

SEGMENTACIÓN DE IMÁGENES TÉRMICAS MEDIANTE MÉTODOS DE UMBRALIZACIÓN AUTOMÁTICOS

MARÍA DÍAZ MARTÍN

MÁSTER EN INGENIERÍA DE SISTEMAS Y CONTROL



TRABAJO FIN DE MÁSTER

06 DE SEPTIEMBRE DE 2017

DIRECTOR: GONZALO PAJARES MARTINSANZ

MÁSTER EN INGENIERÍA DE SISTEMAS Y CONTROL
PROYECTO FIN DE MÁSTER ESPECÍFICO PROPUESTO
POR EL TUTOR

**Segmentación de imágenes
térmicas mediante métodos de
umbralización automáticos**

Autor: María Díaz Martín

Director: Gonzalo Pajares Martinsanz



Autorización de difusión

Autorizamos a la Universidad Complutense y a la UNED a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firmado del/los autores:

María Díaz Martín

Fecha: 06 de Septiembre de 2017

Abstract

A novel algorithm for multi-threshold image segmentation to identify elements and objects in thermal imaging is proposed in this Master Thesis' work. More specifically, an automatic algorithm, capable of estimating the minimum number of regions in which an image must be segmented for distinguishing all the relevant objects present in it, has been developed. For doing so, the well-known thresholding Otsu's Method has been applied in order to divide the image histogram in two parts according to the brightness of the image pixels. The adopted solution is based on individually analyzing each histogram region, measuring the amount of relevant information in each of them. The Otsu's Method is recursively applied to those histogram regions which contains enough relevant information for being segmented again. Finally, the different classes present in the image are defined according to the threshold values worked out by Otsu's Method. The proposed algorithm has been evaluated against other state-of-the-art algorithms, such as the classifiers Fuzzy C-means and Bayes, as well as the multi-threshold extension of Otsu's Method.

Keywords

Thresholding, multi-threshold, thermal imaging, pancromatic image, Otsu's Method, image processing, computer vision

Resumen

En este Trabajo Fin de Máster se propone un nuevo método de segmentación multiumbral para la identificación de elementos y/u objetos de interés en imágenes térmicas. Como objetivo final se ha desarrollado un algoritmo totalmente automático capaz de estimar, por semejanza en las intensidades de los píxeles, el número mínimo de regiones necesarias en las que segmentar las imágenes para que dichos objetos queden definidos. Para ello, se hace uso del conocido método de umbralización llamado Método de Otsu que a partir del histograma de intensidades de los píxeles determina un umbral que segmenta la imagen en dos partes. La solución adoptada se basa en analizar de manera individual cada parte del histograma y determinar mediante un factor de decisión si alguna de ellas presenta información relevante como para inferir que es necesario seguir segmentándola. Ello conlleva a la aplicación recursiva del Método de Otsu sobre aquellas regiones del histograma que lo precisen. Finalmente, las distintas clases presentes en la imagen quedan definidas entre los umbrales de intensidad proporcionados por el Método de Otsu. El método propuesto ha sido evaluado frente a otros algoritmos de la literatura como los clasificadores Fuzzy C-means y Bayesiano así como la extensión multi-umbral del Método de Otsu.

Palabras clave

Umbralización, multi-umbral, imágenes térmicas, imágenes pancromáticas, Método de Otsu, procesamiento de imágenes, visión por computación

Índice general

Lista de Figuras	XI
Lista de Tablas	XVII
Agradecimientos	XIX
Abreviaturas	XXI
Símbolos	XXIII
1. Introducción	1
1.1. Objetivos a alcanzar	3
1.2. Organización de la memoria	4
2. Segmentación de imágenes digitales. Estado del Arte	5
2.1. Umbralización.	6
2.1.1. Umbralización Local.	6
2.1.2. Método de Otsu.	7
2.2. Métodos de crecimiento de regiones.	9
2.3. Métodos de agrupamiento.	10
2.3.1. Algoritmo k -means.	10
2.3.2. Algoritmo Fuzzy C-means.	11
2.3.3. Algoritmo ISODATA.	12
2.4. Clasificadores.	13
2.4.1. Fuzzy C-means.	14
2.4.2. Clasificador Bayesiano.	15
2.4.3. k -vecinos más próximos.	16
3. Método propuesto	19
3.1. Preprocesamiento de las imágenes. Suavizado y reducción de ruido. Realce del contraste.	21

3.2. Planteamiento del problema. Detección de los objetos más brillantes en la escena.	22
3.3. Detección del resto de objetos en la escena.	28
3.4. Representación de resultados.	32
4. Resultados	35
4.1. Descripción del conjunto de imágenes térmicas para evaluación. . . .	37
4.2. Análisis cualitativo de los resultados.	40
4.2.1. Comparativa cualitativa con el Método de Otsu Multiumbral.	40
4.2.2. Comparativa cualitativa con los clasificadores Fuzzy C-means y Bayesiano.	64
4.3. Comparativa cuantitativa. Cálculo de errores en la segmentación. . .	85
5. Conclusiones y líneas futuras	91
Referencias	93

Lista de Figuras

3.1.	Comparativa entre la imagen original a) y la imagen tras sufrir la transformación cuadrática b).	23
3.2.	Línea representativa de posibles valores de umbrales comprendidos entre 0 y 255.	23
3.3.	Línea representativa de posibles valores de umbrales para la imagen reducida, IMG_{reduc} , comprendidos entre <i>level_before_before</i> y 255. . . .	24
3.4.	Franja de valores de niveles de grises que conforman el numerador y el denominador del factor de decisión para el umbral <i>L_1</i>	24
3.5.	Franja de valores de niveles de grises que conforman el numerador y el denominador del factor de decisión para el umbral <i>L_2</i>	25
3.6.	Franja de valores de niveles de grises que conforman el numerador y el denominador del factor de decisión para el umbral <i>level_before</i>	25
3.7.	Franja de valores de niveles de grises que conforman el numerador y el denominador del factor de decisión para el umbral <i>L_1</i>	26
3.8.	Digrama resumen de las etapas del método propuesto para la detección de los elementos más brillantes en la escena.	30
3.9.	Digrama resumen de las etapas del método propuesto para la detección de los elementos más oscuros en la escena.	31
4.1.	Data set de imágenes térmicas.	38
4.2.	Data set de imágenes térmicas.	39
4.3.	Comparativa de los resultados tras la segmentación de la imagen a) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 4 niveles, f) MOM para 5 niveles.	45
4.4.	Comparativa de los resultados tras la segmentación de la imagen b) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 5 niveles, f) MOM para 6 niveles.	46
4.5.	Comparativa de los resultados tras la segmentación de la imagen c) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 5 niveles, f) MOM para 7 niveles.	47

4.6.	Comparativa de los resultados tras la segmentación de la imagen d) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 4 niveles, f) MOM para 5 niveles.	48
4.7.	Comparativa de los resultados tras la segmentación de la imagen e) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 4 niveles, f) MOM para 5 niveles.	49
4.8.	Comparativa de los resultados tras la segmentación de la imagen f) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 4 niveles, f) MOM para 5 niveles.	50
4.9.	Croquis para la identificación y situación de los distintos elementos presentes en las imágenes e) y f).	51
4.10.	Comparativa de los resultados tras la segmentación de la imagen g) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 4 niveles, e) MOM para 5 niveles, f) MOM para 6 niveles.	52
4.11.	Comparativa de los resultados tras la segmentación de la imagen h) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 4 niveles, e) MOM para 5 niveles, f) MOM para 6 niveles.	53
4.12.	Comparativa de los resultados tras la segmentación de la imagen i) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 4 niveles, e) MOM para 5 niveles, f) MOM para 6 niveles.	54
4.13.	Comparativa de los resultados tras la segmentación de la imagen j) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 4 niveles, e) MOM para 5 niveles, f) MOM para 6 niveles.	55
4.14.	Comparativa de los resultados tras la segmentación de la imagen k) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 4 niveles, f) MOM para 5 niveles.	56
4.15.	Comparativa de los resultados tras la segmentación de la imagen l) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 4 niveles, f) MOM para 5 niveles.	57
4.16.	Comparativa de los resultados tras la segmentación de la imagen m) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 4 niveles, f) MOM para 5 niveles.	58

4.17. Comparativa de los resultados tras la segmentación de la imagen n) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 5 niveles, e) MOM para 6 niveles, f) MOM para 7 niveles.	59
4.18. Comparativa de los resultados tras la segmentación de la imagen o) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 5 niveles, e) MOM para 6 niveles, f) MOM para 7 niveles.	60
4.19. Comparativa de los resultados tras la segmentación de la imagen p) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 5 niveles, e) MOM para 6 niveles, f) MOM para 7 niveles.	61
4.20. Comparativa de los resultados tras la segmentación de la imagen q) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 5 niveles, e) MOM para 6 niveles, f) MOM para 7 niveles.	62
4.21. Comparativa de los resultados tras la segmentación de la imagen r) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 5 niveles, e) MOM para 6 niveles, f) MOM para 7 niveles.	63
4.22. Comparativa de los resultados tras la segmentación de la imagen a) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	67
4.23. Comparativa de los resultados tras la segmentación de la imagen b) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	68
4.24. Comparativa de los resultados tras la segmentación de la imagen c) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	69
4.25. Comparativa de los resultados tras la segmentación de la imagen d) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	70
4.26. Comparativa de los resultados tras la segmentación de la imagen e) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	71
4.27. Comparativa de los resultados tras la segmentación de la imagen f) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	72

4.28. Comparativa de los resultados tras la segmentación de la imagen g) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	73
4.29. Comparativa de los resultados tras la segmentación de la imagen h) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	74
4.30. Comparativa de los resultados tras la segmentación de la imagen i) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	75
4.31. Comparativa de los resultados tras la segmentación de la imagen j) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	76
4.32. Comparativa de los resultados tras la segmentación de la imagen k) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	77
4.33. Comparativa de los resultados tras la segmentación de la imagen l) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	78
4.34. Comparativa de los resultados tras la segmentación de la imagen m) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	79
4.35. Comparativa de los resultados tras la segmentación de la imagen n) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	80
4.36. Comparativa de los resultados tras la segmentación de la imagen o) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	81
4.37. Comparativa de los resultados tras la segmentación de la imagen p) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	82
4.38. Comparativa de los resultados tras la segmentación de la imagen q) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.	83

4.39. Comparativa de los resultados tras la segmentación de la imagen r) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano	84
4.40. Ground-truths de 11 imágenes del data set.	88
4.41. Gráfico con los errores en la clasificación (%) cometidos por los distintos algoritmos para 11 de las imágenes del data set.	89

Lista de Tablas

4.1. Tabla con los errores en la clasificación (%) cometidos por los distintos algoritmos para 11 de las imágenes del data set.	89
---	----

Agradecimientos

Me gustaría agradecer a mi tutor Gonzalo Pajares por brindarme la oportunidad de realizar este trabajo bajo su supervisión, aún a sabiendas que todo su desarrollo se haría desde la distancia a través de unos largos, otras veces más breves, correos electrónicos. Gracias por su paciencia y buena disposición, ha sido una experiencia muy enriquecedora que me ha ayudado no sólo a culminar una etapa necesaria para la finalización de los estudios de Máster sino a ampliar mi rango de conocimiento a otras áreas del procesado de imágenes.

Por supuesto, agradecer a mi familia y a amigos. Sin ustedes, nada de ésto sería posible. Gracias por esas lluvias de ideas, gracias por escucharme aunque muchas veces sólo fueran quejas sin sentido.

Abreviaturas

- TFM** Trabajo Fin de Máster
- SSE** Suma de los errores cuadráticos (*Sum of Squared Errors*)
- KNN** Algoritmo de clasificación *k*-vecinos más próximos (*k-nearest neighbors*)
- MO** Método de Otsu
- MOB** Método de Otsu clásico o bimodal
- MOM** Método de Otsu Multiumbral
- RGB** Rojo, verde y azul (*Red, Green, Blue*)
- TPR** Ratio de verdaderos positivos (*True positive rate*)
- TNR** Ratio de verdaderos negativos (*True negative rate*)
- FNR** Ratio de falsos negativos (*False negative rate*)
- FPR** Ratio de falsos positivos (*False positive rate*)
- TP** Verdaderos positivos (*True positives*)
- TN** Verdaderos negativos (*True negatives*)
- FP** Falsos positivos (*False positives*)
- FN** Falsos negativos (*False negatives*)
- PPV** Precisión (*Positive predictive value*)
- ACC** Exactitud (*Accuracy*)

Símbolos

i, j	sub-índices
μm	micrómetros
T	umbral de nivel de intensidad
T_{opt}	umbral óptimo
L	número total de niveles de grises en una imagen
n_i	número total de píxeles con intensidad i
N	número total de píxeles de la imagen
p_i	probabilidad de ocurrencia del nivel de gris i
C_i	grupo i de píxeles con intensidades comprendidas entre T_i y T_j
w_i	probabilidad acumulativa de niveles de grises que conforman C_i
μ_i	intensidad media del grupo C_i
μ_T	intensidad media total de la imagen
μ	media geométrica
σ_B^2	varianza inter-clase
σ	desviación estándar
$max()$	función maximizar
$max(:)$	valor máximo
$min(:)$	valor mínimo
k	número de regiones o clusters a segmentar la imagen
c_j	centroide del cluster j
$\mu_{i,j}$	matriz de pesos
x_i	píxel i de una imagen X
r_i	píxel i de una imagen X
$dist()$	distancia Euclidea
$p()$	función de densidad de probabilidad
p	número de atributos de una variable
$d_M()$	distancia de Mahalanobis

C_i^{-1}	inversa de la matriz de covarianzas de la clase i
IMG	Imagen bidimensional
IMG_{white}	Imagen transformada para detección de los elementos más brillantes
IMG_{reduc}	Sub-imagen conformada por píxeles con niveles de grises entre T_i y T_j
IMG_{dark}	Imagen transformada inversa para detección de los elementos más oscuros
IMG_{result}	Imagen resultado con las k regiones etiquetadas
(x, y)	coordenadas de un píxel en un imagen bidimensional
omega	histograma acumulativo
v_{up}, v_{down}	niveles de grises
<i>level_before</i>	niveles de grises
<i>level_before_before</i>	niveles de grises
L_1, L_2	niveles de grises
level_{final}	vector cuyas componentes son niveles de intensidad
<i>round()</i>	función redondear
pos	vector de posiciones de ciertos píxeles en la imagen original

Capítulo 1

Introducción

Todos los cuerpos emiten cierta cantidad de radiación de energía infrarroja en función de su temperatura, siendo ésta mayor a medida que la temperatura aumenta. Las cámaras térmicas o infrarrojas son dispositivos sensibles a este tipo de radiación, operando en el rango del espectro electromagnético correspondiente a la zona del infrarrojo térmico (3-14 μm). Dicha zona del espectro queda fuera del rango visible por el ojo humano, por lo que los sensores térmicos tratan de emular la cantidad de energía percibida en forma de imágenes luminosas, normalmente monocromáticas, que muestran las áreas más calientes de un cuerpo en blanco y las menos en negro, correspondiendo los matices grises con las temperaturas intermedias entre los límites térmicos. En ocasiones, existen cámaras infrarrojas que procesan las imágenes para mostrarlas coloreadas puesto que son más fáciles de interpretar a simple vista. Sin embargo, dichos colores no se corresponden con la realidad, sino que se asignan arbitrariamente en función del rango de intensidad dentro de una determinada longitud de onda infrarroja. El uso de las cámaras térmicas como instrumento de diagnóstico para el mantenimiento preventivo, en vigilancia y defensa, entre muchas otras aplicaciones, se ha extendido notablemente en los últimos años puesto que la termografía permite detectar anomalías que con frecuencia son imperceptibles a simple vista. Sin embargo, la resolución del tipo de imágenes capturadas por estos dispositivos es bastante baja, por lo que el proceso de detectar y delimitar objetos de interés presentes en ellas se complica considerablemente.

Las técnicas de segmentación de imágenes digitales tratan de dividir la imagen en ciertas regiones hasta un nivel de subdivisión en el que se aislen los objetos de interés presentes en ellas. Algunos algoritmos de segmentación dividen las imágenes

basándose en cambios bruscos del nivel de gris ya sea mediante detección de bordes, líneas o puntos aislados. Otros, en cambio, se basan en la búsqueda de regiones con valores similares de intensidad ya sea mediante técnicas de crecimiento de regiones, mediante métodos de umbralización o mediante técnicas de agrupamiento. Otras técnicas hacen uso de las máquinas de aprendizaje que permiten aprender de la experiencia con respecto a alguna clase de tareas y una medida de desempeño. Sin embargo, la mayoría de estos algoritmos requieren conocer de antemano el número de regiones o partes en las que segmentar la imagen, por lo que la autonomía que caracteriza a muchos de ellos cuando se pretende realizar una segmentación bimodal se pierde al intentar resolver un problema basado en la detección de múltiples regiones. Una estimación por defecto del número de regiones conlleva a que algunos objetos se queden camuflados en segundo plano, mientras que una estimación por exceso daría lugar a regiones que no aportan información relevante y por tanto, podrían fusionarse con otras que representan al mismo ente.

En este Trabajo Fin de Máster (TFM) se plantea un nuevo método de segmentación basado en técnicas de umbralización que sea capaz de estimar de manera automática el número mínimo de regiones necesarias para definir a todos los objetos de interés presentes en imágenes digitales, bajo la premisa de englobar en una única región agrupaciones de píxeles con características similares. Las técnicas de segmentación por umbralización tratan de estimar múltiples umbrales de tal manera que los píxeles que se encuentren entre cada par de niveles T_i y T_j representan a una clase de objeto. En el caso concreto de detección de objetos en imágenes térmicas, aspecto a abordar en este TFM, es fácil deducir la viabilidad de la utilización de este tipo de técnicas puesto que aquellos píxeles que forman parte de un mismo ente presentan niveles de intensidad similares ya que la cantidad de energía térmica emitida por éste difiere de la emitida por sus alrededores. A pesar de la aparente facilidad del proceso de umbralización, en la realidad es un proceso bastante complejo debido a la amplia gama de circunstancias y entornos que son únicos para cada imagen, como son: presencia de ruido, histogramas planos o iluminación inadecuada, siendo esta última uno de los principales desafíos de los algoritmos de segmentación por umbral sobre todo en aplicaciones en tiempo real.

Sin embargo, estos aspectos no son los únicos a tener en cuenta en la detección y delimitación de objetos en imágenes térmicas, puesto que hay que hacer frente a un fenómeno muy común en la naturaleza denominado inversión térmica. Ciertos elementos, como los metales, durante el día aparecen muy calientes y por consiguiente,

con mayor reflectividad, mientras que por la noche su comportamiento es totalmente opuesto, caracterizándose por valores de intensidad muy bajos. En definitiva, este fenómeno obliga a generar un algoritmo que estudie el histograma al completo para determinar tanto los objetos más brillantes de la escena como también los más oscuros, de tal manera que elementos con esta cualidad tan peculiar sean detectables en cualquier momento del día.

1.1. Objetivos a alcanzar

En este TFM se pretende desarrollar un algoritmo de segmentación de imágenes en escala de grises, concretamente imágenes con información del infrarrojo, que cumpla los siguientes objetivos y requisitos:

1. Determinar de manera automática el mínimo número de umbrales necesarios para definir los distintos objetos presentes en una escena. Con ello se consigue destacar los elementos de interés y unificar en una única región el resto de zonas representativas del entorno o *background*, obteniéndose unos resultados más homogéneos y concisos donde los elementos de interés destacan sobre el resto. Además, se evita la necesidad tener un conocimiento previo de la escena en relación al número de elementos presentes, el cual determina el número de regiones en los que segmentar la imagen.
2. Definir todos los objetos presentes en la escena, ya sean aquellos caracterizados por agrupaciones de píxeles con altos niveles de grises como por objetos con baja emitancia y por tanto, caracterizados por tonalidades más oscuras. La detección de este último conjunto de elementos presenta un reto puesto que las imágenes térmicas se caracterizan por altas concentraciones de píxeles con bajos niveles de intensidad.
3. Los resultados de la segmentación obtenidos deben ser equiparables a los obtenidos por los principales algoritmos de segmentación del estado del arte. Para ello, se realiza un exhaustivo estudio cualitativo y cuantitativo de los resultados con el objetivo de compararlos con los arrojados por tres conocidos métodos de la literatura.

