0.996188318
010.JPG	25628	39873	7	28	0.999465942	0.998908637	0.999824473	0.999366555	0.99887147	0.013547994	0.001128156	0.998815575
011.JPG	22679	42832	11	14	0.99961853	0.99938307	0.999743249	0.999563159	0.999084878	0.011407964	0.000915043	0.998989363
012.JPG	30130	35365	3	38	0.99937439	0.998740387	0.99915178	0.999327782	0.998719064	0.014739111	0.001280264	0.998710854
013.JPG	31598	33922	5	11	0.999755859	0.999651998	0.999532625	0.999752311	0.999359326	0.008361803	0.000640662	0.999358527
014.JPG	29236	36226	17	57	0.99887085	0.998054143	0.999540944	0.998732543	0.997683318	0.025506092	0.002315647	0.99765674
015.JPG	29970	35508	41	17	0.99911499	0.999433088	0.998846662	0.999139875	0.998106671	0.021437742	0.001889782	0.998096109
016.JPG	15955	49561	2	18	0.999694824	0.998873098	0.99959647	0.999416373	0.999359326	0.007960747	0.000640245	0.999130953
017.JPG	28124	37371	31	10	0.999228096	0.999644558	0.999171167	0.999407863	0.998658108	0.015766345	0.01341439	0.998630889
018.JPG	22046	43444	32	14	0.999160767	0.999365367	0.999263962	0.999314664	0.998475262	0.017658148	0.001532395	0.998293577
019.JPG	31811	33670	44	11	0.999450684	0.999288538	0.999489404	0.999389003	0.998780023	0.014973764	0.001767794	0.998230119
020.JPG	12641	52859	27	9	0.999450684	0.999288538	0.999489404	0.999389003	0.998780023	0.014973764	0.001767794	0.998230119
021.JPG	20595	44881	27	33	0.999084473	0.998400233	0.999308771	0.998899502	0.997866019	0.023933288	0.002133465	0.997526869
022.JPG	21088	44419	20	9	0.999557495	0.999573399	0.999549945	0.999561672	0.998962925	0.01254545	0.001336638	0.998812535
023.JPG	24112	41380	29	15	0.999328613	0.99937829	0.999299609	0.999333898	0.998627631	0.016195104	0.001371932	0.998525379
024.JPG	28790	36722	18	6	0.999633789	0.999791638	0.999510071	0.999550854	0.999176352	0.010251192	0.000823478	0.999164121
025.JPG	14571	50933	22	10	0.999511719	0.999314176	0.999568246	0.999441211	0.999022899	0.011789919	0.00097549	0.998550994
026.JPG	28148	37352	10	26	0.999450684	0.999077163	0.999732348	0.999404716	0.99887147	0.013658749	0.001128319	0.998848801
027.JPG	25125	40490	2	9	0.999832153	0.999641919	0.999950498	0.99976208	0.999603344	0.005392459	0.000396618	0.999805067
028.JPG	30272	35238	22	4	0.999603271	0.999687882	0.999376064	0.999621973	0.999115368	0.010761008	0.000884266	0.999110271
029.JPG	26440	39082	8	6	0.999786377	0.999773123	0.999795344	0.999784233	0.999420325	0.00757666	0.00579611	0.999397926
030.JPG	15692	49784	13	47	0.999084473	0.997013787	0.99973894	0.998376364	0.998079211	0.02125353	0.001918167	0.99736832
031.JPG	18308	47203	9	16	0.99961853	0.999126828	0.99980937	0.999468099	0.999206846	0.009955676	0.000792966	0.999015642
032.JPG	20090	45404	23	19	0.999359131	0.999055149	0.999493693	0.999274421	0.998627631	0.016227034	0.001372023	0.998387316
033.JPG	28007	37488	27	14	0.99937439	0.999500375	0.999280288	0.999390331	0.998688585	0.015630397	0.001311181	0.998660597
034.JPG	33735	31743	40	18	0.99911499	0.999466714	0.998741466	0.99910409	0.998079211	0.021777962	0.001920377	0.998077475
035.JPG	24253	41231	47	5	0.999206543	0.999793882	0.998861379	0.999327631	0.998353384	0.01826345	0.001644915	0.998235127
036.JPG	30553	34955	5	23	0.999572754	0.999247776	0.999856979	0.999552378	0.999115368	0.010900443	0.00088445	0.999111374
037.JPG	19618	45909	1	8	0.999862671	0.999592377	0.999978218	0.999785298	0.999572839	0.005765157	0.000427123	0.999490919
038.JPG	31334	34168	20	14	0.999481201	0.999553401	0.999414999	0.99948	0.998901954	0.013441903	0.001097987	0.998899913
039.JPG	27699	37792	8	37	0.999313354	0.998665994	0.99978836	0.999227177	0.998566681	0.016662805	0.001432739	0.998531918
040.JPG	35003	30486	35	12	0.999282837	0.99965729	0.998853249	0.999255269	0.998505734	0.01738693	0.001439867	0.998498641
041.JPG	22481	43006	30	19	0.999252319	0.999155556	0.999302909	0.999229232	0.99844791	0.018128614	0.001554752	0.998275938
042.JPG	30965	34531	17	23	0.998389648	0.999257777	0.999507931	0.999382854	0.998658108	0.016085276	0.001341863	0.998654139
043.JPG	31881	33601	35	19	0.999176025	0.999404389	0.998950448	0.999181918	0.998201056	0.020644173	0.00179868	0.998199837
044.JPG	13282	52189	37	28	0.999008179	0.997896319	0.999291541	0.99850393	0.997896472	0.023149643	0.002101403	0.996754963
045.JPG	30912	34318	290	16	0.995330811	0.999482669	0.991620435	0.995551552	0.990553928	0.080215791	0.009405623	0.990528443
046.JPG	22314	43189	5	28	0.99949646	0.998746755	0.999884243	0.999315499	0.998901954	0.01313907	0.001097517	0.998778097
047.JPG	27222	38259	25	30	0.999160767	0.998899163	0.999346986	0.999123075	0.99823152	0.020465733	0.001768387	0.998179989
048.JPG	32037	33439	16	44	0.999084473	0.998628472	0.999521746	0.999075109	0.998170594	0.020822018	0.00182898	0.998169784
049.JPG	29984	35523	26	3	0.999557495	0.999899957	0.999268615	0.999584286	0.999023899	0.011607678	0.0009757	0.999016891
050.JPG	28686	36794	30	26	0.999145508	0.999094455	0.999185314	0.999139885	0.998109671	0.021679249	0.001890152	0.998808328
051.JPG	30445	35053	31	7	0.999420166	0.99977013	0.999116406	0.999443268	0.998658108	0.015591679	0.00134136	0.998651452
052.JPG	28374	37104	27	31	0.99911499	0.998908643	0.999272845	0.999090744	0.998048753	0.022320978	0.001951165	0.999013552
053.JPG	26067	38976	486	7	0.992477417	0.999731533	0.987684355	0.993707944	0.984917502	0.0114901283	0.0014920369	0.98431705
054.JPG	32462	33018	9	47	0.999145508	0.998552427	0.999727496	0.999140871	0.998261984	0.019620513	0.001737336	0.998261884
055.JPG	13764	51731	27	14	0.99937439	0.998983887	0.999478342	0.999231114	0.998749543	0.014659027	0.001249492	0.998119064