1.2. Organización de la memoria

Este documento está organizado en los siguientes capítulos:

1. Capítulo 1: Capítulo actual introductorio donde se ha planteado el problema a tratar así como los objetivos a alcanzar.
2. Capítulo 2: Presentación de un breve resumen del estado de arte en técnicas de segmentación en múltiples regiones típicamente utilizadas en imágenes digitales.
3. Capítulo 3: Descripción detallada del método propuesto.
4. Capítulo 4: Análisis y evaluación cualitativa y cuantitativa de los resultados. Dichos resultados serán comparados con algunas de las alternativas presentes en la literatura.
5. Capítulo 5: Resumen de las conclusiones extraídas así como una breve descripción de posibles líneas futuras de investigación.

Capítulo 2

Segmentación de imágenes digitales. Estado del Arte

La segmentación en el campo de la Visión Artificial es un método de procesamiento de imágenes digitales muy comúnmente utilizado cuyo objetivo es dividir dichas imágenes en varios grupos o regiones de píxeles con características visuales similares, a los cuales se les asigna una etiqueta común representativa de la región a la que pertenecen. Por lo tanto, es fácil deducir que una de las principales aplicaciones de este método es la localización de objetos en imágenes y la delimitación de sus bordes, tema a abordar en este TFM. El problema de la segmentación de imágenes digitales ha sido ampliamente analizado y estudiado, por lo que existen muchos algoritmos en el estado del arte que tratan de abordarlo. En este capítulo se recogen algunos de estos métodos de segmentación los cuales se agrupan en las siguientes categorías:

1. Umbralización (*Thresholding*).
2. Métodos de crecimiento de regiones (*Region Growing*).
3. Métodos de agrupamiento (*Clustering*).
4. Clasificadores.

2.1. Umbralización.

Los métodos de umbralización tratan de determinar un valor de intensidad, llamado umbral (*threshold*), que separe las clases deseadas en una imagen. La segmentación se logra etiquetando todos los píxeles que presenten mayor intensidad que la fijada por el umbral T con un valor determinado y el resto de píxeles con otro. Es una técnica bastante efectiva para obtener los diferentes objetos presentes en una imagen que se caracterizan por intensidades contrastantes u otras características diferenciables. Los distintos métodos de umbralización se basan en el estudio de los picos y valles presentes en el histograma unidimensional de las imágenes. Normalmente, los umbrales de segmentación suelen coincidir con los valles de éste puesto que representan la frecuencia menor entre dos picos, es decir, entre dos distribuciones [1]. Cuando se trata de obtener múltiples objetos con diferentes niveles de grises, determinar más de un umbral es necesario y a ello se le llama umbralización multiumbral.

Una de las principales desventajas de los métodos de umbralización es que muchos de ellos no tienen en cuenta la información espacial de la imagen, lo que conlleva a que sean métodos sensibles al ruido y a zonas mal iluminadas. Otros métodos, sin embargo, no sólo tienen en cuenta las características globales de la imagen sino que también consideran alguna característica local, como el valor medio de los píxeles vecinos, o incluso la posición.

2.1.1. Umbralización Local.

En ocasiones, intentar encontrar un umbral óptimo global para toda la imagen que de resultados satisfactorios no es posible ya sea debido a que estas imágenes no presentan un fondo (*background*) constante, existen diversidad de objetos y elementos con diferentes niveles de grises, etc. En estos casos, es más conveniente encontrar diversos umbrales para distintas partes de la imagen. Para ello, la imagen se divide en múltiples sub-imágenes y luego se calcula un umbral para cada una considerando que todas presentan un histograma bimodal. Luego, los umbrales obtenidos se interpolan para obtener un umbral global. Un ejemplo de aplicación se encuentra en [2] donde se desarrolla un método de umbral basado en principios estadísticos y heurísticos para detectar límites en imágenes radiográficas. Cada región local de la imagen que contiene una porción de frontera entre dos regiones se caracteriza por

una mezcla de dos distribuciones normales de intensidad. Los umbrales se establecen dinámicamente de acuerdo con las características locales estimadas a partir de los histogramas de intensidad observados.

2.1.2. Método de Otsu.

Dentro de los algoritmos de segmentación por umbralización, el Método de Otsu (MO) es uno de los más conocidos y ampliamente utilizados [3]. Una de las principales ventajas que presenta es que es un método no paramétrico y totalmente automático puesto que no necesita de supervisión humana ni información previa de la imagen antes de su procesamiento. Además, es muy fácil extrapolarlo al cálculo de múltiples umbrales aunque a medida que el número de éstos aumenta, lo hace también la carga computacional. Para la búsqueda del umbral óptimo, el MO busca maximizar la varianza inter-clase y minimizar la varianza intra-clase. Consideremos que existen L niveles de grises en la imagen, que el número total de píxeles con intensidad i es n_i y que el número total de píxeles de la imagen es $N = n_1 + n_2 + \dots + n_L$, la probabilidad de ocurrencia del nivel de gris i es:

$$p_i = \frac{n_i}{N} \quad (2.1)$$

En el caso de umbralización en dos niveles, los píxeles son divididos en dos grupos: C_1 , píxeles con niveles de grises menores o iguales a T y C_2 , píxeles con niveles de grises mayores a T . Por tanto, la distribución de probabilidades de cada clase es:

$$C_1 = \frac{p_1}{w_1(T)} \quad (2.2)$$

$$C_2 = \frac{p_2}{w_2(T)} \quad (2.3)$$

siendo:

$$w_1 = \sum_{i=1}^T p_i \quad (2.4)$$

$$w_2 = \sum_{i=T+1}^L p_i \quad (2.5)$$

$$w_1 + w_2 = 1 \quad (2.6)$$

A su vez las intensidades medias para cada clase así como la intensidad media total son:

$$\mu_1 = \sum_{i=1}^T \frac{i \cdot p_i}{w_1(T)} \quad (2.7)$$

$$\mu_2 = \sum_{i=T+1}^L \frac{i \cdot p_i}{w_2(T)} \quad (2.8)$$

$$w_1 \cdot \mu_1 + w_2 \cdot \mu_2 = \mu_T \quad (2.9)$$

Otsu definió la varianza inter-clase como:

$$\sigma_B^2 = w_1(\mu_1 - \mu_T)^2 + w_2(\mu_2 - \mu_T)^2 \quad (2.10)$$

Para una umbralización en dos niveles, Otsu consideró que el umbral óptimo es aquel que maximiza σ_B^2 .

$$T_{opt} = \max(\sigma_B^2(T)) \quad (2.11)$$

Finalmente, el problema se puede fácilmente extrapolar a múltiples umbrales. Por ejemplo: asumamos que se quiere segmentar la imagen en 3 regiones (C_1, C_2 y C_3), por lo que se necesitan estimar dos umbrales $1 \leq T_1 \leq T_2 \leq L$. Ahora la varianza inter-clase σ_B^2 es función de dos variables T_1 y T_2 cuyos valores óptimos surgen de maximizar σ_B^2 :

$$(T_{opt_1}, T_{opt_2}) = \max_{1 \leq T_1 \leq T_2 \leq L} (\sigma_B^2(T_1, T_2)) \quad (2.12)$$

Muchas mejoras y optimizaciones del MO existen en la literatura. En [4] se propone una estrategia para combinar el MO con distintos algoritmos de extracción de bordes para delimitar el contorno de imágenes médicas tomadas por rastreadores de tomografía computerizada. En [5] y [6] se muestra una extensión recursiva del MO para binarización de manuscritos con el objeto de extraer los elementos más oscuros los cuales corresponden con las palabras presentes en dichos documentos. En concreto, en [5] se aplica el MO para segmentar la imagen y eliminar las áreas más brillantes, normalmente relacionadas con el fondo o (*background*). Luego, se va aplicando reiteradamente otra vez el MO sobre píxeles de la imagen cada vez más oscuros hasta obtener un umbral que separe correctamente el texto del fondo. En [6], en cambio, se utiliza reiteradamente el MO sobre los píxeles pertenecientes al *background* con el objetivo de extraer de ellos la mayor cantidad de información del texto posible. Para ello, se le aplica a la imagen original un filtro de la mediana para extraer la información del *background* general. Luego, se aplica el MO para separar las zonas más oscuras, que corresponden con el texto, de las las zonas más brillantes. Posteriormente, se aplica el MO reiteradamente sobre las regiones más brillantes de la imagen para extraer posibles píxeles que puedan pertenecer al texto. En [7] se propone una modificación a la varianza inter-clase con el objetivo de obtener una versión del MO multiumbral más rápida y eficaz computacionalmente. En [8] se va más allá, utilizando el MO en imágenes RGB (*Red*, *Green*, *Blue*) con el objetivo de reducir los niveles de color .

2.2. Métodos de crecimiento de regiones.

Los métodos de segmentación basados en crecimiento de regiones son métodos iterativos que agrupan regiones de la imagen que están conectadas según un cierto criterio o condición predefinido basado en información de la intensidad y/o bordes de la imagen, entre otros. En su forma más sencilla, es necesario para la inicialización del método definir un punto semilla (*seed point*) que es el origen de la región a definir. Dicha región va creciendo en función de ciertas propiedades de homogeneidad que cumplan los píxeles vecinos al punto semilla, como por ejemplo, que la

distancia entre la media de los píxeles que van conformando la región y un nuevo píxel a evaluar sea inferior a un umbral [9], [10].

Una de las principales desventajas de este método es que el resultado final de la segmentación depende significativamente del punto semilla escogido el cual es seleccionado manualmente por el usuario. En contraposición, es menos sensible al ruido que los métodos de umbralización aunque, dependiendo del criterio de homogeneidad a utilizar, puedan quedar áreas sin conectar.

2.3. Métodos de agrupamiento.

Las técnicas de *clustering* o agrupamiento tratan de identificar de forma automática agrupaciones o *clústeres* de elementos de acuerdo a una alta medida de similitud entre elementos del mismo grupo y baja para elementos pertenecientes a otros grupos. Su función es la misma que la de los métodos clasificadores que se estudiarán en la siguiente sección con la salvedad de que no necesitan datos de entrenamiento, por lo que son métodos no supervisados. Muchos algoritmos de agrupamiento se encuentran en el estado del arte, sin embargo, en esta sección se centra la atención en tres de los más populares: *k*-means, fuzzy C-means e ISODATA (*Iterative Self-Organizing Data Analysis Techniques*).

2.3.1. Algoritmo *k*-means.

El algoritmo *k*-means [11], [12] es uno de los algoritmos de agrupamiento más conocido y más ampliamente utilizado. Este método trata de dividir la imagen en *k* regiones basándose en la minimización de la distancia interna. Para ello, se realizan los siguientes pasos:

1. Se eligen de manera aleatoria *k* centroides en la imagen, los cuales definen las *k* regiones deseadas.
2. Cada píxel de la imagen es asignado al *clúster* más cercano en función de la mínima distancia Euclídea.
3. Se actualizan los centroides calculando la media de todos los píxeles pertenecientes a cada *clúster*.

4. Se repiten los pasos descritos en 2 y 3 hasta que ya no haya variación en los centroides, lo que significa que la suma de las distancias ya ha sido minimizada.

La principal desventaja de este método es que el número de *clústeres*, k , debe ser conocido a priori por el usuario puesto que es un parámetro de entrada del algoritmo. Además, al procesar los patrones secuencialmente la configuración inicial de las regiones queda determinada por los primeros patrones procesados y su comportamiento depende enormemente del parámetro k , aumentando con éste la carga computacional.

2.3.2. Algoritmo Fuzzy C-means.

El agrupamiento difuso, *fuzzy clustering*, es un tipo de algoritmo de agrupamiento donde cada elemento tiene un grado de pertenencia difuso a cada grupo. Este tipo de algoritmos surge ante las deficiencias de los algoritmos de agrupamiento exclusivos que consideran que un elemento sólo se puede agrupar exclusivamente a un *clúster* y no consideran la posibilidad de que pueda asemejarse al resto de elementos de los otros grupos. Una solución para este problema consiste en caracterizar la similitud de cada elemento a cada uno de los grupos a través de una función de pertenencia que toma valores entre cero y uno. Los valores cercanos a uno indican una mayor similitud, mientras que los cercanos a cero indican una menor similitud. Por lo tanto, el problema del agrupamiento difuso se reduce a encontrar una caracterización de este tipo que sea óptima.

El algoritmo Fuzzy C-means [13], [14], [15] es uno de los principales algoritmos utilizados en el agrupamiento difuso. Para su interpretación, consideremos que $\mathbf{X} = x_1, x_2, \dots, x_n$ es el conjunto de píxeles que conforman la imagen, que k es el número de *clústeres* deseados y que $\mathbf{c} = c_1, c_2, c_j, \dots, c_k$ son los centroides de dichas regiones. Se asume por tanto que w es una matriz de pesos que muestra el grado de similitud de cada elemento a cada *clúster*:

$$\mathbf{w} = \begin{bmatrix} w_{1,1} & w_{n,1} \\ w_{1,k} & w_{n,k} \end{bmatrix}$$

$$w_{i,j} \in [0, 1] \tag{2.13}$$

Los pasos a seguir en el algoritmo Fuzzy C-means son:

1. Inicializar la matriz de pesos de manera aleatoria.
2. Calcular los centroides mediante la fórmula:

$$c_j = \frac{\sum_{i=1}^n w_{ij}^p \cdot x_i}{\sum_{i=1}^n w_{ij}} \quad (2.14)$$

3. Actualizar la matriz de pesos:

$$w_{ij} = \frac{(1/\text{dist}(x_i, c_j^2))^{\frac{1}{1-p}}}{\sum_{j=1}^k (1/\text{dist}(x_i, c_j^2))^{\frac{1}{1-p}}} \quad (2.15)$$

4. Repetir los pasos descritos en 2 o 3 hasta que se cumpla una función objetivo que suele ser la función de mínimo error cuadrático (SSE):

$$SSE = \sum_{j=1}^k \sum_{i=1}^n w_{ij}^p \text{dist}(x_i, c_j)^2 \quad (2.16)$$

siendo $\text{dist}()$ la distancia Euclídea y p un factor que especifica el impacto del peso w_{ij} . Una detallada discusión sobre el número de iteraciones para la condición de parada de este algoritmo se encuentra en [16] y [17].

2.3.3. Algoritmo ISODATA.

El algoritmo ISODATA [18] es un método iterativo cuyo objetivo último es la minimización del criterio SSE. Al igual que en el algoritmo k -means los patrones se procesan repetidamente y en cada iteración se asignan al grupo más cercano, actualizándose los centroides. Las diferencias más relevantes con el método k -means es que incorpora una serie de heurísticas con tres objetivos:

1. Eliminar agrupaciones poco numerosas.
2. Mezclar agrupaciones cercanas.
3. Dividir agrupaciones dispersas.

Para ello, es necesario definir un umbral del número de patrones para la eliminación de un agrupamiento (Θ_N), un umbral de distancia para la unión de agrupamientos (Θ_c), un umbral de desviación típica para la división de un agrupamiento (Θ_s), un número máximo de agrupamientos (k), el máximo número de agrupamientos que pueden mezclarse en una iteración (L) y el máximo número de iteraciones permitidas (I).

En líneas generales, los pasos a seguir en este algoritmo son:

1. En cada iteración se procesan todos los píxeles, asignándose al grupo más cercano y actualizándose los centroides.
2. Se eliminan los grupos que tengan menos de Θ_N miembros.
3. Se aplican una serie de heurísticas para aumentar o reducir el número de agrupamientos, como son:
 - a) Se divide un agrupamiento por la variable de máxima varianza si ésta supera el umbral Θ_s .
 - b) Se unen agrupaciones si las distancias entre los centroides son menores al umbral Θ_c .

2.4. Clasificadores.

Los métodos de clasificación son técnicas de reconocimiento de patrones que buscan dividir un espacio característico derivado de una imagen, por ejemplo los niveles de grises o intensidades de ésta, en función de un conjunto de datos etiquetados y conocidos. Los métodos de clasificación son técnicas de aprendizaje supervisadas puesto que se basan en la disponibilidad de áreas de entrenamiento de las que se conoce a priori la clase a la que pertenecen. Es decir, estos métodos necesitan ser entrenados previamente utilizando un conjunto de datos etiquetados para generar unos modelos característicos de cada clase, los cuales luego se utilizan para adjudicar/clasificar nuevos datos no conocidos en las distintas clases disponibles.

Algunos de los métodos de clasificación a estudiar en esta sección son: Fuzzy C-means, clasificador Bayesiano y k -vecinos más próximos (KNN).

2.4.1. Fuzzy C-means.

El algoritmo clasificador Fuzzy C-means es un algoritmo iterativo de clasificación de imágenes basado en la técnica de agrupamiento Fuzzy *Clustering* expuesta en la sección anterior. Este algoritmo está compuesto por dos fases: entrenamiento no supervisado y clasificación. La etapa de entrenamiento no supervisado se realiza aplicando el algoritmo de *clustering* Fuzzy C-means que, como ya se explicó, trata de estimar los centroides característicos de cada región presente en una imagen utilizando la distancia espectral mínima para asignar cada píxel a un centroide candidato. Para ello, el número de *clústeres* en los que la imagen se va a particionar debe ser conocido previamente puesto que representa un parámetro de entrada al algoritmo.

Una vez obtenido el conjunto de centroides resultantes, su valor es utilizado para realizar la clasificación de las imágenes deseadas. El proceso de clasificación consiste en la obtención del valor de pertenencia de cada uno de los píxeles que componen la imagen con respecto a cada una de las clases definidas. Para ello se calcula la distancia de cada píxel al centro correspondiente de cada clase y se asigna luego a la clase que mayor valor de pertenencia haya obtenido.

En modo resumen, los pasos a realizar en el método de clasificación Fuzzy C-means son:

1. Aplicar el algoritmo de agrupamiento no supervisado Fuzzy C-means, especificando el número de clases a determinar, k . Como parámetros de salida se obtienen los k centroides característicos de cada región.
2. Calcular la distancia Euclídea de cada muestra a cada centroide.
3. Calcular la función de pertenencia de la muestra a cada uno de las clases sustituyendo el valor de la distancia Euclídea dada en el paso 1 en la ecuación 2.15.
4. Clasificar la muestra con aquel *clúster* que obtenga un mayor valor de la función de pertenencia.

2.4.2. Clasificador Bayesiano.

El algoritmo de clasificación Bayesiana [19], [20], [21], [22], [23] se basa en que el problema de la decisión se enfoque en términos probabilísticos y que todas las probabilidades relevantes resulten conocidas. Se trata de un método supervisado que en una de sus variantes utiliza el método de agrupamiento difuso como etapa de entrenamiento inicial para obtener los centroides característicos de cada clase presente en la imagen. La clasificación de cada píxel se determina según la probabilidad de pertenecer a cada una de las clases. Esa probabilidad vendrá dada en nuestro caso por la distancia de Mahalanobis, que a su vez se calcula con los datos obtenidos en el entrenamiento.