056.JPG	26715	38742	72	7	0.998794556	0.999738044	0.998144999	0.998941521	0.997531094	0.025960731	0.002465447	0.997445272
057.JPG	29827	35630	37	42	0.998794556	0.99859386	0.998962627	0.998778243	0.99743977	0.028310413	0.00260165	0.99741959
058.JPG	25529	39988	10	9	0.999710083	0.999647584	0.999749087	0.999698786	0.999328828	0.008676349	0.000671123	0.999294507
059.JPG	28576	36939	8	13	0.999679565	0.99954528	0.999783474	0.999664377	0.999298331	0.009050862	0.000701641	0.99928673
060.JPG	31566	33934	5	31	0.999450684	0.999018894	0.999852677	0.999435786	0.998749543	0.014871175	0.001250202	0.998747922
061.JPG	19502	46002	10	22	0.999511719	0.998873182	0.999782665	0.999327924	0.998901954	0.013301774	0.001097746	0.998687548
062.JPG	32387	33104	16	29	0.999313354	0.99910538	0.999516908	0.999311144	0.998566681	0.016993393	0.001433236	0.998566529
063.JPG	22165	43342	11	18	0.999557495	0.999188568	0.999746269	0.999467418	0.999023899	0.012061947	0.000975979	0.9989102
064.JPG	23294	42220	5	17	0.999664307	0.999270731	0.999881587	0.999576159	0.999176352	0.010361983	0.000823524	0.999101449
065.JPG	19492	46006	18	20	0.999420166	0.99897499	0.9996089	0.999291945	0.998749543	0.014958603	0.001250266	0.998505136
066.JPG	19626	45881	10	19	0.999557495	0.999032833	0.999782092	0.999407463	0.999054388	0.011659811	0.000945388	0.998873794
067.JPG	32813	32697	16	10	0.999603271	0.999695336	0.999510898	0.999603117	0.999084878	0.011401799	0.000915061	0.999084889
068.JPG	19481	46019	30	6	0.999450684	0.999692102	0.99934852	0.999520311	0.99840987	0.013561782	0.001158042	0.998613957
069.JPG	33399	32106	28	3	0.999526978	0.999910185	0.999128649	0.999519417	0.998962925	0.012228487	0.001036677	0.998962539
070.JPG	21993	43493	45	5	0.999237061	0.999772707	0.99896642	0.999369563	0.998353384	0.018210624	0.001644607	0.99815102
071.JPG	22429	43068	11	28	0.999404907	0.998753173	0.999744655	0.999248914	0.998749543	0.01490269	0.001250035	0.998612052
072.JPG	30372	35093	68	3	0.9998916626	0.999901235	0.998066039	0.998983637	0.997744215	0.023653716	0.002253239	0.997732491
073.JPG	27431	38074	12	19	0.999526978	0.999307832	0.999684924	0.999496378	0.999023899	0.012106984	0.000976026	0.998997495
074.JPG	31583	33399	8	6	0.999786377	0.99981006	0.999764339	0.999787199	0.999481328	0.006895248	0.00051865	0.999408663
075.JPG	29267	36227	4	38	0.999359131	0.998703293	0.999889597	0.999296445	0.998627631	0.015883428	0.001371746	0.998612005
076.JPG	23473	41975	53	35	0.998657227	0.998511145	0.998738936	0.998625041	0.997104988	0.031198754	0.00289361	0.996854886
077.JPG	18797	46696	41	2	0.999343872	0.999893611	0.999122751	0.9				

Tabla B.16: Resultados del método de segmentación PCA con el conjunto de datos LSM

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
1.jpg	6294	297477	68	3361	0.9888	0.6519	0.9998	0.8258	0.9779	0.1207	0.0147	0.6473
10.jpg	5650	296979	34	4537	0.9851	0.5546	0.9999	0.7773	0.9707	0.1460	0.0166	0.5528
11.jpg	7156	298131	143	1770	0.9938	0.8017	0.9995	0.9006	0.9876	0.0809	0.0102	0.7891
12.jpg	4736	298982	565	2917	0.9887	0.6188	0.9981	0.8085	0.9776	0.1288	0.0154	0.5763
13.jpg	5237	298144	281	3538	0.9876	0.5968	0.9991	0.7979	0.9754	0.1339	0.0156	0.5783
14.jpg	6117	296196	17	4870	0.9841	0.5567	0.9999	0.7783	0.9687	0.1539	0.0178	0.5559
15.jpg	5869	297355	35	3941	0.9871	0.5983	0.9999	0.7991	0.9744	0.1323	0.0156	0.5961
16.jpg	6671	296539	76	3914	0.9870	0.6302	0.9997	0.8150	0.9744	0.1361	0.0166	0.6257
17.jpg	6621	294417	86	6076	0.9799	0.5215	0.9997	0.7606	0.9607	0.1867	0.0212	0.5180
18.jpg	3813	301505	210	1672	0.9939	0.6952	0.9993	0.8472	0.9878	0.0765	0.0089	0.6695
19.jpg	0	303341	0	3859	0.9874	0.0000	1.0000	0.5000	0.9752	0.0973	0.0000	0.0000
2.jpg	6310	297392	54	3444	0.9886	0.6469	0.9998	0.8234	0.9775	0.1220	0.0149	0.6434
20.jpg	3576	302244	245	1135	0.9955	0.7591	0.9992	0.8791	0.9911	0.0609	0.0072	0.7215
21.jpg	5712	298799	110	2579	0.9912	0.6889	0.9996	0.8443	0.9826	0.1010	0.0123	0.6799
22.jpg	5728	298860	108	2504	0.9915	0.6958	0.9996	0.8477	0.9831	0.0988	0.0120	0.6868
23.jpg	3424	301866	222	1688	0.9938	0.6698	0.9993	0.8345	0.9876	0.0768	0.0088	0.6419
24.jpg	3458	301840	210	1692	0.9938	0.6715	0.9993	0.8354	0.9877	0.0764	0.0088	0.6451
25.jpg	5790	296404	42	4964	0.9837	0.5384	0.9999	0.7691	0.9679	0.1568	0.0177	0.5363
26.jpg	5971	296105	42	5082	0.9833	0.5402	0.9999	0.7700	0.9672	0.1600	0.0181	0.5382
3.jpg	6342	296647	60	4151	0.9863	0.6044	0.9998	0.8021	0.9730	0.1402	0.0167	0.6010
4.jpg	6383	297142	50	3625	0.9880	0.6378	0.9998	0.8188	0.9764	0.1266	0.0154	0.6346
5.jpg	6451	297203	29	3517	0.9885	0.6472	0.9999	0.8235	0.9772	0.1226	0.0150	0.6453
6.jpg	6506	296626	23	4045	0.9868	0.6166	0.9999	0.8083	0.9739	0.1355	0.0164	0.6153
7.jpg	6491	296196	30	4483	0.9853	0.5915	0.9999	0.7957	0.9711	0.1466	0.0175	0.5899
8.jpg	3485	300861	262	2592	0.9907	0.5735	0.9991	0.7863	0.9816	0.1039	0.0114	0.5498
9.jpg	3618	300924	281	2377	0.9913	0.6035	0.9991	0.8013	0.9828	0.0994	0.0112	0.5763

Tabla B.17: Resultados del método de segmentación Otsu con el conjunto de datos LSM

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
1.jpg	6195	297478	67	3460	0.9885	0.6416	0.9998	0.8207	0.9773	0.1230	0.0149	0.6372
10.jpg	5549	296988	25	4638	0.9848	0.5447	0.9999	0.7723	0.9701	0.1475	0.0166	0.5434
11.jpg	7013	298164	110	1913	0.9934	0.7857	0.9996	0.8927	0.9869	0.0834	0.0105	0.7761
12.jpg	4736	299105	442	2917	0.9891	0.6188	0.9985	0.8087	0.9784	0.1238	0.0146	0.5851
13.jpg	5207	298245	180	3568	0.9878	0.5934	0.9994	0.7964	0.9759	0.1297	0.0150	0.5815