La probabilidad de que un píxel pertenezca a cualquiera de las clases es a priori la misma y su suma es la unidad. Sea y una variable aleatoria que designa la pertenencia de un píxel a una de las clases en las que hay que dividir la imagen, se puede decir que existe una probabilidad a priori $p(y = c_i)$ de que el píxel pertenezca a cada clase c_i . Como la única información disponible son esas probabilidades a priori se puede afirmar que una muestra x cualquiera va a pertenecer a aquella clase que genere la mayor probabilidad. Considerando a x como una variable aleatoria continua cuya distribución depende del tipo de clase, $p(x/y = c_i)$ representa la función de densidad de probabilidad condicionales para x dado que el píxel pertenezca a la clase c_i y suponiendo que se conocen tanto las probabilidades a priori como estas funciones de densidad de probabilidad, es posible determinar para una observación dada x la probabilidad de que esa observación pertenezca a una determinada clase mediante la regla de Bayes:

$$p(y = c_i/x) = \frac{p(x/y = c_i)P(y = c_i)}{p(x)} \quad (2.17)$$

donde:

$$p(x) = \sum_{i=1}^k P(x/y = c_i)P(y = c_i) \quad (2.18)$$

La regla de Bayes muestra cómo la observación del valor x cambia las probabilidades a priori a las probabilidades a posteriori. Una vez que se determinan esas probabilidades a posteriori, $P(y = c_i/x)$, se utiliza la siguiente regla para clasificar:

$$x \in c_i \quad \text{si} \quad P(y = c_i/x) > P(y = c_j/x) \quad \forall i \neq j \quad i, j = 1, 2, k \quad (2.19)$$

Una forma alternativa de expresar la pertenencia de x es:

$$x \in c_i \quad \text{si} \quad P(y = c_i/x)P(y = c_i) > P(y = c_j/x)P(y = c_j) \quad \forall i \neq j \quad i, j = 1, 2, k \quad (2.20)$$

Generalmente la distribución de densidad de probabilidad más ampliamente utilizada es la Gaussiana o Normal. En el caso de que las probabilidades a priori sean iguales para todas las clases, la distancia de Mahalanobis se puede usar como función discriminante, por lo que la regla anterior quedaría:

$$x \in c_i \quad \text{si} \quad d_M^2(x, m_i) < d_M^2(x, m_j) \quad \forall i \neq j \quad i, j = 1, 2, k \quad (2.21)$$

siendo m_i y m_j las medias de las clases c_i y c_j respectivamente y d_M la distancia de Mahalanobis que se define como:

$$d_M^2(x, m_i) = (x - m_i)^t C_i^{-1} (x - m_i) \quad (2.22)$$

siendo C_i^{-1} la inversa de la matriz de covarianzas de la clase i .

Durante el entrenamiento, como paso previo a la clasificación Bayesiana, se calculan las medias y matrices de covarianza de las muestras para cada clase.

2.4.3. k -vecinos más próximos.

El KNN [24], [22], [23] es un método de clasificación supervisada no paramétrico que estima el valor de la función de densidad de probabilidad o directamente la probabilidad a posteriori de que un elemento x pertenezca a la clase c_i a partir de la información proporcionada por el conjunto de ejemplos cercanos en el espacio de los elementos. Por tanto, este algoritmo supone que los vecinos más cercanos nos dan la mejor clasificación.

Los ejemplos de entrenamiento, \mathbf{x}_i , son vectores en un espacio característico multidimensional donde cada ejemplo está descrito en términos de p atributos considerando k clases para la clasificación, es decir, $\mathbf{x}_i = (x_{1i}, x_{2i}, \dots, x_{pi}) \in \mathbf{X}$. El espacio es dividido en regiones por localizaciones y etiquetas de los ejemplos de entrenamiento. Un punto en el espacio es asignado a la clase c_i si ésta es la clase más frecuente entre los n ejemplos de entrenamiento más cercano. Para ello, generalmente se usa la distancia Euclídea.

La fase de entrenamiento del algoritmo consiste en almacenar los vectores característicos y las etiquetas de las clases de los ejemplos de entrenamiento. En la fase de clasificación, la evaluación de un nuevo ejemplo del que no se conoce a qué clase pertenece es representada por un vector en el espacio característico. Se calcula la distancia entre los vectores almacenados y el nuevo vector, y se seleccionan los n ejemplos más cercanos. El nuevo ejemplo es clasificado con la clase que más se repite en los vectores seleccionados.

El principal problema del proceso de clasificación es que utiliza todos los atributos, siendo algunos irrelevantes y quitándole peso a otros que no lo son. Para solucionarlo es posible asignar un peso a las distancias de cada atributo dándole así mayor importancia a unos más que a otros [25].

Capítulo 3

Método propuesto

La mayoría de los métodos de selección de umbrales por diferencia de intensidades no dan buenos resultados cuando se trabaja con imágenes del mundo real debido a la alta presencia de ruido o a iluminación inadecuada, entre otros factores adversos. Sin embargo, el MO destaca entre ellos por una mayor robustez en la selección de umbrales para este tipo de aplicaciones y por ello, es uno de los métodos de segmentación más ampliamente utilizados. Su principal ventaja radica en que es un método automático que no necesita supervisión humana ni información previa de la imagen antes de su procesamiento.

Sin embargo, en aplicaciones donde el objetivo es determinar el número de objetos presentes en una imagen por semejanzas en las intensidades de los píxeles pertenecientes a éste y sus diferencias respecto al resto, es necesario extender el método a múltiples umbrales, lo que conlleva a que el MO pierda parte de esa autonomía que lo caracteriza puesto que el usuario debe deducir a priori cuántas clases hay presentes en la escena. Si el número de regiones predichas es insuficiente, muchos objetos quedan camuflados en el entorno, en cambio, si se definen por exceso, los objetos quedan bien definidos pero muchas otras zonas del entorno en general son segmentadas en múltiples regiones que en el mundo real representan al mismo ente.

Ante ello, en este TFM se propone un método de segmentación multiumbral para imágenes en escala de grises tomadas desde una cámara térmica que sea capaz de predecir el número de regiones mínimo necesario para definir los objetos de interés presentes en las imágenes, bajo la premisa de agrupar en una región común todas

aquellas zonas de la imagen que presenten similitudes y no aporten información nueva.

Partiendo del MO clásico o bimodal (MOB) que a partir del histograma de una imagen determina un umbral de forma automática separando las intensidades de ésta en dos partes, la idea principal del método propuesto es analizar cada uno de los dos histogramas resultantes, uno para cada parte de la imagen, y determinar mediante la definición de un factor de decisión si cada una de ellas presenta información relevante como para inferir que es necesario seguir segmentando cada uno de los histogramas de manera individual, lo que conlleva al cálculo de nuevos umbrales que definen otras sub-regiones que engloban elementos de interés. Luego, sobre cada parte de la imagen se vuelve a aplicar el MOB para dividirla en otras dos regiones y así sucesivamente. De esta forma, se tienen distintas regiones separadas por los umbrales definidos por el MOB. Esta técnica recursiva se ha empleado en algunas publicaciones del estado del arte como en [8] para la reducción de niveles de color en imágenes RGB.

A lo largo de este capítulo se explican detalladamente las distintas etapas que conforman el método de segmentación propuesto, las cuales pueden básicamente resumirse en:

1. Preprocesado de las imágenes mediante técnicas de suavizado y reducción de ruido a la vez que realce del contraste.
2. Detección de los objetos de mayor intensidad (más brillantes) en la escena.
3. Detección del resto de elementos presentes en la escena.
4. Representación de resultados.

3.1. Preprocesamiento de las imágenes. Suavizado y reducción de ruido. Realce del contraste.

Las técnicas de preprocesado tratan de mejorar y realzar las propiedades de las imágenes digitales para facilitar el posterior estudio de éstas mediante técnicas de Visión Artificial, tales como la segmentación en múltiples regiones como método para detectar objetos, tema a tratar en este TFM.

La presencia de ruido en imágenes digitales es una variable amplia y muy común que distorsiona y contamina la información dada por cada píxel, dando lugar por ejemplo a distorsión de bordes, píxeles saturados, efecto o ruido del tipo *salt and pepper*, etc. El origen del ruido es múltiple pero en la práctica existen tres tipos fundamentales: gaussiano, impulsional y multiplicativo [26].

El ruido gaussiano es el ruido más común que se origina debido al proceso de captación de la imagen al convertir los fotones en carga eléctrica [27], proceso que se explica mediante el modelo de función de densidad normal o gaussiano. El segundo tipo de ruido se le denomina *salt and pepper* y se origina tras la saturación o ausencia de carga que recibe un píxel ya sea por exceso o por defecto. La saturación se produce debido a la sensibilidad de las cámaras al infrarrojo y al encontrar sobre el escenario objetos que están calientes, por lo que la presencia de este tipo de ruido en imágenes térmicas es muy común. Por último, el ruido multiplicativo se genera por la falta de iluminación uniforme sobre la escena capturada.

En este TFM se han utilizado dos tipos de filtros de ruido: gaussiano y de la mediana. El filtro gaussiano utilizado usa una máscara de 3 x 3 que modela la función gaussiana con desviación estándar igual a 0.5 y definida por:

$$P(x, y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \quad (3.1)$$

siendo σ la desviación estándar y μ la media.

El filtro de la mediana se recomienda ampliamente para tratar con el ruido del tipo *salt and pepper*. En este caso, el valor de cada píxel de la imagen se modifica por la mediana de los valores de sus píxeles vecinos, es decir, se ordenan los valores de los píxeles vecinos de menor a mayor y se selecciona el que queda en medio.

Finalmente, para aumentar la variabilidad de los niveles de intensidad y ocupar todo el rango dinámico se ha efectuado una normalización lineal de la forma:

$$\mathbf{IMG}(x, y) = \frac{\mathbf{IMG}(x, y) - \min(\mathbf{IMG}(:))}{\max(\mathbf{IMG}(:)) - \min(\mathbf{IMG}(:))} * 255 \quad (3.2)$$

siendo $\min(\mathbf{IMG}(:))$ y $\max(\mathbf{IMG}(:))$ el valor mínimo y máximo respectivamente de todos los píxeles presentes en la imagen original (\mathbf{IMG}).

Con ello, se ha conseguido que los valores de los píxeles se encuentren entre 0 y 255, ambos incluidos.

3.2. Planteamiento del problema. Detección de los objetos más brillantes en la escena.

Una transformación cuadrática de la imagen original dará lugar a una nueva imagen más oscura donde los píxeles con mayor intensidad de la escena seguirán siéndolo, mientras que el valor del resto de píxeles disminuirá [19]. Por consiguiente, se consigue una nueva imagen donde los píxeles más brillantes ahora destacan mejor de sus alrededores. Un ejemplo de ello se muestra en la Figura 3.1 donde la imagen a) muestra la imagen original y la imagen b) la imagen resultante de la transformación cuadrática aplicada y que se rige bajo la fórmula:

$$\mathbf{IMG}_{\text{white}}(x, y) = \frac{\mathbf{IMG}(x, y)^2}{255} \quad (3.3)$$

El objetivo de dicha transformación es maximizar la varianza inter-clase entre los objetos más brillantes de la imagen, es decir, los objetos que emiten o reflejan mayor cantidad de energía infrarroja, y el resto de píxeles con niveles de grises más oscuros, objetos más fríos, para así ayudar posteriormente al MOB a obtener el umbral óptimo que separe ambas clases.

La Figura 3.2 muestra una línea marcada con distintos umbrales cuyos valores van desde 0 a 255. Se partirá de ésta para explicar las distintas etapas que prosiguen del método propuesto.

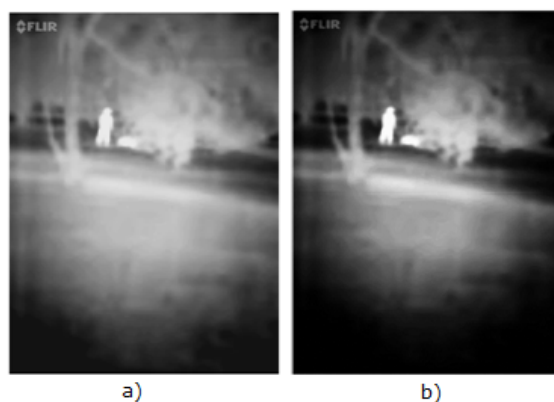


FIGURA 3.1: Comparativa entre la imagen original a) y la imagen tras sufrir la transformación cuadrática b).

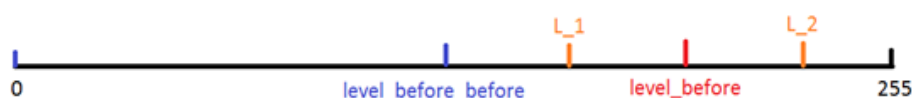


FIGURA 3.2: Línea representativa de posibles valores de umbrales comprendidos entre 0 y 255.

Una vez obtenida la imagen transformada se le aplica el MOB, el cual estimará el mejor umbral que separe los niveles de grises más claros de los más oscuros. Este umbral se corresponde con el término *level_before_before* definido en la Figura 3.2. Como el objetivo de esta primera etapa es encontrar los píxeles más brillantes, se descartan todos aquellos con niveles de grises inferiores a *level_before_before* y se obtiene una imagen reducida, $\mathbf{IMG}_{\text{reduc}}$, con sólo aquellos píxeles de la imagen transformada con valores mayores o iguales a *level_before_before*.

Una vez obtenida dicha imagen reducida, se aplica de nuevo el MOB obteniéndose un nuevo umbral comprendido entre *level_before_before* y 255 al cual se le denomina *level_before*. A continuación, se analiza qué ocurre a ambos lados del umbral *level_before* con el objetivo de determinar si existen diferentes objetos dentro del conjunto de los más brillantes. Para ello, se aplica de nuevo el MOB para el conjunto de píxeles cuyos valores están comprendidos entre *level_before_before* y *level_before* y entre *level_before* y 255, obteniéndose dos nuevos umbrales denominados *L_1* y *L_2* situados a la izquierda y a la derecha respectivamente del umbral *level_before*. Para una mayor compresión de lo explicado, la Figura 3.3 muestra una representación de dicha distribución de umbrales.



FIGURA 3.3: Línea representativa de posibles valores de umbrales para la imagen reducida, IMG_{reduc} , comprendidos entre $level_before_before$ y 255.

Una vez calculados los umbrales L_{-1} y L_{-2} se debe determinar si éstos aportan información adicional relevante, lo que podría ser indicativo de posibles objetos caracterizados por distintos niveles de grises dentro del conjunto de los más brillantes. Para ello, se hará uso de un factor de decisión que se define como:

$$\text{factor} = \left| \frac{\mathbf{\omega}(v_{up} + 1) - \mathbf{\omega}(v_{down} + 1)}{\mathbf{\omega}(256) - \mathbf{\omega}(level_before_before + 1)} \right| \quad (3.4)$$

donde $\mathbf{\omega}()$ es un vector de 256 componentes que hace referencia al histograma acumulativo de la imagen transformada, es decir, cada elemento de dicho vector representa la suma acumulativa del número de píxeles de la imagen con valores de intensidad menores a esa posición. En el caso de estar analizando si L_{-1} aporta información adicional, entonces v_{down} representa al umbral $level_before_before$ y v_{up} al umbral $level_before$. En el caso de analizar L_{-2} , entonces v_{down} representa a $level_before$ y v_{up} al nivel de gris 255. En resumidas cuentas, el factor de decisión calcula la relación entre el número de píxeles que intervienen en la determinación del umbral L_{-1} o L_{-2} , dependiendo de cuál de los dos se esté analizando, y el número de píxeles que intervino justo en el paso anterior, cuando se definió $level_before$. La Figura 3.4 muestra un esquema de la franja de niveles de grises que intervienen en la definición del numerador, línea de color naranja, y denominador, línea de color verde, del factor de decisión para el umbral L_{-1} . La Figura 3.5 muestra el mismo esquema pero para el umbral L_{-2} .



FIGURA 3.4: Franja de valores de niveles de grises que conforman el numerador y el denominador del factor de decisión para el umbral L_{-1} .

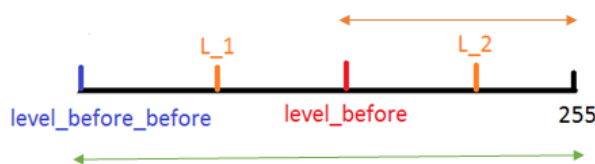


FIGURA 3.5: Franja de valores de niveles de grises que conforman el numerador y el denominador del factor de decisión para el umbral L_2 .

Finalmente, si la varianza inter-clase estimada por el MOB cuando el umbral L_1 , o L_2 en su caso, fue definido, dividida por la varianza inter-clase estimada cuando se obtuvo $level_before$ es mayor o igual que el valor del factor de decisión menos un 10% de intervalo de confianza, es decir, mayor o igual que el 90% del factor de decisión, entonces el umbral L_1 , o en su caso L_2 , es significativo y por tanto, se va a considerar para la segmentación final.

Si ninguno de los dos umbrales L_1 y L_2 cumplen la condición, entonces se analiza el umbral $level_before$ de la misma manera, estando en este caso el factor de decisión definido como:

$$\text{factor} = \left| \frac{\omega(256) - \omega(level_before_before + 1)}{\omega(256) - \omega(0 + 1)} \right| \quad (3.5)$$

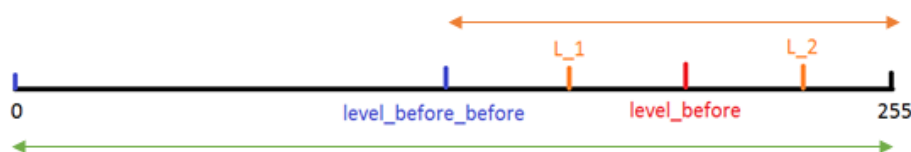


FIGURA 3.6: Franja de valores de niveles de grises que conforman el numerador y el denominador del factor de decisión para el umbral $level_before$.

En el caso de que éste tampoco cumpla la condición, entonces solamente se considerará para la segmentación final el umbral $level_before_before$.

A continuación, se procederá a analizar el caso en que L_1 , L_2 o ambos cumplan la condición impuesta para el ratio de varianzas y el factor de decisión. Antes de considerarlos como candidatos para la segmentación final de la imagen se deben realizar más comparaciones y simulaciones. Por simplicidad, se va a considerar que sólo L_1 cumple la condición. En el caso de que L_2 también la cumpla, se seguirá exactamente el mismo conjunto de pasos descritos en líneas siguientes. Una vez chequeado que L_1 cumple la condición, se debe estudiar qué pasa a la izquierda y

a la derecha de éste del mismo modo a como se ha hecho hasta ahora: se estima un nuevo umbral a la izquierda de L_1 , denominado L_{1_1} , y otro a la derecha de éste, denominado L_{1_2} , aplicando el MOB. Posteriormente, se calcula para ambos casos el factor de decisión y se compara con el ratio de varianzas (la Figura 3.7 muestra un esquema de esta nueva distribución de umbrales). A partir de este punto, cuatro situaciones son posibles:

1. Si L_{1_1} y L_{1_2} cumplen la condición, se vuelve a repetir el proceso sucesivamente.
2. Si ninguno de los dos cumple con la condición, entonces L_1 se considera para la umbralización final de la imagen.
3. Si L_{1_1} cumple con la condición pero L_{1_2} no, entonces L_1 se considera para la umbralización final de la imagen y se vuelve a repetir el mismo proceso a ambos lados del umbral L_{1_1} hasta que no se cumpla más la condición.
4. Si L_{1_2} cumple con la condición pero L_{1_1} no, entonces L_1 se considera para la umbralización final de la imagen y se vuelve a repetir el mismo proceso a ambos lados del umbral L_{1_2} hasta que no se cumpla más la condición.

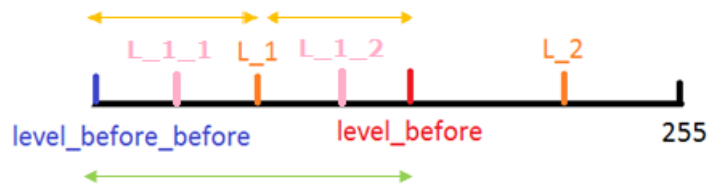


FIGURA 3.7: Franja de valores de niveles de grises que conforman el numerador y el denominador del factor de decisión para el umbral L_1 .

Para finalizar, se muestra a modo de resumen las posibles situaciones a enfrentar y qué umbrales se seleccionan para la segmentación final de la imagen, los cuales se van a considerar agrupados en un nuevo vector llamado **level_{final}**:

1. Si ni L_1 ni L_2 cumplen la condición, entonces se chequea ésta para *level_before*. Si éste cumple dicha condición, entonces *level_before* forma parte de **level_{final}**, si no, lo hace directamente *level_before_before*.
2. Si L_1 cumple la condición pero L_2 no, entonces se chequea qué ocurre a ambos lados de L_1 calculando L_{1_1} y L_{1_2} .

- a) Si L_{-1-1} y L_{-1-2} cumplen la condición, entonces se sigue repitiendo el proceso.
 - b) Si L_{-1-1} cumple la condición pero L_{-1-2} no, entonces se sigue repitiendo el proceso a ambos lados de L_{-1-1} y se guarda en **level_{final}** el valor de L_{-1} .
 - c) Si L_{-1-2} cumple la condición pero L_{-1-1} no, entonces se sigue repitiendo el proceso a ambos lados de L_{-1-2} y se guarda en **level_{final}** el valor de L_{-1} .
 - d) Si ni L_{-1-1} ni L_{-1-2} cumplen la condición, entonces se guarda en **level_{final}** el valor de L_{-1} .
 - e) Como L_{-2} no cumplió la condición, directamente se chequea la condición para *level_before*. Si la cumple, forma parte de **level_{final}**.
3. Si L_{-2} cumple la condición pero L_{-1} no, entonces se chequea que ocurre a ambos lados de L_{-2} calculando L_{-2-1} y L_{-2-2} .
- a) Si L_{-2-1} y L_{-2-2} cumplen la condición, entonces se sigue repitiendo el proceso.
 - b) Si L_{-2-1} cumple la condición pero L_{-2-2} no, entonces se sigue repitiendo el proceso a ambos lados de L_{-2-1} y se guarda en **level_{final}** el valor de L_{-2} .
 - c) Si L_{-2-2} cumple la condición pero L_{-2-1} no, entonces se sigue repitiendo el proceso a ambos lados de L_{-2-2} y se guarda en **level_{final}** el valor de L_{-2} .
 - d) Si ni L_{-2-1} ni L_{-2-2} cumplen la condición, entonces se guarda en **level_{final}** el valor de L_{-2} .
 - e) Como L_{-1} no cumplió la condición, directamente se chequea la condición para *level_before*. Si la cumple, forma parte de **level_{final}**.
4. Si L_{-1} y L_{-2} cumplen la condición, entonces se realizan por separado los pasos descritos en 2 y 3.

3.3. Detección del resto de objetos en la escena.

En la sección 3.2 se trabajó con una porción de píxeles reducidos de la imagen original con el objetivo de definir los objetos más brillantes presentes en la escena bajo estudio. En esta sección se procederá a detectar el resto de elementos presentes en la imagen, proceso algo más arduo ya que las imágenes térmicas se caracterizan por presentar amplias extensiones de píxeles oscuros, por lo que detectar nuevos objetos es un proceso más complicado ya que en la mayoría de los casos quedan camuflados en el entorno. Sin embargo, se ha adoptado una solución muy sencilla para tratar dicho inconveniente.

Una vez actualizado el vector $\mathbf{level}_{\text{final}}$ tras el análisis efectuado para detectar los objetos más brillantes de la escena, se escoge el umbral de mínimo valor de todos los presentes en dicho vector. A continuación, se umbraliza la imagen original para obtener una nueva imagen reducida, $\mathbf{IMG}_{\text{reduc}}$, formada por todos aquellos píxeles con valores de intensidad menores a dicho umbral mínimo. No se debe olvidar que dicho umbral ha sido calculado en función de la imagen modificada que ha sufrido la transformación cuadrática, por lo que antes de ser aplicado para umbralizar la imagen original debe de calcularse su equivalente mediante la fórmula:

$$\text{min_level} = \text{round}(\sqrt{\min(\mathbf{level}_{\text{final}}(:)) * 255}) \quad (3.6)$$

siendo $\text{round}()$ una función que redondea un determinado valor decimal a su entero más cercano.

El siguiente paso consiste en invertir esta nueva imagen reducida y elevarla al cuadrado de tal manera que los píxeles más oscuros ahora serán los más brillantes y viceversa. Al invertir la imagen, ahora prevalece un amplio conjunto de píxeles blancos, destacándose entre ellos algunas agrupaciones de píxeles oscuros que serán candidatos potenciales a ser objetos de interés. Para poder reutilizar parte de la metodología definida en la sección 3.2, se vuelve a invertir la imagen obtenida para que los píxeles oscuros ahora sean brillantes y viceversa. Los siguientes ecuaciones muestran las transformaciones realizadas:

$$\mathbf{IMG}_{\text{reduc}}(x, y) = \max(\mathbf{IMG}_{\text{reduc}}(:)) - \mathbf{IMG}_{\text{reduc}}(x, y) \quad (3.7)$$

$$\mathbf{IMG}_{\text{dark}}(x, y) = \frac{\mathbf{IMG}_{\text{reduc}}(x, y)^2}{255} \quad (3.8)$$

$$\mathbf{IMG}_{\text{dark}}(x, y) = \max(\mathbf{IMG}_{\text{reduc}}(:)) - \mathbf{IMG}_{\text{dark}}(x, y) \quad (3.9)$$

Una vez obtenida la nueva imagen transformada, $\mathbf{IMG}_{\text{dark}}$, se aplica sobre ella todo el proceso descrito en la sección 3.2 varias veces (ello es debido a que el número de píxeles que constituyen $\mathbf{IMG}_{\text{dark}}$ es mucho mayor que cuando se definió $\mathbf{IMG}_{\text{white}}$ en la sección anterior) de tal manera que los valores de $\mathbf{IMG}_{\text{dark}}$ se van actualizando en cada iteración y el número de píxeles que la constituyen se va reduciendo progresivamente al umbralizar por valores cada vez menores. Dicho proceso se repite hasta que se cumplan una o las dos de las siguientes condiciones: que el valor mínimo del conjunto de umbrales presentes en el vector $\mathbf{level}_{\text{final}}$ sea cero o que lo sea el número de píxeles que constituyen $\mathbf{IMG}_{\text{dark}}$.

En modo resumen, las Figuras 3.8 y 3.9 muestran un esquema representativo de las fases o etapas del método propuesto con el fin de afianzar y esclarecer los conceptos desarrollados en este capítulo.

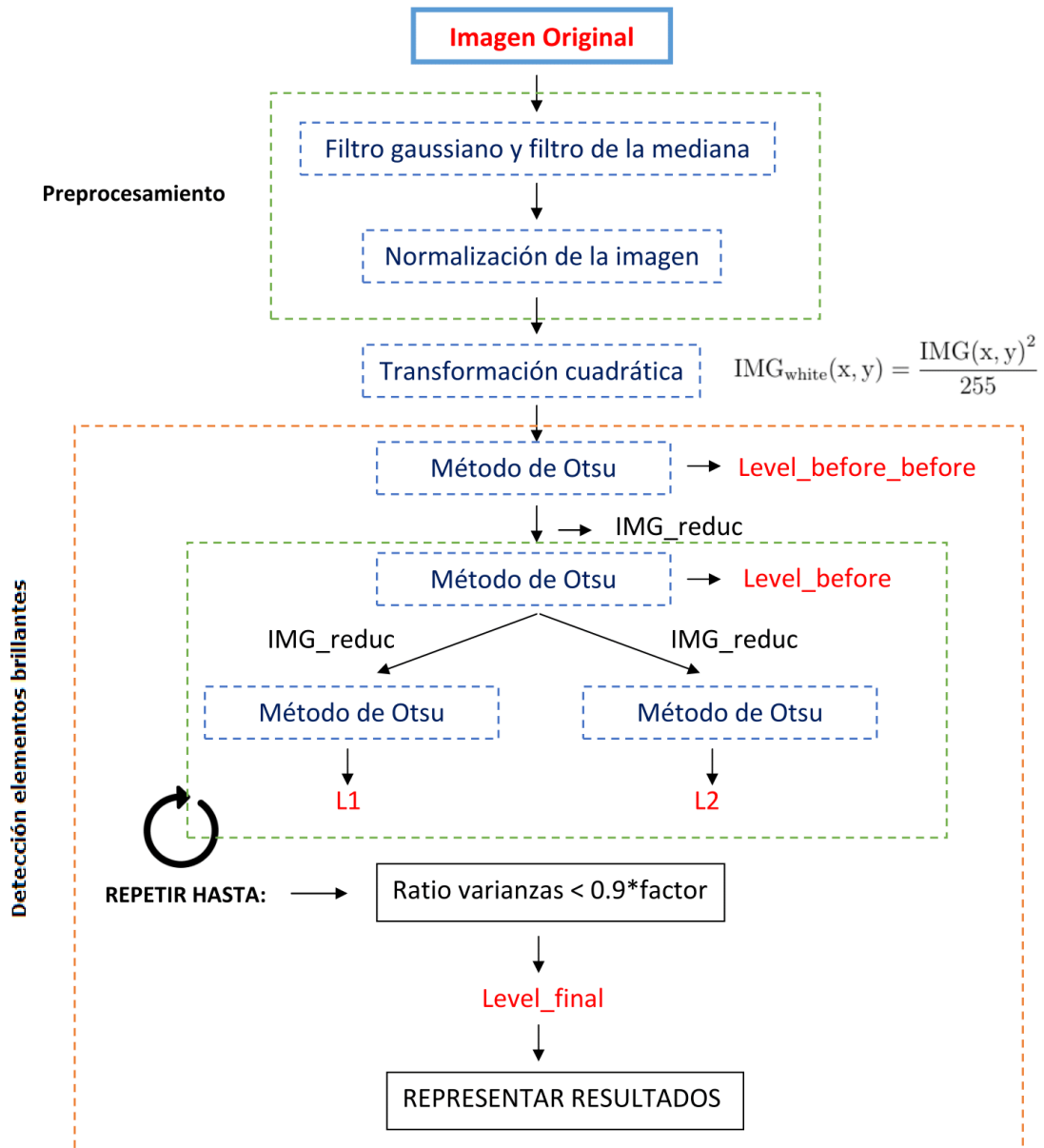


FIGURA 3.8: Digrama resumen de las etapas del método propuesto para la detección de los elementos más brillantes en la escena.

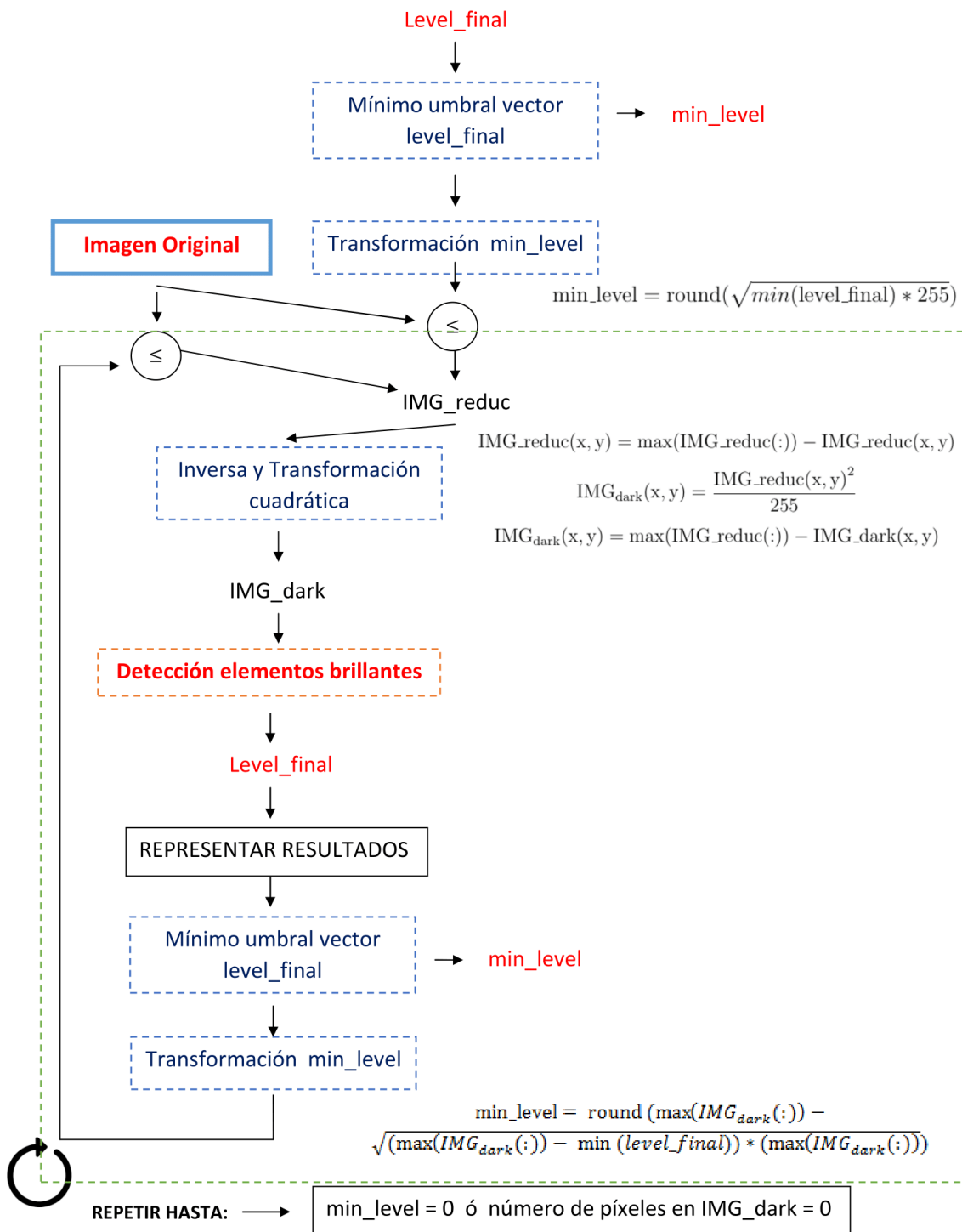


FIGURA 3.9: Digrana resumen de las etapas del método propuesto para la detección de los elementos más oscuros en la escena.

3.4. Representación de resultados.

En los esquemas representados en las Figuras 3.8 y 3.9 se puede observar cómo cada vez que se obtiene un nuevo vector $\mathbf{level}_{\text{final}}$ actualizado se pasa a una etapa/función denominada "Representar Resultados" que no se ha explicado aún.

Como dicho vector se obtiene en función de imágenes transformadas distintas a la original cuyo número de píxeles constituyentes es cada vez menor en cada iteración ($\mathbf{IMG}_{\text{reduc}}$), se consideró por simplicidad computacional generar una imagen de resultados que sea combinación de dichas imágenes reducidas.

Antes de comenzar, la matriz de resultados, con mismas dimensiones que la matriz original, se inicializa a cero. Después de realizar los pasos descritos en la sección 3.2 y de obtener el vector $\mathbf{level}_{\text{final}}$ actualizado con los umbrales que definen los objetos más brillantes de la escena, dichos umbrales son ordenados de menor a mayor. Luego se recorre la matriz de entrada a dicha etapa que ha sufrido una transformación cuadrática y se etiquetan con el valor 1 todos aquellos píxeles con valores superiores al mínimo umbral contenido en $\mathbf{level}_{\text{final}}$. Luego se repite el mismo proceso etiquetando los píxeles que sean mayores al siguiente umbral del vector $\mathbf{level}_{\text{final}}$ con un dos y así sucesivamente, incrementándose en una unidad los píxeles con valores superiores a $\mathbf{level}_{\text{final}}(1, i)$ en cada iteración, donde i va desde 1 hasta el número de umbrales presentes en $\mathbf{level}_{\text{final}}$.

Posteriormente, se procede a detectar los elementos más oscuros de la imagen por lo que la imagen de entrada a dicho proceso, $\mathbf{IMG}_{\text{dark}}$, es una imagen inversa con menor número de elementos que la original, que además ha sufrido una transformación cuadrática. Una vez obtenida dicha imagen y un nuevo vector que guarde las posiciones en la imagen original de los píxeles que la constituyen (se debe hacer hincapié que al trabajar con una imagen con menor número de elementos que la imagen original se pierde la correspondencia espacial) se procede a efectuar todos los pasos descritos en la sección 3.3. Una vez obtenido en cada iteración un nuevo vector $\mathbf{level}_{\text{final}}$ actualizado se realizan los siguientes pasos: el vector $\mathbf{level}_{\text{final}}$ es ordenado de menor a mayor, se calcula el valor máximo presente en la imagen resultado y se etiquetan a partir de este máximo todos los píxeles presentes en $\mathbf{IMG}_{\text{dark}}$ de la misma manera descrita en el párrafo anterior. Una vez acabado el proceso de etiquetado, la imagen resultado es actualizada utilizando para el ello el vector de posiciones.

A continuación, se muestra un pseudocódigo con las operaciones efectuadas en la función “Representar Resultados” donde n representa el número total de elementos de la imagen de entrada **IMG** (ésta será **IMG_{dark}** o **IMG_{white}**) que debe ser igual al número de elementos constituyentes del vector de posiciones **pos**, m el número de umbrales presentes en **level_{final}**, **IMG_{result}** la imagen resultado y N el número total de píxeles de la imagen original.

Algorithm 3.1: Función ”Representar Resultados”

Inputs: **IMG** = $[x_1, x_2, \dots, x_n]$, **level_{final}** = $[l_1, l_2, \dots, l_m]$, **pos** = $[p_1, p_2, \dots, p_n]$, **IMG_{result}** = $[r_1, r_2, \dots, r_N]$

```

1   $max = \max(\mathbf{IMG}_{\text{result}}(\cdot));$ 
2  for  $i = 1$  to  $n$  do
3     $exit = 0;$ 
4    for  $j = 1$  to  $m$  do
5      if  $\mathbf{IMG}(i) > \mathbf{level}_{\text{final}}(m)$  then
6         $\mathbf{IMG}_{\text{result}}(\mathbf{pos}(i)) = max + 1;$ 
7         $exit = 1;$ 
8      end
9    end
10 end

```

Outputs:

IMG_{result} = $[r_1, r_2, \dots, r_N]$ {Imagen resultado};

Finalmente, se obtiene una imagen resultado que muestra con un determinado color las distintas regiones en las que se ha segmentado la imagen original. Dichas regiones a su vez han sido etiquetadas desde 1 hasta r , siendo r el número total de regiones estimadas. Sin embargo, se comprueba que algunos píxeles aislados quedan sin etiquetar, es decir, con valor nulo. Esto es debido a errores en el redondeo cuando el valor mínimo del vector **level_{final}** es escalado a niveles de la imagen original para luego umbralizar ésta y obtener las sucesivas **IMG_{reduc}**. Dicho problema se subsanó de manera sencilla localizando dichos píxeles y obteniendo la moda de los 8 vecinos más próximos a él. Si la moda es distinta de cero, entonces dicho píxel queda etiquetado con su valor, en caso contrario, se analizan los 16 vecinos más próximos y así sucesivamente.