14.jpg	6041	296199	14	4946	0.9839	0.5498	1.0000	0.7749	0.9682	0.1552	0.0178	0.5491
15.jpg	5764	297371	19	4046	0.9868	0.5876	0.9999	0.7937	0.9739	0.1336	0.0156	0.5864
16.jpg	6566	296540	75	4019	0.9867	0.6203	0.9997	0.8100	0.9737	0.1384	0.0167	0.6159
17.jpg	6475	294427	76	6222	0.9795	0.5100	0.9997	0.7549	0.9598	0.1888	0.0212	0.5069
18.jpg	3803	301583	132	1682	0.9941	0.6933	0.9996	0.8465	0.9883	0.0729	0.0085	0.6771
19.jpg	0	303341	0	3859	0.9874	0.0000	1.0000	0.5000	0.9752	0.0973	0.0000	0.0000
2.jpg	6230	297393	53	3524	0.9884	0.6387	0.9998	0.8193	0.9770	0.1239	0.0150	0.6353
20.jpg	3555	302296	193	1156	0.9956	0.7546	0.9994	0.8770	0.9913	0.0591	0.0069	0.7249
21.jpg	5684	298862	47	2607	0.9914	0.6856	0.9998	0.8427	0.9829	0.0980	0.0119	0.6817
22.jpg	5700	298929	39	2532	0.9916	0.6924	0.9999	0.8461	0.9834	0.0955	0.0117	0.6892
23.jpg	3422	301947	141	1690	0.9940	0.6694	0.9995	0.8345	0.9882	0.0729	0.0083	0.6514
24.jpg	3458	301934	116	1692	0.9941	0.6715	0.9996	0.8355	0.9883	0.0717	0.0082	0.6567
25.jpg	5695	296405	41	5059	0.9834	0.5296	0.9999	0.7647	0.9673	0.1586	0.0177	0.5276
26.jpg	5874	296116	31	5179	0.9830	0.5314	0.9999	0.7657	0.9667	0.1612	0.0181	0.5300
3.jpg	6248	296651	56	4245	0.9860	0.5954	0.9998	0.7976	0.9724	0.1420	0.0168	0.5923
4.jpg	6297	297147	45	3711	0.9878	0.6292	0.9998	0.8145	0.9758	0.1282	0.0155	0.6264
5.jpg	6339	297203	29	3629	0.9881	0.6359	0.9999	0.8179	0.9765	0.1252	0.0152	0.6341
6.jpg	6395	296626	23	4156	0.9864	0.6061	0.9999	0.8030	0.9732	0.1380	0.0165	0.6048
7.jpg	6346	296199	27	4628	0.9848	0.5783	0.9999	0.7891	0.9702	0.1495	0.0176	0.5769
8.jpg	3458	300943	180	2619	0.9909	0.5690	0.9994	0.7842	0.9819	0.1007	0.0109	0.5527
9.jpg	3607	301015	190	2388	0.9916	0.6017	0.9994	0.8005	0.9834	0.0954	0.0106	0.5832

Tabla B.18: Resultados del método de segmentación GA con el conjunto de datos LSM

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
1.jpg	8727	297442	103	928	0.9966	0.9039	0.9997	0.9518	0.9442	0.1881	0	0.8943
10.jpg	9537	296980	33	650	0.9978	0.9362	0.9999	0.9680	0.9396	0.2001	0	0.9332
11.jpg	8520	298013	261	406	0.9978	0.9545	0.9991	0.9768	0.9445	0.1872	0	0.9274
12.jpg	7337	298946	601	316	0.9970	0.9587	0.9980	0.9784	0.9497	0.1731	0	0.8889
13.jpg	8587	298119	306	188	0.9984	0.9786	0.9990	0.9888	0.9438	0.1891	0	0.9456
14.jpg	10522	296198	15	465	0.9984	0.9577	0.9999	0.9788	0.9338	0.2155	0	0.9564
15.jpg	9315	297325	65	495	0.9982	0.9495	0.9998	0.9747	0.9408	0.1971	0	0.9433
16.jpg	9661	296494	121	924	0.9966	0.9127	0.9996	0.9561	0.9383	0.2035	0	0.9024
17.jpg	11026	294427	76	1671	0.9943	0.8684	0.9997	0.9341	0.9303	0.2243	0	0.8632
18.jpg	5361	301479	236	124	0.9988	0.9774	0.9992	0.9883	0.9642	0.1313	0	0.9371
19.jpg	3717	303037	304	142	0.9985	0.9632	0.9990	0.9811	0.9742	0.1006	0	0.8929