Capítulo 4

Resultados

En este capítulo se realiza un extenso análisis comparativo de los resultados arrojados por distintos algoritmos de segmentación multiumbral del estado del arte y el método propuesto en este TFM, con el objetivo de analizar la capacidad y eficacia de éste último para la segmentación en múltiples regiones de imágenes en escala de grises en función de la distribución de los niveles de intensidad predominantes en ellas. Para ello, se han utilizado un conjunto de 400 imágenes térmicas de las que se han seleccionado 18 como las más representativas para ilustrar los resultados, las cuales han sido tomadas desde una cámara termográfica para smartphones y como algoritmos de referencia se ha elegido el Método de Otsu Multiumbral (MOM) y dos clasificadores: Fuzzy C-means y Bayes. El algoritmo propuesto así como las versiones de los métodos de referencia utilizados han sido desarrollados con el software MATLAB 2015 y ejecutados en un ordenador portátil con procesador Intel[®] Core[™] i7 3610QM.

Los resultados arrojados por los algoritmos de segmentación multiumbral son básicamente una nueva representación de la imagen en escala de grises original a la que se le ha superpuesto un mapa de colores donde distintas partes o zonas de ésta, con una distribución uniforme de niveles de intensidad, se consideran como una única región a la que se le asigna un determinado color. Además, todos los píxeles constituyentes de cada región definida se etiquetan con un único valor representativo de la región a la que pertenecen. Por lo tanto, los resultados arrojados por los algoritmos de segmentación se pueden definir como mapas de colores donde cada color representa una agrupación de píxeles con una distribución similar de niveles de intensidad, lo que

conlleve que con una correcta segmentación se puedan identificar todos los posibles objetos ocultos en la escena.

A partir de ellos, en este capítulo se realizan dos exhaustivos análisis comparativos: uno cualitativo y otro cuantitativo. El análisis cualitativo parte de la observación directa de los resultados y se puede definir como una extensa descripción visual y subjetiva de cómo cada uno de los algoritmos considerados ha segmentado la imagen, haciendo hincapié sobre todo en la capacidad de cada uno para agrupar distintos objetos de interés en una o varias regiones propias distintas al resto representativo del entorno o *background*, en los bordes y el área abarcada por dichas regiones, entre otros aspectos descriptivos. Con el análisis cuantitativo se pretende afianzar un poco más las conclusiones extraídas del análisis cualitativo anterior, con el objetivo de proporcionar una idea general sobre qué método logra realizar una segmentación de los objetos de interés más precisa y correcta para la mayoría de las imágenes bajo estudio. Para ello, se hace uso de la definición de una nueva métrica matemática de evaluación cuya definición implica el conteo de píxeles y da una idea del error cometido por cada método de segmentación a la hora de delimitar las regiones que engloban a los objetos de interés. Además, para este tipo de análisis es necesario definir previamente unos mapas de referencia binarios, comúnmente conocidos como *ground truths*, que muestren los objetos de interés a definir para luego realizar una comparativa entre el número de píxeles y el área abarcada por éstos en los mapas de referencia y en los mapas resultado de los algoritmos de segmentación. Debido a la ausencia de dichos mapas de referencia para las imágenes a evaluar, se han generado algunos de manera sintética para aquellas imágenes donde los elementos a identificar están claramente delimitados.

Por lo tanto, las distintas partes o secciones en las que se estructura este capítulo son:

1. Descripción del conjunto representativo de imágenes térmicas para evaluación.
2. Análisis cualitativo de los resultados
 - a) Comparativa cualitativa con el MOM.
 - b) Comparativa cualitativa con los clasificadores Fuzzy C-means y Bayesiano.
3. Comparativa cuantitativa. Cálculo de errores en la segmentación.

4.1. Descripción del conjunto de imágenes térmicas para evaluación.

Como se ha mencionado previamente, el conjunto de datos de evaluación o *data set* utilizado está formado por 18 imágenes térmicas, las cuales se muestran en las Figuras 4.1 y 4.2.

Las imágenes a) y b) representan a un hombre paseando con su perro por un parque. Las imágenes han sido captadas de tal manera que un árbol ocupa el primer plano de la escena, quedando el hombre y su mascota en un plano más alejado. La figura c) muestra una calle con una serie de edificios al fondo y personas paseando por la acera. En la esquina derecha inferior se puede apreciar una tapa de alcantarilla metálica. En la imagen d) se puede apreciar un parque con una laguna de donde emerge un chorro de agua desde una fuente. Las imágenes e) y f) corresponden a una escena del río Manzanares canalizado a su paso por Madrid. La franja inclinada es el muro de la canalización. La figura g) fue tomada en la bahía de Santander donde los objetos más cálidos son personas a bordo de embarcaciones semejantes a canoas y el fondo es el mar. Las figuras h), i) y j) fueron tomadas en el mismo parque que las primeras. En la imagen h) un hombre pasea con su perro, en la imagen i) dos personas se encuentran caminando y en la imagen j) dos individuos juegan con un balón (objeto brillante a la derecha de la imagen). Las sombras que se aprecian se corresponden con arbustos y vegetación de los alrededores. Las imágenes k) - o) representan distintos frames de la misma escena donde varios transeúntes y vehículos se desplazan por una calle en la que se sitúa un edificio al fondo con grandes ventanales sobre los que incide la luz del sol. Finalmente, las imágenes p), q) y r) muestran el mismo edificio desde distintas perspectivas, el cual se encuentra rodeado de arboles, arbustos y vegetación en general.

Dichas imágenes fueron tomadas con una cámara térmica FlirOne [28] para dispositivos Android con una resolución espacial de 640 x 480 píxeles. Esta cámara capta la energía térmica o de infrarrojos que desprenden los objetos y que percibimos como calor. Dicha energía no resulta visible para el ojo humano ya que su longitud de onda es demasiado larga para detectarla (el rango de longitudes de onda del espectro visible abarca aproximadamente entre los 380 y 760 nm mientras que el infrarrojo cercano presenta longitudes mayores a 760 nm). Las cámaras termográficas son una herramienta excelente para mantenimiento predictivo, inspecciones de fugas en

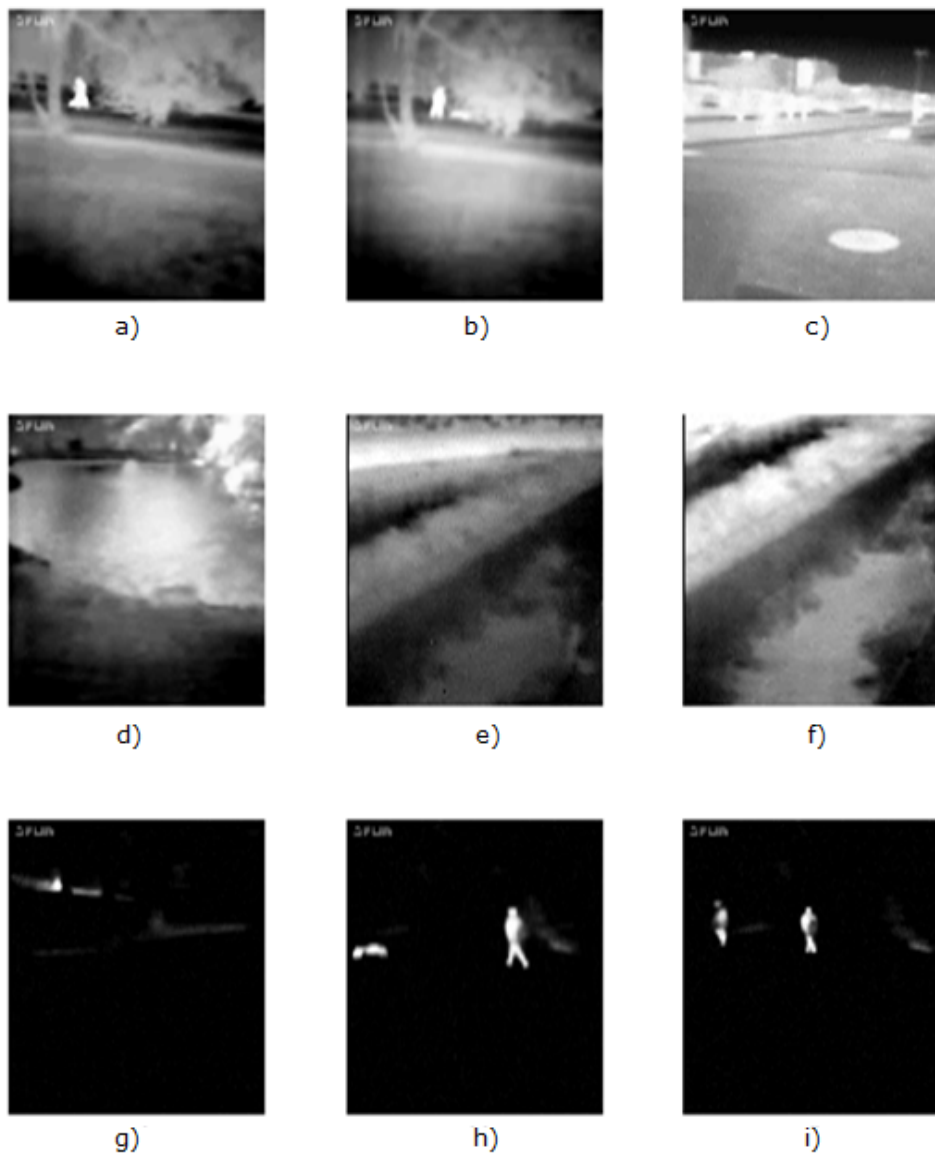


FIGURA 4.1: Data set de imágenes térmicas.

edificios, investigación y desarrollo, automatización y otras áreas como seguridad y vigilancia, aplicaciones marítimas, de la automoción, entre otras.

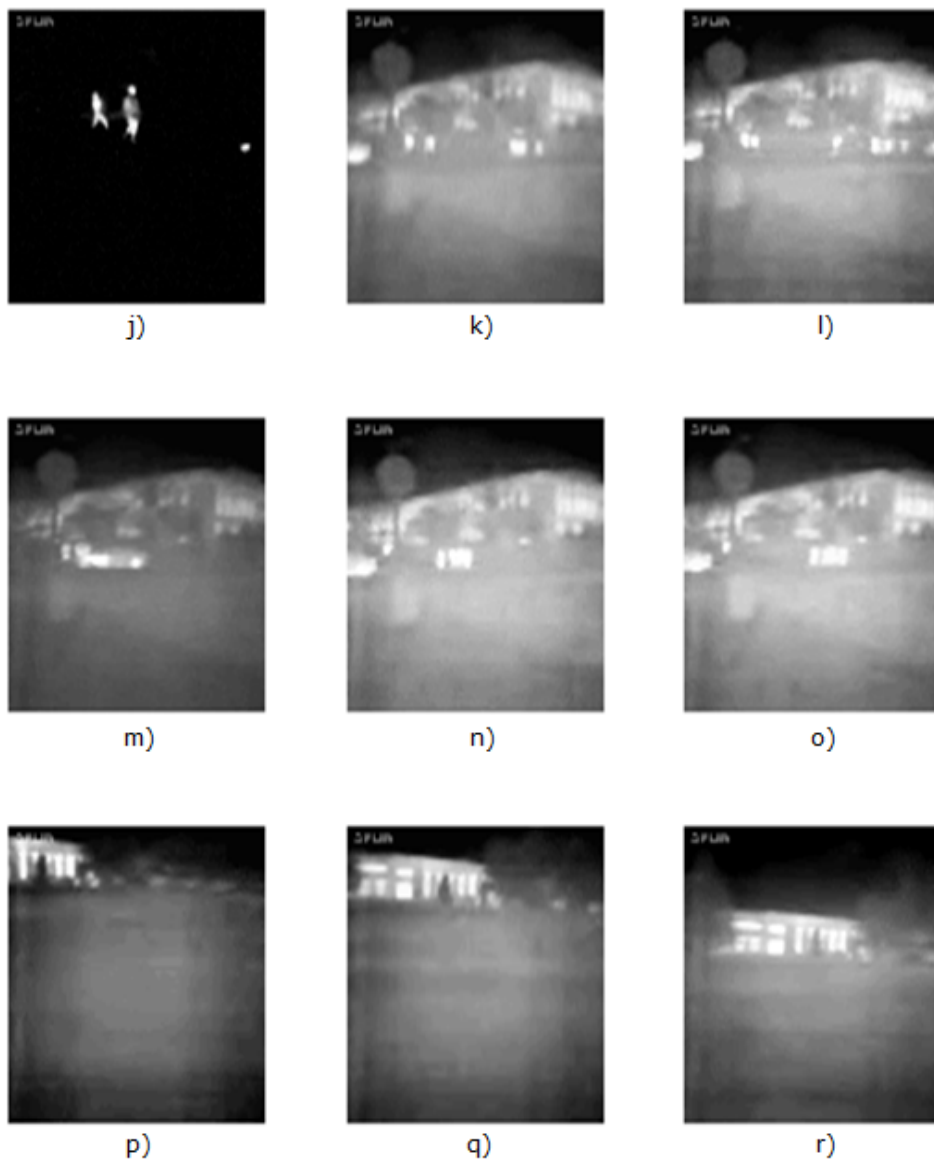


FIGURA 4.2: Data set de imágenes térmicas.

4.2. Análisis cualitativo de los resultados.

4.2.1. Comparativa cualitativa con el Método de Otsu Multiumbrales.

El MOB es un método de segmentación por umbralización que asume que la imagen bajo estudio contiene dos clases o grupos de píxeles con características homogéneas (histograma bimodal). Por tanto, calcula un único umbral óptimo que minimiza la varianza intra-clase y maximiza la varianza inter-clase. Es fácil deducir que dicho método es insuficiente para detectar diferentes objetos en las imágenes térmicas puesto que éstos se caracterizan por presentar distintos niveles de grises dependiendo de la energía térmica que emitan. En la mayoría de los casos, dichos elementos quedarán camuflados en una de las dos regiones fruto de la segmentación por un único umbral. Por ello, como se ha indicado se ha utilizado el MOM, una variante del método clásico de Otsu que permite obtener múltiples umbrales, quedando finalmente la imagen fragmentada en tantos grupos como desee el usuario.

Ante ello, en esta sección se comparan los resultados obtenidos tras la segmentación de las imágenes de test en distintas zonas, por el algoritmo propuesto en este trabajo y por el MOM para distintas configuraciones, las cuales deben ser especificadas por el usuario y constituyen un parámetro de entrada del algoritmo. El objetivo de la comparativa es resaltar la capacidad del algoritmo propuesto para deducir de manera automática el número de umbrales necesarios para definir los distintos objetos presentes en las imágenes.

A continuación, se procederá a realizar un análisis cualitativo de los resultados obtenidos para las distintas imágenes de test. En lo que sigue, para todos los casos la Figura a) representa la imagen original, la Figura b) la imagen fraccionada obtenida con el algoritmo propuesto y las Figuras c) - f) los distintos resultados obtenidos tras aplicar el MOM para distinto número de grupos o *clústeres*. Cabe destacar que el número de niveles en la leyenda al pie de cada uno de los resultados hace referencia al número de umbrales que el MOM debe estimar, por lo que, si el número de niveles es igual a 2, el número de regiones definidas serán 3 y así sucesivamente.

La Figura 4.3 muestra los resultados de la segmentación para la primera imagen de test. Se puede deducir a simple vista que existen tres grupos o zonas predominantes:

las figuras más blancas formadas por el hombre y el perro, la vegetación de los alrededores como el césped y el árbol que ocupa el plano principal, los cuales quedan definidos por tonos más grises, y por último, regiones del césped camufladas tras el árbol así como en las esquinas inferiores de la imagen caracterizadas por tonos muy oscuros, casi negros. A partir de la Figura 4.3 b) se puede observar cómo el algoritmo propuesto delimita claramente estas tres regiones, quedando el hombre y el perro marcados en color azul, las zonas más oscuras en color rojizo y la vegetación de los alrededores en color verde. La Figura 4.3 c) muestra los resultados del MOM para 2 umbrales quedándose definidas 3 regiones, la Figura 4.3 d) para 3 umbrales y 4 zonas definidas, la Figura 4.3 e) para 4 umbrales y 5 regiones y finalmente, la Figura 4.3 f) para 5 umbrales y 6 zonas. Se puede observar claramente cómo el MOM necesita definir muchas más regiones, concretamente 6, para delimitar al hombre y el perro como un grupo independiente. A diferencia que el algoritmo propuesto que intenta homogeneizar los resultados agrupando regiones con características semejantes y destacar solamente las zonas de interés, el MOM hace más hincapié en encontrar el número de niveles deseados que maximicen la varianza inter-clase sin considerar que ciertas regiones puedan llegar a representar al mismo objeto. Se puede concluir, que el algoritmo propuesto presenta unos resultados más homogéneos que se ajustan más a la realidad, sin embargo, parece que los bordes del hombre y el perro quedan más nítidos y mejor definidos en el caso del MOM con 5 niveles de umbral.

Un caso similar ocurre con la Figura 4.4 que recoge los resultados para la segunda imagen de test. Una vez más el algoritmo propuesto consigue delimitar claramente al hombre y al perro definiendo solamente 3 zonas. Sin embargo, el MOM no lo consigue aún definiendo hasta 7 regiones, ver Figura 4.4 f). Se ha comprobado que generando mayor número de regiones el MOM no es capaz de definir al hombre y al perro como una única parte constituyente. Se puede concluir que para esta imagen de test el algoritmo propuesto supera al MOM.

La comparativa de la tercera imagen del conjunto disponible es un poco más complicada y tediosa ya que los objetos a distinguir no quedan bien delimitados con respecto al entorno. Los transeúntes que pasean por la calle presentan niveles de intensidad semejantes a determinadas zonas de los edificios que quedan tras ellos, así como a la tapa de alcantarilla que se encuentra en la esquina inferior derecha de la imagen. La Figura 4.5 muestra los resultados del algoritmo propuesto así como los obtenidos por el MOM para 2, 3, 5 y 7 niveles de umbral. En la Figura 4.5 b) se puede observar cómo nuestra propuesta fragmenta la imagen en tres regiones bien

diferenciadas: el cielo que queda definido por las zonas más oscuras, el *background* formado por la carretera así como por la mayor parte de los edificios y la acera por la que pasean los transeúntes y finalmente, los objetos más brillantes incluyendo la tapa de alcantarilla metálica que al incidir el sol sobre ella hace que esté más caliente que sus alrededores. Para el mismo número de regiones, el MOM considera la mayor parte del área abarcada por los edificios así como una considerable porción de la carretera dentro de la misma región que la tapa de alcantarilla así como las personas que pasean por la calle, quedando dichos elementos camuflados (Figura 4.5 c)). Aumentando el número de niveles se consigue separar la tapa de la alcantarilla metálica del asfalto de sus alrededores. Sin embargo, es fácil deducir tras los resultados arrojados por el algoritmo propuesto que los elementos importantes de la escena a destacar podrían quedar muy bien definidos con tan sólo tres regiones, distinguiéndose mejor, por ejemplo, el metal caliente de la tapa de alcantarilla de sus alrededores, el cual queda más confuso y camuflado en las otras opciones.

Los resultados para la cuarta imagen del conjunto de datos se muestran en la Figura 4.6. Para todos los casos analizados parece que el agua del estanque sobrepasa los límites físicos de la fuente. Sin embargo, el MOM con 3 niveles de umbrales, Figura 4.6 d), consigue diferenciar mejor entre el agua más caliente que es impulsada hacia arriba, y cae luego al estanque, y el agua de éste.

Las Figuras 4.7 y 4.8 recogen los resultados para las imágenes de test e) y f) las cuales representan una escena del río Manzanares a su paso por Madrid. Para facilitar la comprensión de dichos resultados, la Figura 4.9 muestra un gráfico indicativo de los distintos elementos que conforman las escenas: árboles y vegetación tras el muro de la canalización por la que circula el río, el agua del río, vegetación presente en la rivera y sombras en general. Para los dos casos bajo estudio, el método propuesto define 3 de las 4 zonas descritas, englobando parte del área abarcada por el agua del río y la vegetación tras el muro de la canalización en una única región. En este caso, el método propuesto se ha quedado un poco corto a la hora de definir el número de zonas en las que segmentar la imagen, resultando necesario una región más que defina mejor los límites del agua del río. Ello se puede apreciar claramente observando los resultados obtenidos por el MOM que con 3 niveles de umbrales es capaz de definir mayor área de agua del río.

Los resultados de la segmentación para la imagen g) se muestran en la Figura 4.10. Para esta imagen parece que el método propuesto logra definir muchos más detalles de las personas así como de las embarcaciones sobre las que se desplazan. Los resultados del MOM para 6 niveles de umbral, es decir, 7 regiones definidas, consigue unos resultados similares a los arrojados por nuestra propuesta que tan sólo considera 5 zonas necesarias para definir todos los objetos [Figura 4.10 f) y Figura 4.10 b), respectivamente]. Ello quiere decir, que para un número elevado de niveles de umbrales, el MOM comienza a estimar umbrales para la segmentación muy próximos, separando regiones conformadas por muy pocos píxeles, que incluso podrían ser representativas del mismo objeto, sin llegar a identificar totalmente los elementos presentes en la escena.

De manera general, los resultados para las imágenes de test h), i) y j) son bastante buenos para todas las opciones (Figuras 4.11, 4.12 y 4.13, respectivamente). Tanto las personas que pasean, como el perro en la Figura 4.12 y la pelota en la Figura 4.13 quedan muy bien definidos, incluso, las distintas partes del cuerpo del perro y las personas que emiten más o menos calor quedan marcadas. Sin embargo, a partir de los resultados obtenidos para las imágenes h) y j) se aprecia que el método propuesto es un poco más sensible al ruido. En contraposición, consigue definir mejor algunas sombras que corresponden con vegetación de los alrededores.

En las imágenes de test desde la k) hasta la o) los objetos a detectar son las personas y los vehículos que circulan por la zona. Para facilitar la comprensión de la escena, la Figura 4.40 muestra una serie de *ground truths* de dichas imágenes donde los elementos en blanco representan a los transeúntes y los vehículos. Los resultados de la segmentación para los dos métodos bajo estudio se recogen desde la Figura 4.14 hasta la 4.18. Para las tres primeras imágenes, el número de umbrales para el MOM fueron 2, 3, 4 y 5, mientras que para las dos imágenes restantes fueron 3, 5, 6 y 7. Analizando los resultados, el MOM consigue definir mayor número de detalles a medida que el número de umbrales aumenta. De forma general, el método propuesto otorga mayor importancia a detectar a las personas y vehículos que circulan por la zona, considerando en un plano secundario al resto de elementos de la escena, englobándolos en una única región marcada en color verde. En algunos casos, como en la Figura 4.15, 4.16 y 4.18, también detecta otros elementos como los ventanales del edificio situados en la parte superior derecha de las imágenes. Sin embargo, los resultados para el MOM con 5 y 7 niveles de umbrales definen mejor los objetivos deseados para todos los casos. Un claro ejemplo lo representa la Figura 4.14 pues

el método propuesto no ha sido capaz de destacar a la persona situada más a la derecha de la imagen.

Finalmente, los resultados de las imágenes p), q) y r), se muestran en las Figuras 4.19, 4.20 y 4.21, respectivamente. Para todos los casos, el algoritmo propuesto muestra unos resultados bastante homogéneos diferenciando cuatro regiones bien definidas: el cielo y zonas muy oscuras; césped, árboles y vegetación en general; el cerramiento y muros del edificio construidos en su mayoría de hormigón y por último, las amplias cristaleras. En algunos casos, Figuras 4.19 b) y 4.21 b), partes del cerramiento del edificio quedan camufladas entre la vegetación y son etiquetadas como tal. A diferencia con el método propuesto, para el mismo número de regiones definidas, el MOM considera todas las partes del edificio como un todo y divide las zonas correspondientes a vegetación en dos regiones. Claramente, a medida que el número de umbrales aumenta, muchos más elementos y regiones son definidos. Sin embargo, se generan nuevas zonas redundantes, sobre todo de vegetación, que en la realidad representan lo mismo y que deberían fusionarse conformando una única región. Incluso para la imagen q), el MOM no es capaz de hacer distinción entre los ventanales del edificio y el resto de elementos de éste.

Por tanto, se puede concluir que a medida que el número de niveles de umbral aumenta, más elementos son identificados pero muchas de las regiones definidas no muestran información nueva, generándose mapas de segmentación con múltiples zonas que representan lo mismo. Sin embargo, la tendencia del método propuesto es definir el mayor número de objetos de interés cuyos niveles de grises difieran significativamente del resto de elementos que más predominan en la escena bajo la premisa de generar el menor número de regiones posibles. Por tanto, los mapas de segmentación resultado son más fáciles de procesar y analizar puesto que dividen la imagen en un número de zonas suficiente como para detallar los elementos de mayor interés. Por el contrario, una de las desventajas de homogeneizar regiones caracterizadas por niveles de grises semejantes es que, en ocasiones, algunos objetos quedan camuflados en ellas.

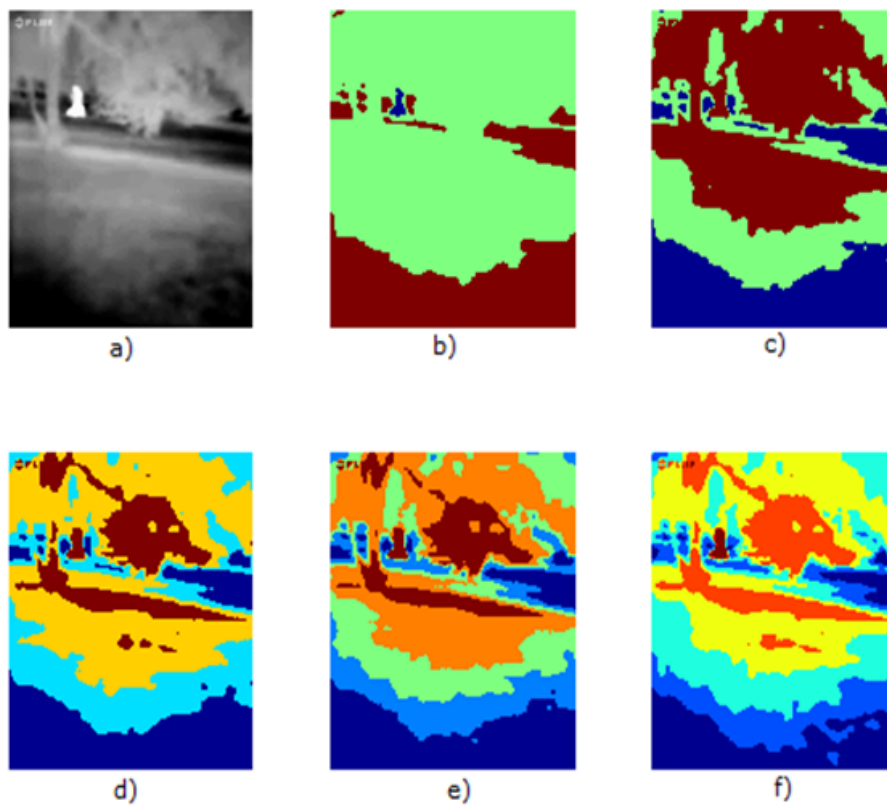


FIGURA 4.3: Comparativa de los resultados tras la segmentación de la imagen a) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 4 niveles, f) MOM para 5 niveles.

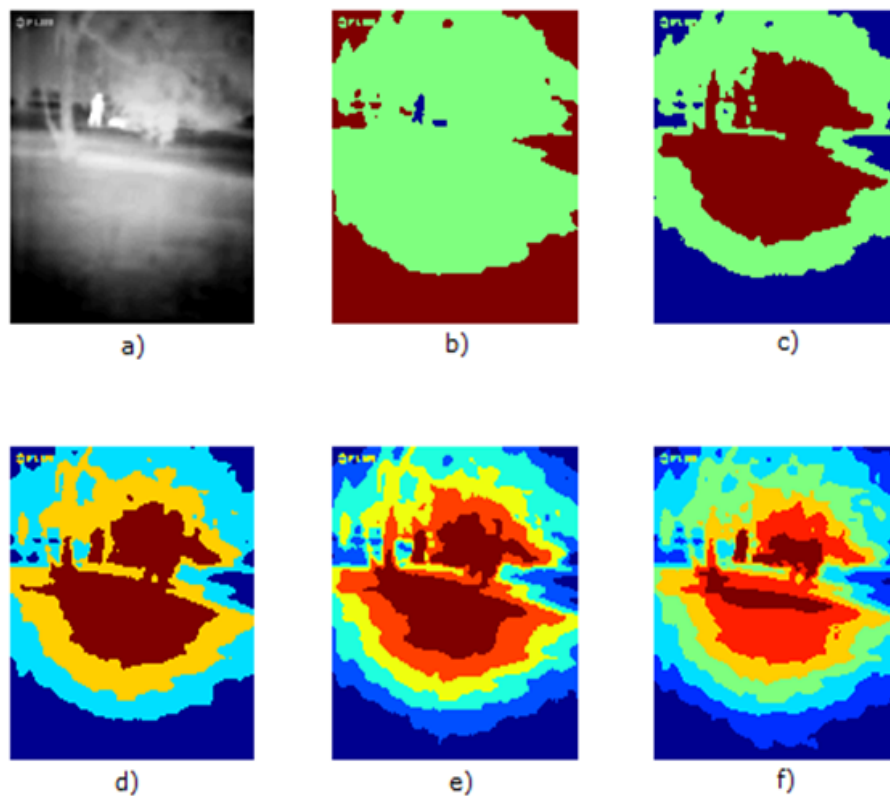


FIGURA 4.4: Comparativa de los resultados tras la segmentación de la imagen b) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 5 niveles, f) MOM para 6 niveles.

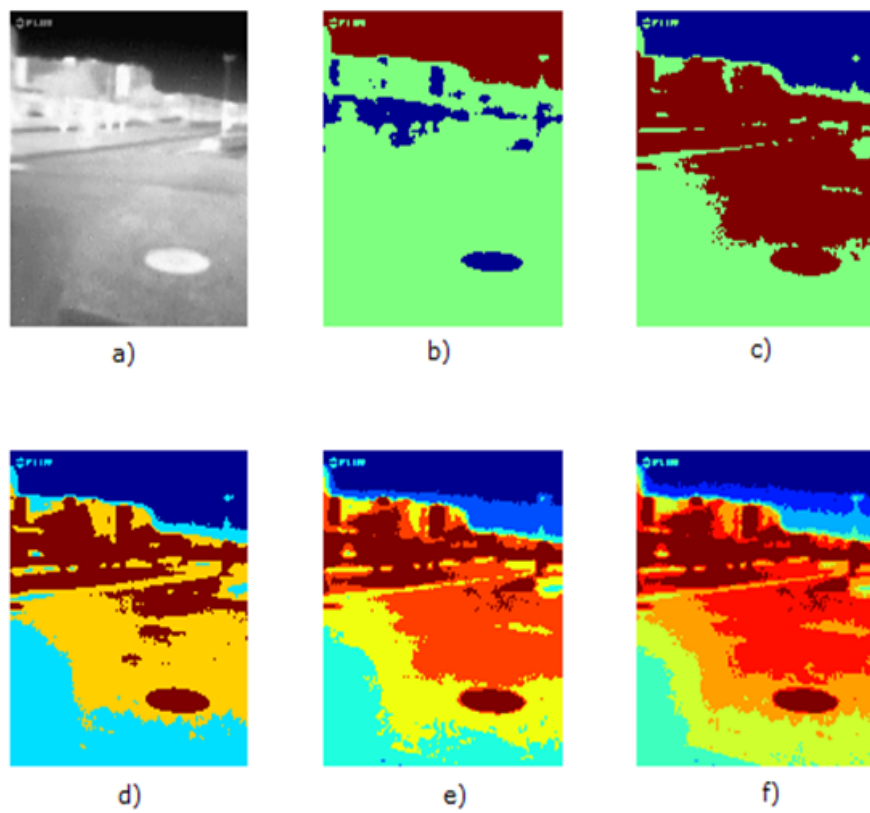


FIGURA 4.5: Comparativa de los resultados tras la segmentación de la imagen c) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 5 niveles, f) MOM para 7 niveles.

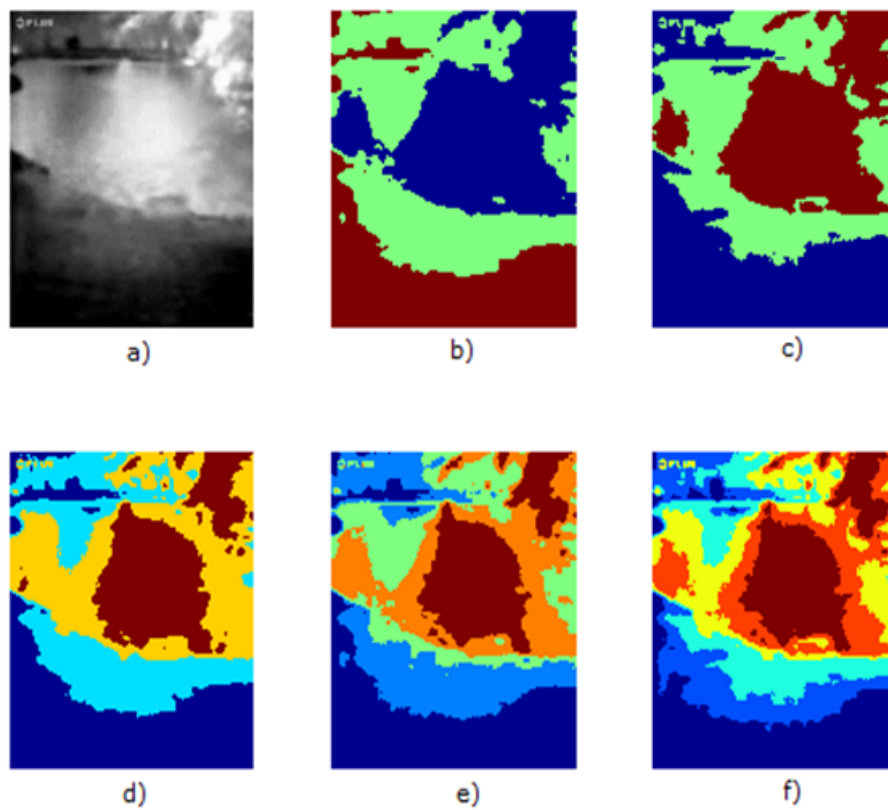


FIGURA 4.6: Comparativa de los resultados tras la segmentación de la imagen d) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 4 niveles, f) MOM para 5 niveles.

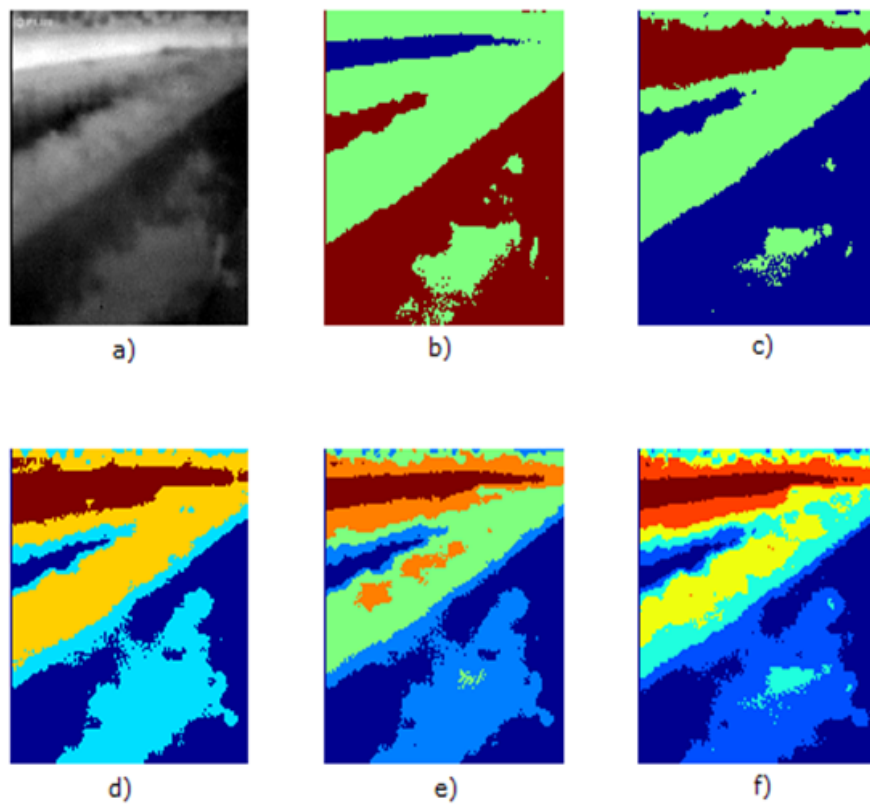


FIGURA 4.7: Comparativa de los resultados tras la segmentación de la imagen e) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 4 niveles, f) MOM para 5 niveles.

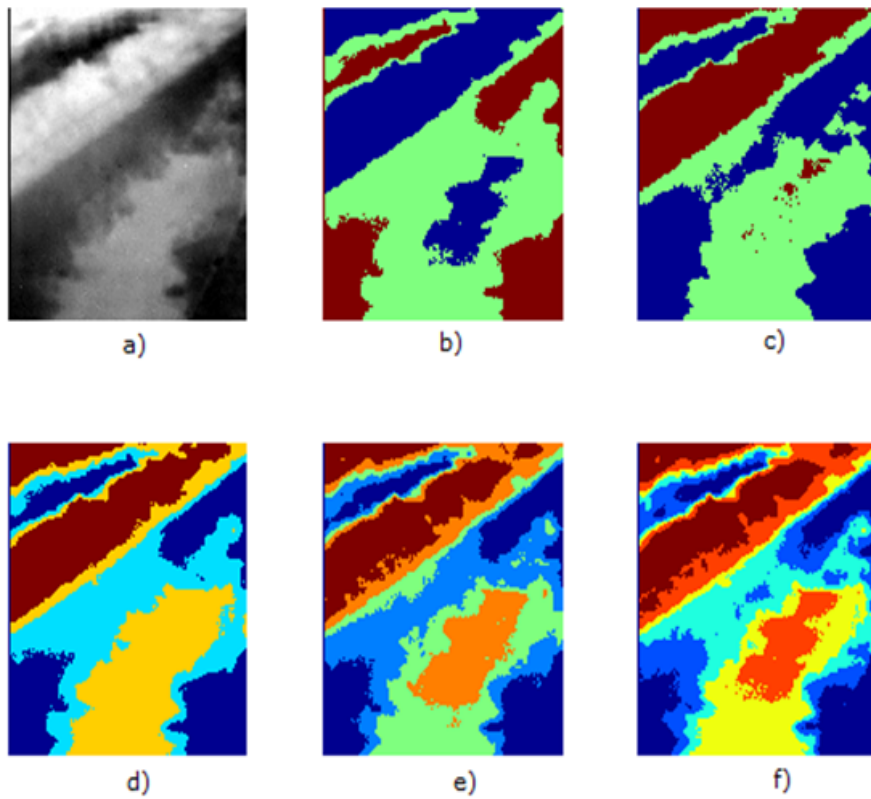


FIGURA 4.8: Comparativa de los resultados tras la segmentación de la imagen f) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 4 niveles, f) MOM para 5 niveles.