2.jpg	8875	297357	89	879	0.9968	0.9099	0.9997	0.9548	0.9433	0.1903	0	0.9017
20.jpg	4541	302229	260	170	0.9986	0.9639	0.9991	0.9815	0.9692	0.1161	0	0.9135
21.jpg	8169	298798	111	122	0.9992	0.9853	0.9996	0.9925	0.9475	0.1789	0	0.9723
22.jpg	8169	298863	105	63	0.9995	0.9923	0.9996	0.9960	0.9476	0.1788	0	0.9798
23.jpg	5083	301804	284	29	0.9990	0.9943	0.9991	0.9967	0.9657	0.1270	0	0.9420
24.jpg	5122	301790	260	28	0.9991	0.9946	0.9991	0.9969	0.9656	0.1273	0	0.9468
25.jpg	9890	296400	46	864	0.9970	0.9197	0.9998	0.9598	0.9374	0.2060	0	0.9157
26.jpg	10283	296084	63	770	0.9973	0.9303	0.9998	0.9651	0.9349	0.2125	0	0.9251
3.jpg	9347	296639	68	1146	0.9960	0.8908	0.9998	0.9453	0.9406	0.1976	0	0.8850
4.jpg	8828	297147	45	1180	0.9960	0.8821	0.9998	0.9410	0.9439	0.1888	0	0.8781
5.jpg	8849	297200	32	1119	0.9963	0.8877	0.9999	0.9438	0.9439	0.1889	0	0.8849
6.jpg	9380	296615	34	1171	0.9961	0.8890	0.9999	0.9445	0.9406	0.1976	0	0.8862
7.jpg	9604	296199	27	1370	0.9955	0.8752	0.9999	0.9375	0.9393	0.2011	0	0.8730
8.jpg	5953	300820	303	124	0.9986	0.9796	0.9990	0.9893	0.9601	0.1435	0	0.9331
9.jpg	5852	300901	304	143	0.9985	0.9761	0.9990	0.9876	0.9607	0.1417	0	0.9290

Tabla B.19: Resultados del método de segmentación PSO con el conjunto de datos LSM

Imagen	VP	VN	FP	FN	Accuracy	Sensibilidad	Especificidad	Valor FI	Precisión	PRI	VOI	Dice
1.jpg	8985	297441	104	670	0.9975	0.9306	0.9997	0.9651	0.9426	0.1923	0	0.9207
10.jpg	9779	296965	48	408	0.9985	0.9599	0.9989	0.9799	0.9381	0.2043	0	0.9554
11.jpg	8728	297951	323	198	0.9983	0.9778	0.9989	0.9884	0.9428	0.1917	0	0.9437
12.jpg	7453	298719	828	200	0.9967	0.9739	0.9972	0.9856	0.9475	0.1789	0	0.8788
13.jpg	8686	297871	554	89	0.9979	0.9899	0.9981	0.9940	0.9417	0.1948	0	0.9311
14.jpg	10781	296186	27	206	0.9992	0.9813	0.9999	0.9906	0.9821	0.2197	0	0.9788
15.jpg	9570	297293	97	240	0.9989	0.9755	0.9997	0.9876	0.9390	0.2017	0	0.9660
16.jpg	10199	296452	163	386	0.9982	0.9635	0.9995	0.9815	0.9348	0.2128	0	0.9489
17.jpg	11760	294417	86	937	0.9967	0.9262	0.9997	0.9630	0.9259	0.2357	0	0.9200
18.jpg	5485	301231	484	0	0.9984	1.0000	0.9984	0.9992	0.9619	0.1382	0	0.9189
19.jpg	3734	302893	448	125	0.9981	0.9676	0.9985	0.9831	0.9731	0.1039	0	0.8670
2.jpg	9108	297348	98	646	0.9976	0.9338	0.9997	0.9667	0.9419	0.1942	0	0.9245
20.jpg	4562	302139	350	149	0.9984	0.9684	0.9988	0.9836	0.9685	0.1183	0	0.9014
21.jpg	8290	298646	263	1	0.9991	0.9999	0.9991	0.9995	0.9459	0.1835	0	0.9691
22.jpg	8229	298755	213	3	0.9993	0.9996	0.9993	0.9995	0.9465	0.1816	0	0.9744
23.jpg	5085	301687	401	27	0.9986	0.9947	0.9987	0.9967	0.9649	0.1292	0	0.9224
24.jpg	5122	301670	380	28	0.9987	0.9946	0.9987	0.9967	0.9648	0.1295	0	0.9262
25.jpg	10237	296391	55	517	0.9981	0.9519	0.9998	0.9759	0.9352	0.2117	0	0.9471