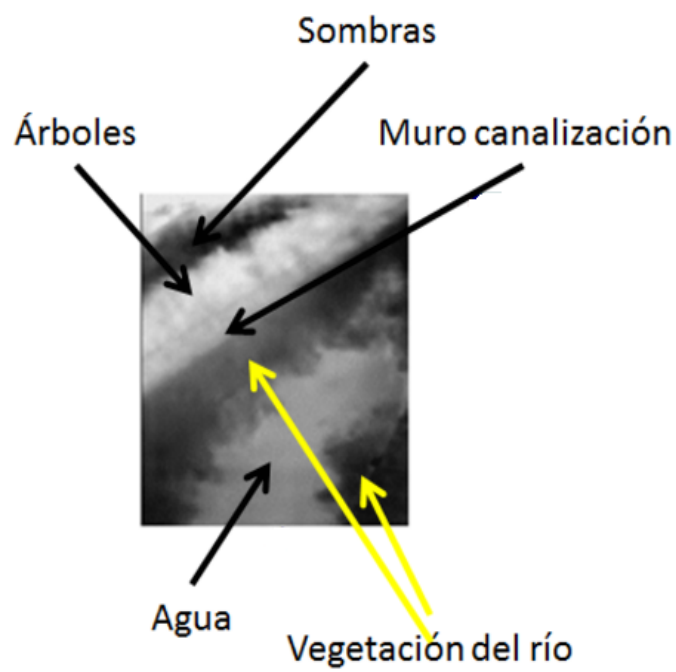


FIGURA 4.9: Croquis para la identificación y situación de los distintos elementos presentes en las imágenes e) y f).

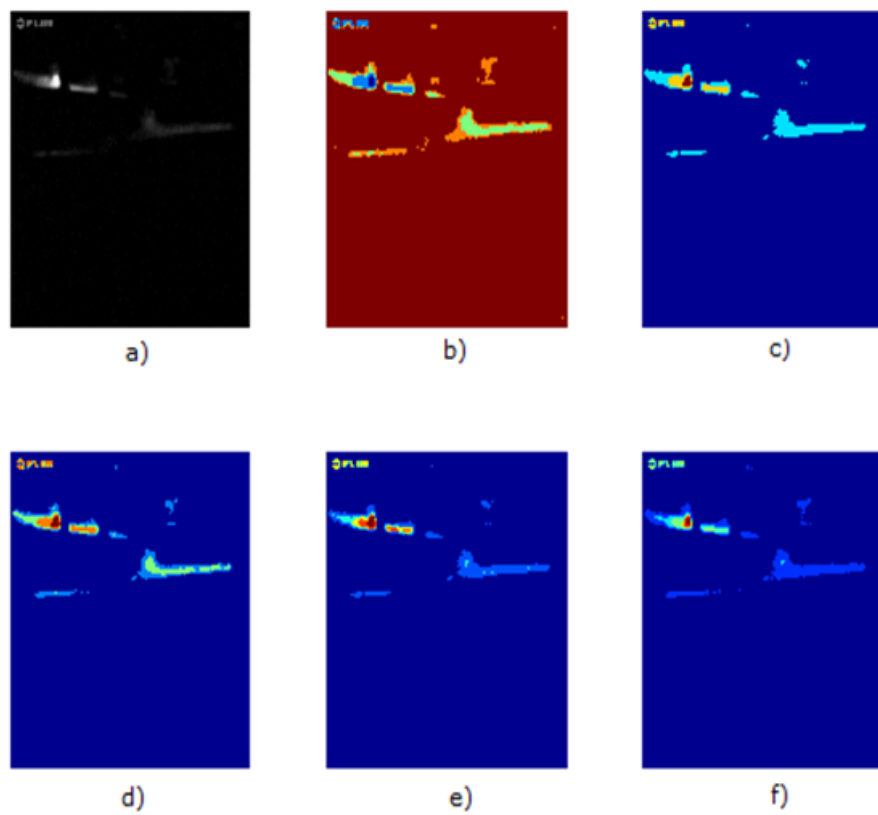


FIGURA 4.10: Comparativa de los resultados tras la segmentación de la imagen g) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 4 niveles, e) MOM para 5 niveles, f) MOM para 6 niveles.

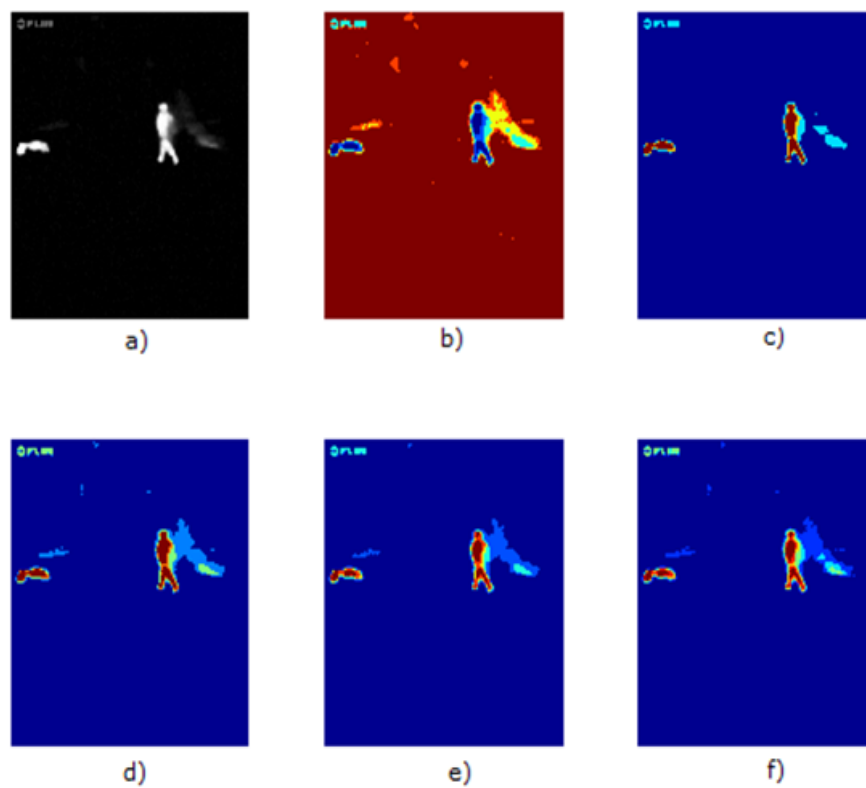


FIGURA 4.11: Comparativa de los resultados tras la segmentación de la imagen h) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 4 niveles, e) MOM para 5 niveles, f) MOM para 6 niveles.

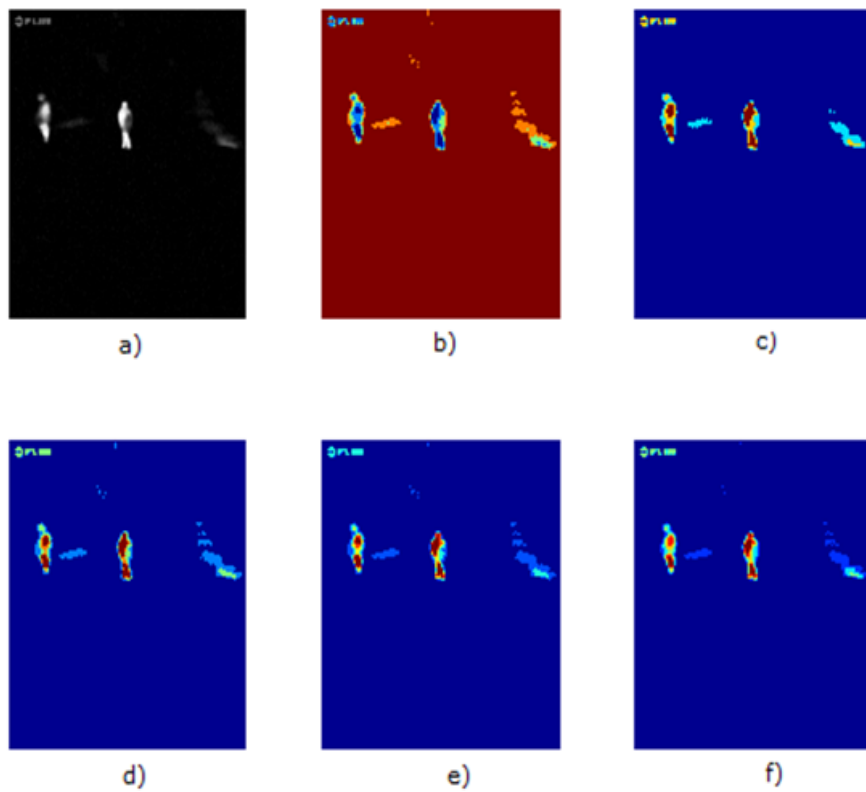


FIGURA 4.12: Comparativa de los resultados tras la segmentación de la imagen i) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 4 niveles, e) MOM para 5 niveles, f) MOM para 6 niveles.

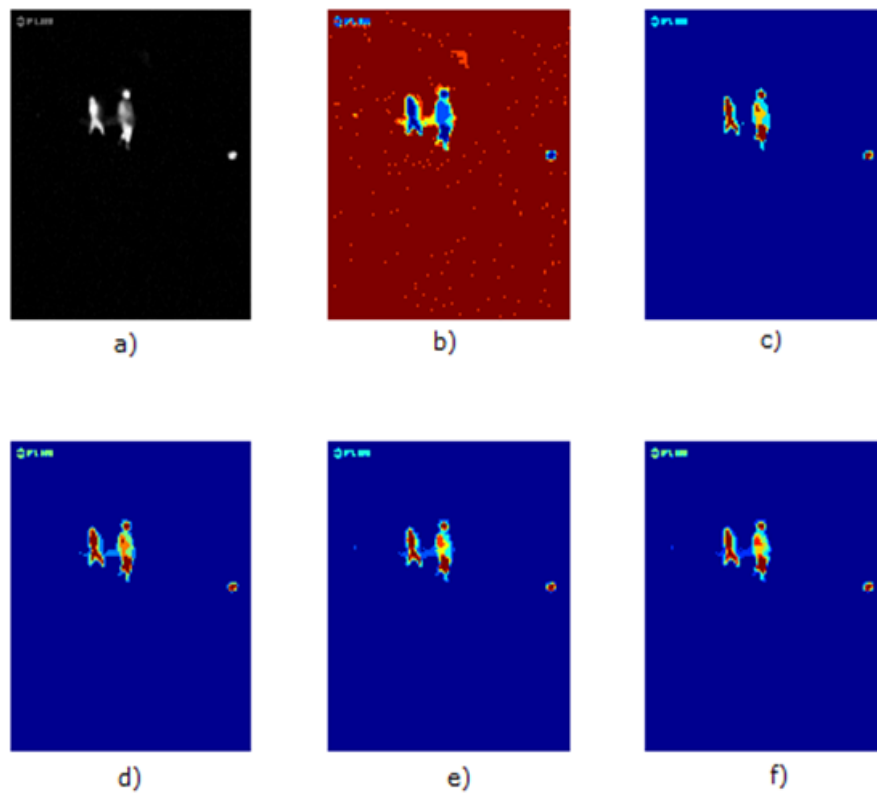


FIGURA 4.13: Comparativa de los resultados tras la segmentación de la imagen j) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 4 niveles, e) MOM para 5 niveles, f) MOM para 6 niveles.

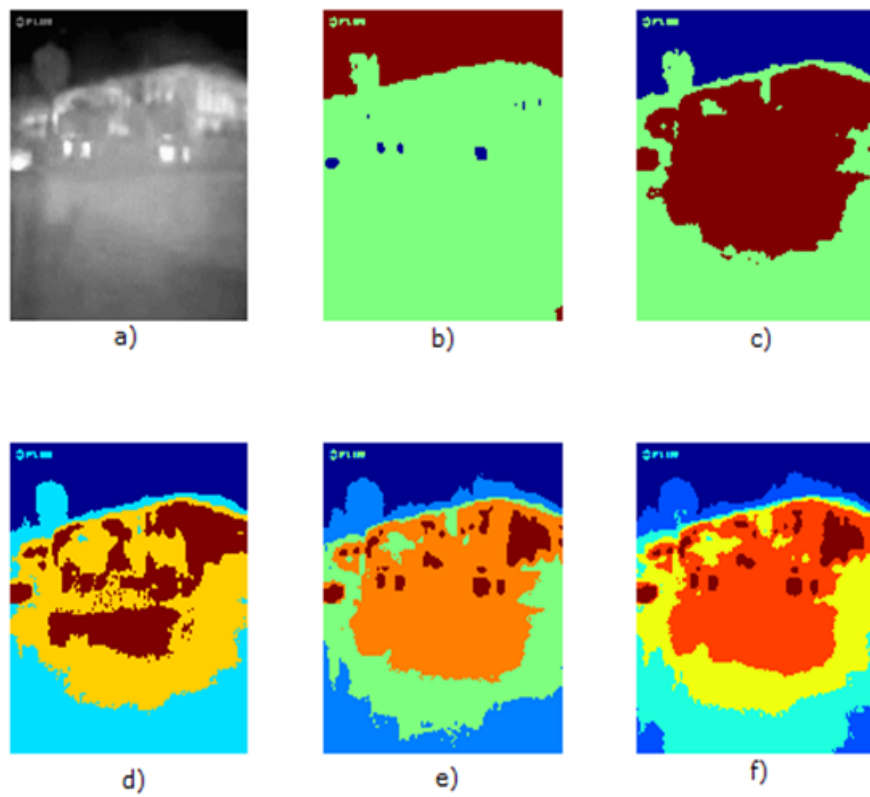


FIGURA 4.14: Comparativa de los resultados tras la segmentación de la imagen k) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 4 niveles, f) MOM para 5 niveles.

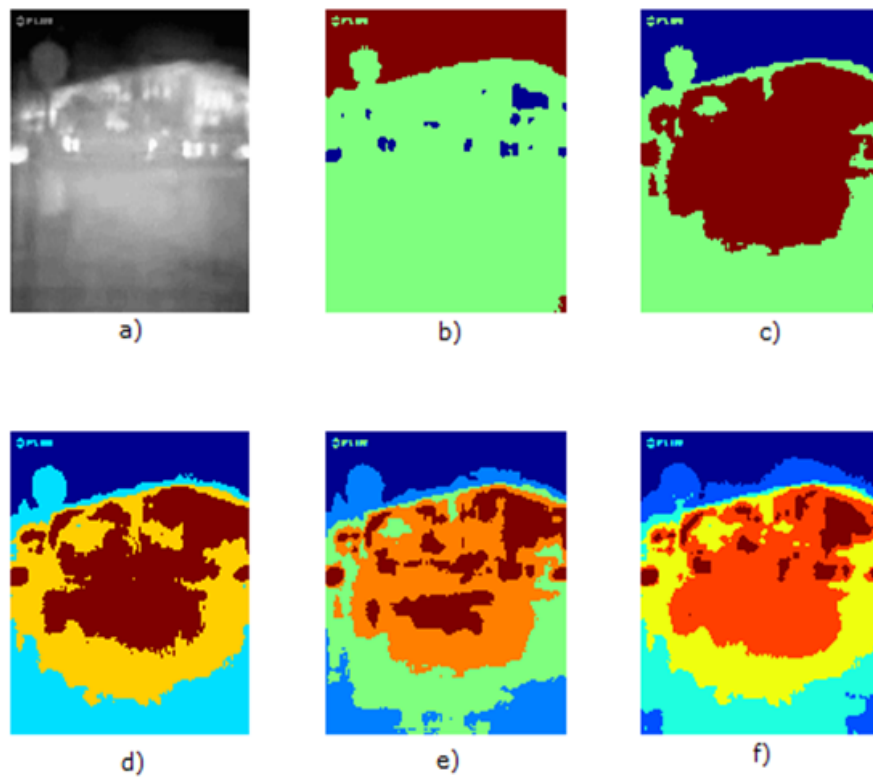


FIGURA 4.15: Comparativa de los resultados tras la segmentación de la imagen l) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 4 niveles, f) MOM para 5 niveles.

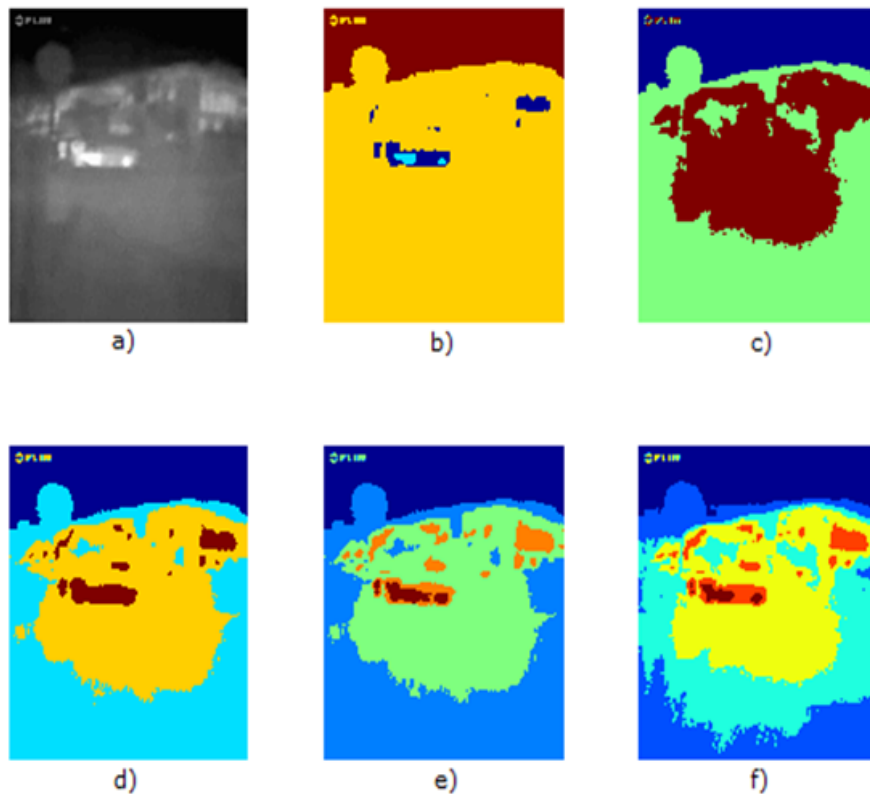


FIGURA 4.16: Comparativa de los resultados tras la segmentación de la imagen m) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 2 niveles, d) MOM para 3 niveles, e) MOM para 4 niveles, f) MOM para 5 niveles.