26.jpg	10605	296062	85	448	0.9983	0.9595	0.9997	0.9796	0.9328	0.2179	0	0.9521
3.jpg	9698	296635	72	795	0.9972	0.9242	0.9998	0.9620	0.9384	0.2034	0	0.9179
4.jpg	9146	297142	50	862	0.9970	0.9139	0.9998	0.9569	0.9419	0.1941	0	0.9093
5.jpg	9124	297195	37	844	0.9971	0.9153	0.9999	0.9576	0.9421	0.1935	0	0.9119
6.jpg	9678	296608	41	873	0.9970	0.9173	0.9999	0.9586	0.9387	0.2025	0	0.9137
7.jpg	10036	296196	30	938	0.9968	0.9145	0.9999	0.9572	0.9366	0.2081	0	0.9120
8.jpg	6019	300647	476	58	0.9983	0.9905	0.9984	0.9944	0.9586	0.1478	0	0.9185
9.jpg	5917	300685	520	78	0.9981	0.9870	0.9983	0.9926	0.9590	0.1468	0	0.9082

Tabla B.20: Resultados del método de segmentación CP con el conjunto de datos LSM

Imagen	TP	TN	FP	FN	Accuracy	Sensibilidad	Especificidad	Exactitud	PRI	VOI	GCE	DJ
1.jpg	9362	297420	125	293	0.9986	0.9697	0.9996	0.9846	0.9401	0.1988	0	0.9573
10.jpg	9856	296955	58	331	0.9987	0.9675	0.9998	0.9837	0.9375	0.2057	0	0.9620
11.jpg	8773	297962	312	153	0.9985	0.9829	0.9990	0.9909	0.9426	0.1922	0	0.9497
12.jpg	7417	298752	795	236	0.9966	0.9692	0.9973	0.9833	0.9480	0.1777	0	0.8780
13.jpg	8666	297924	501	109	0.9980	0.9876	0.9983	0.9929	0.9421	0.1936	0	0.9342
14.jpg	10785	296186	27	202	0.9993	0.9816	0.9999	0.9908	0.9321	0.2198	0	0.9792
15.jpg	9594	297307	83	216	0.9990	0.9780	0.9997	0.9889	0.9390	0.2019	0	0.9698
16.jpg	10346	296449	166	239	0.9987	0.9774	0.9994	0.9884	0.9339	0.2151	0	0.9623
17.jpg	12214	294407	96	483	0.9981	0.9620	0.9997	0.9808	0.9231	0.2426	0	0.9547
18.jpg	5485	301107	608	0	0.9980	1.0000	0.9980	0.9990	0.9611	0.1405	0	0.9002
19.jpg	3738	302799	542	121	0.9978	0.9686	0.9982	0.9834	0.9725	0.1059	0	0.8494
2.jpg	9521	297340	106	233	0.9989	0.9761	0.9996	0.9879	0.9393	0.2011	0	0.9656
20.jpg	4563	302072	417	148	0.9982	0.9686	0.9986	0.9836	0.9681	0.1196	0	0.8898
21.jpg	8289	298615	294	2	0.9990	0.9998	0.9990	0.9994	0.9457	0.1840	0	0.9655
22.jpg	8232	298690	278	0	0.9991	1.0000	0.9991	0.9995	0.9461	0.1827	0	0.9673
23.jpg	5085	301686	402	27	0.9986	0.9947	0.9987	0.9967	0.9649	0.1293	0	0.9222
24.jpg	5122	301673	377	28	0.9987	0.9946	0.9988	0.9967	0.9648	0.1295	0	0.9267
25.jpg	10304	296386	60	450	0.9983	0.9582	0.9998	0.9790	0.9348	0.2128	0	0.9528
26.jpg	10675	296073	74	378	0.9985	0.9658	0.9998	0.9828	0.9325	0.2188	0	0.9594
3.jpg	10147	296628	79	346	0.9986	0.9670	0.9997	0.9834	0.9356	0.2106	0	0.9598
4.jpg	9687	297130	62	321	0.9988	0.9679	0.9998	0.9839	0.9385	0.2030	0	0.9620
5.jpg	9614	297188	44	354	0.9987	0.9645	0.9999	0.9822	0.9391	0.2016	0	0.9602
6.jpg	10177	296600	49	374	0.9986	0.9646	0.9998	0.9822	0.9356	0.2106	0	0.9601
7.jpg	10566	296196	30	408	0.9986	0.9628	0.9999	0.9814	0.9334	0.2164	0	0.9602
8.jpg	6029	300542	581	48	0.9980	0.9921	0.9981	0.9951	0.9579	0.1499	0	0.9055
9.jpg	5917	300518	687	78	0.9975	0.9870	0.9977	0.9924	0.9579	0.1498	0	0.8855