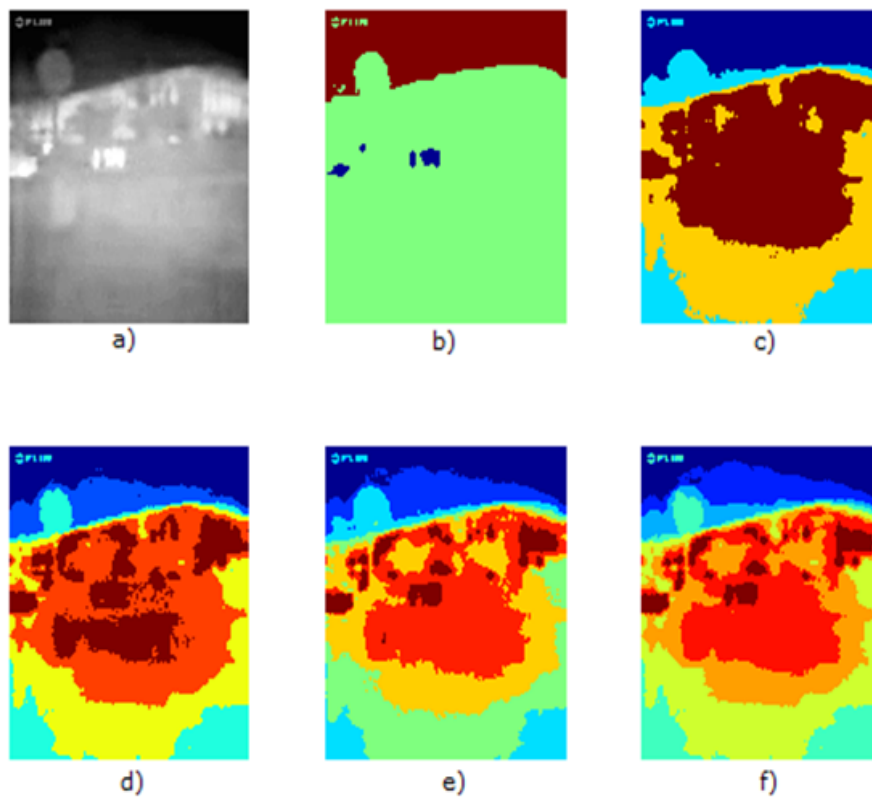


FIGURA 4.17: Comparativa de los resultados tras la segmentación de la imagen n) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 5 niveles, e) MOM para 6 niveles, f) MOM para 7 niveles.

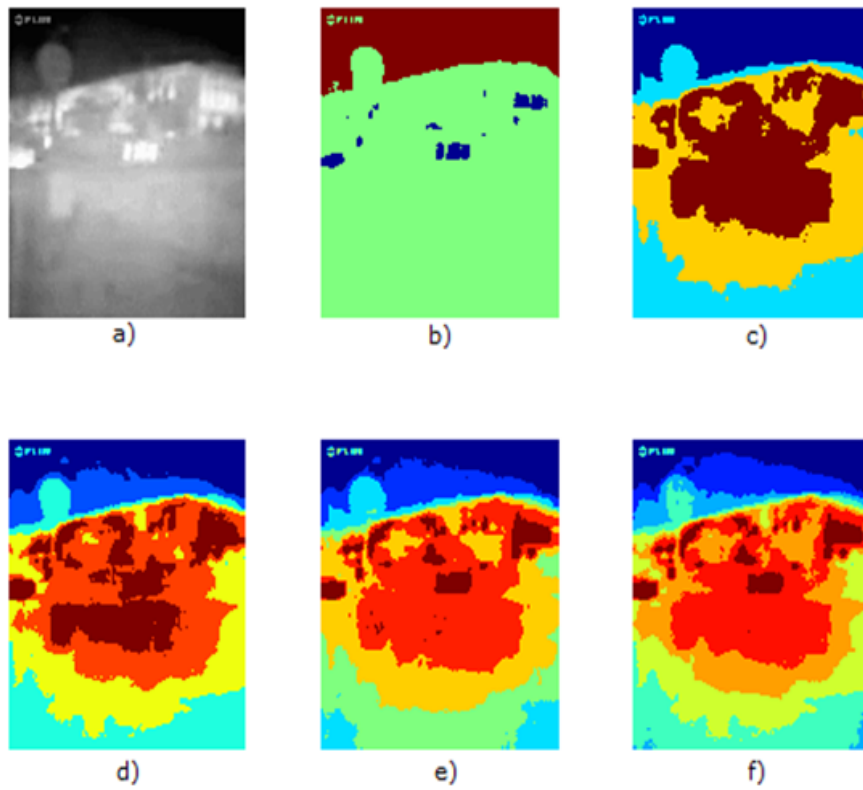


FIGURA 4.18: Comparativa de los resultados tras la segmentación de la imagen o) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 5 niveles, e) MOM para 6 niveles, f) MOM para 7 niveles.

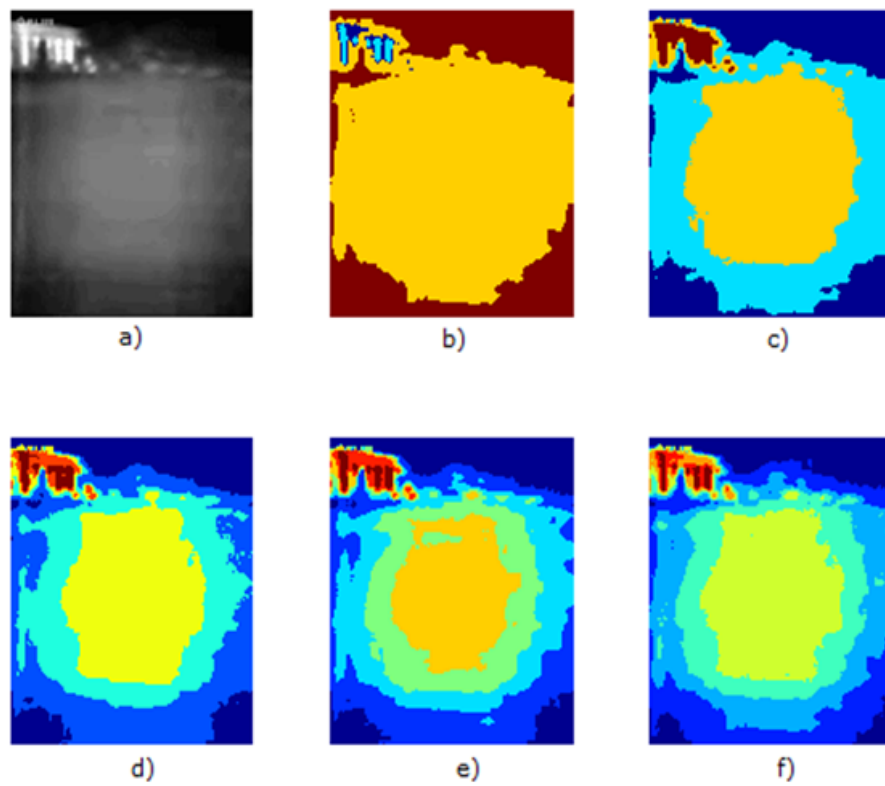


FIGURA 4.19: Comparativa de los resultados tras la segmentación de la imagen p) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 5 niveles, e) MOM para 6 niveles, f) MOM para 7 niveles.

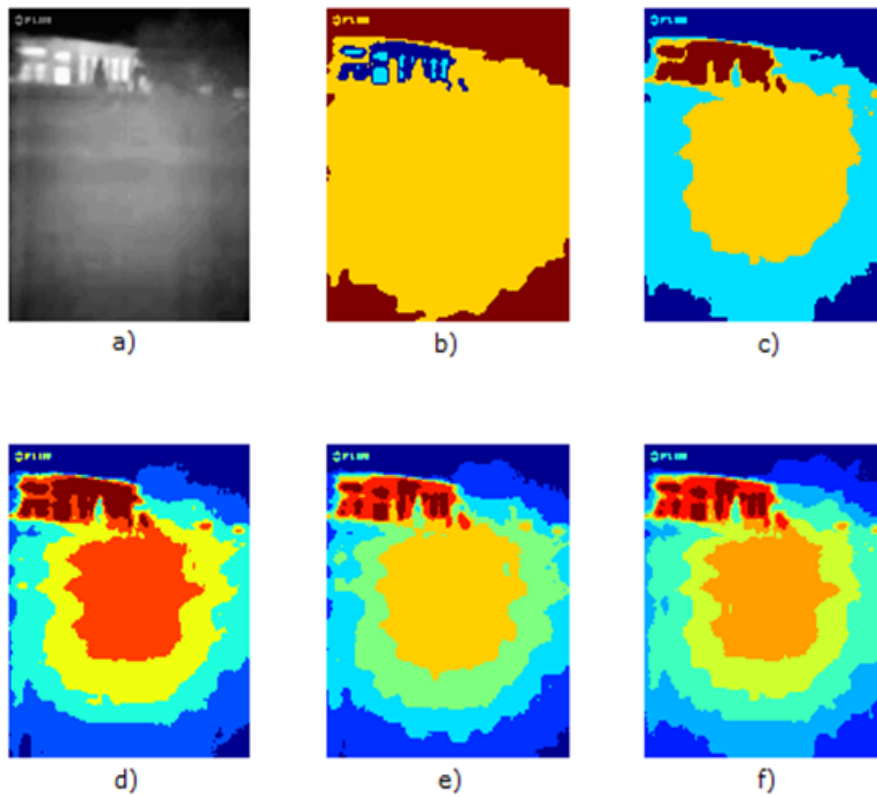


FIGURA 4.20: Comparativa de los resultados tras la segmentación de la imagen q) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 5 niveles, e) MOM para 6 niveles, f) MOM para 7 niveles.

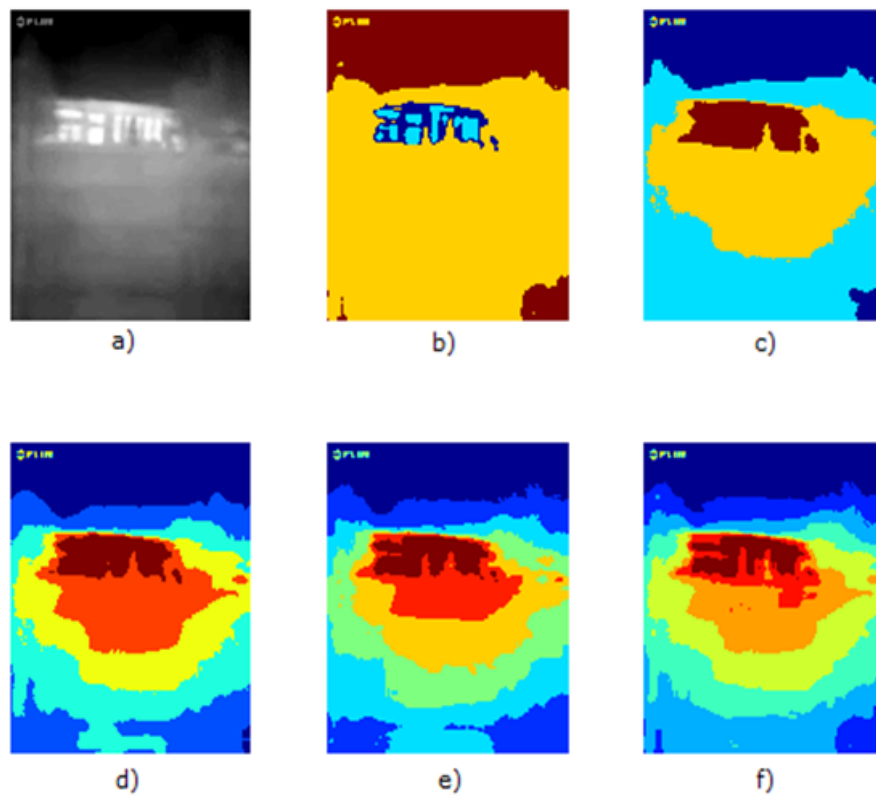


FIGURA 4.21: Comparativa de los resultados tras la segmentación de la imagen r) obtenidos por el algoritmo propuesto y por el MOM. a) Imagen original, b) Algoritmo propuesto, c) MOM para 3 niveles, d) MOM para 5 niveles, e) MOM para 6 niveles, f) MOM para 7 niveles.

4.2.2. Comparativa cualitativa con los clasificadores Fuzzy C-means y Bayesiano.

En esta sección se procede a realizar un análisis cualitativo similar al efectuado en la sección anterior donde esta vez los resultados se comparan con los dos conocidos clasificadores supervisados del estado del arte: Fuzzy y Bayes. Además, también se hace referencia a los resultados obtenidos con el MOM con el objetivo de realizar una comparativa más general.

Antes de comenzar con el análisis de los resultados, se debe destacar que para cada imagen se emplearon 2500 píxeles extraídos de manera aleatoria de 5 imágenes diferentes del data set para la etapa de entrenamiento. Finalmente, todos los píxeles constituyentes de la imagen bajo estudio fueron clasificados según los modelos definidos anteriormente durante el entrenamiento.

En términos generales, los resultados arrojados por los dos clasificadores son muy similares para todos los casos. En las Figuras 4.22 y 4.23 se puede apreciar cómo para las imágenes de test a) y b) ningún clasificador es capaz de diferenciar a la persona que pasea junto a su mascota por el parque de ciertas zonas del césped y del árbol situados en el plano principal. A semejanza con los resultados obtenidos por el algoritmo propuesto, ambos clasificadores también segmentan la imagen en tres regiones. Recordando los resultados obtenidos con el MOM se puede ver cómo entre todas las opciones, nuestra propuesta obtiene para estas imágenes los mejores resultados definiendo el mínimo número de regiones posibles, sin olvidar, que para ninguna combinación de niveles de umbral, el MOM logra separar al individuo y a su perro del *background* que lo rodea en el caso de la segunda imagen.

Para la tercera imagen de test, Figura 4.24, los clasificadores también definen el mismo número de regiones que nuestra propuesta, aunque la distribución de éstas difiere bastante, siendo similar a la obtenida con el MOM para 2 niveles de umbral. En este caso, ambos clasificadores clasifican los edificios presentes en la escena así como parte de la calzada y la tapa metálica de la alcantarilla dentro de la misma región, quedando esta última camuflada en el entorno que la rodea. En contraposición, el método propuesto consigue destacar la tapa de alcantarilla del resto de elementos, definiendo una vez más el mínimo número de regiones.

Algo similar ocurre con las imágenes de test d), e) y f), Figuras 4.25, 4.26 y 4.27 respectivamente, ambos clasificadores han definido el mismo número de regiones que nuestra propuesta, sin embargo, en contrapartida han conseguido separar espacialmente la zona del estanque sobre la que cae el agua que emerge de la fuente de las ramas del árbol situadas en la parte derecha superior de la imagen d), han establecido una mayor área abarcada por el agua que lleva el río Manzanares en la imagen f), consiguiendo incluso mejores resultados que el MOM para mayor número de zonas definidas, y por último, para la imagen e) solamente el clasificador Bayesiano define algo más de agua que lleva el río, siendo los resultados del Método Fuzzy semejantes a los obtenidos por el algoritmo propuesto.

Para la imagen de test g), los resultados de la segmentación se recogen en la Figura 4.28. A simple vista la segmentación realizada por ambos clasificadores es muy similar también en este caso, aunque parece que el método de Bayes consigue delimitar mejor uno de los extremos de la canoa situada en el plano principal. No obstante, los resultados arrojados por los clasificadores no logran el grado de precisión a la hora de delimitar los distintos elementos presentes en la escena como consigue el algoritmo propuesto definiendo tan solo una región más..

Para la imagen de test h), Figura 4.29, ambos clasificadores consiguen delimitar muy bien a la persona y a su mascota, además de definir las zonas de la fisonomía más o menos cálidas de éstos. Sin embargo, parece que los resultados proporcionados por el clasificador Bayesiano se asemejan más a los del método propuesto puesto que define mayor número de elementos del *background* que el clasificador Fuzzy. A pesar de ello, nuestra propuesta consigue extraer mayor información y detalle de los alrededores.

Algo muy similar ocurre con la imagen de test i), Figura 4.30. Sin embargo, a pesar de que ambos clasificadores definen muy bien a las dos personas que se encuentran en la escena, proporcionado incluso mayor detalle sobre la fisonomía de los individuos el método Fuzzy, en lo referente a la vegetación de los alrededores tanto el comportamiento del método propuesto como del MOM sobresalen notablemente.

Para la imagen de test j), Figura 4.31, ocurre el mismo fenómeno analizado durante la comparativa con el MOM. En este caso, los resultados obtenidos por los clasificadores no se ven afectados por el ruido presente en la imagen al igual que con el MOM, sin embargo, no ocurre lo mismo con el método propuesto.

Para las escenas representadas por las imágenes k) - o), el comportamiento de los clasificadores Bayesiano y Fuzzy es prácticamente idéntico, ambos segmentan las imágenes en 3 zonas de manera semejante al MOM para el mismo número de regiones. Sin embargo, con la segmentación efectuada por dichos algoritmos ninguna persona, vehículo o ventanal es identificado, quedando cualquier elemento camuflado en el *background*. No obstante, la imagen m) representa una excepción. Ello es debido a que la veracidad de los resultados de cualquier método de clasificación supervisado se fundamenta en el conjunto de datos utilizado durante el entrenamiento. Debido a que dichos datos se escogieron al azar, otra combinación podría haber arrojado resultados diferentes para el resto de imágenes, lo que demuestra que los métodos de clasificación muestran un comportamiento poco robusto a cambios en el conjunto de muestras de entrenamiento. En contraposición, el algoritmo propuesto define el mismo número de regiones pero ofrece una única solución que consigue reestructurar debidamente el entorno e identificar aquellos objetos de interés presentes en la escena englobándolos en una única región. Los resultados se recogen en las Figuras 4.32 - 4.36.

Finalmente, para las escenas mostradas en las imágenes p) - r), una vez más el comportamiento de ambos clasificadores es muy similar, ambos definiendo 3 regiones para cada imagen, una menos que nuestra propuesta. La segmentación efectuada por los clasificadores es bastante aceptable y lógica; no obstante, al definir el método propuesto una región más permite diferenciar los ventanales del edificio del resto de cerramiento y vegetación de los alrededores. En contrapartida, ambos clasificadores logran delimitar una serie de setos y arbustos situados a la derecha del edificio mientras que nuestra propuesta los considera dentro del *background* general. Además, el modo en que cada método segmenta al fondo es totalmente diferente. Mientras que el método propuesto lo divide en dos regiones en función de las zonas más oscuras de la imagen como el cielo y toda la vegetación presente en la escena ya sean árboles, arbustos y la totalidad de la superficie de césped que se extiende en frente del edificio, los clasificadores dividen a éste en función de las zonas de vegetación más cálidas o mejor captadas por el campo de visión o FOV (*field of view*) de la cámara. Los resultados se muestra en las Figuras 4.37 - 4.39.

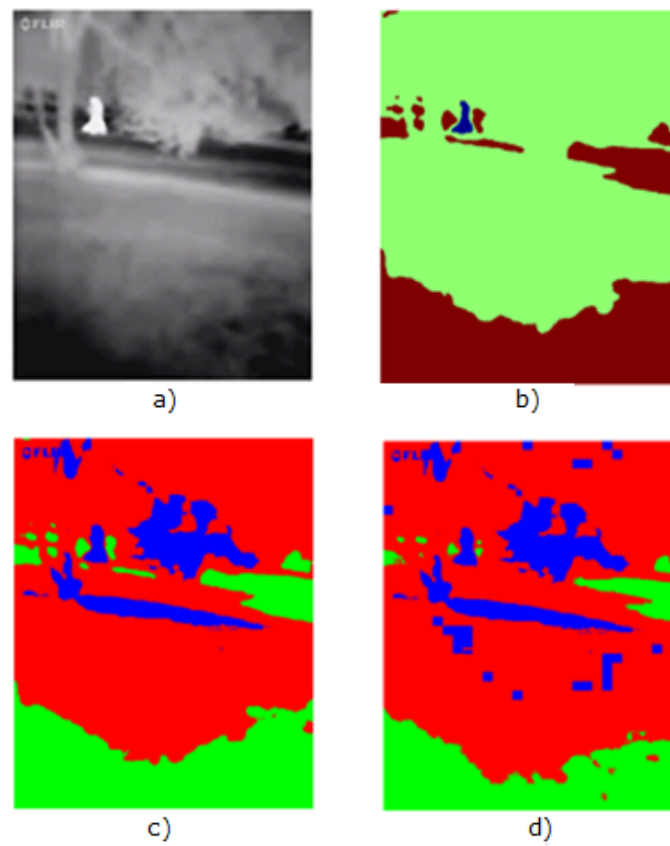


FIGURA 4.22: Comparativa de los resultados tras la segmentación de la imagen a) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

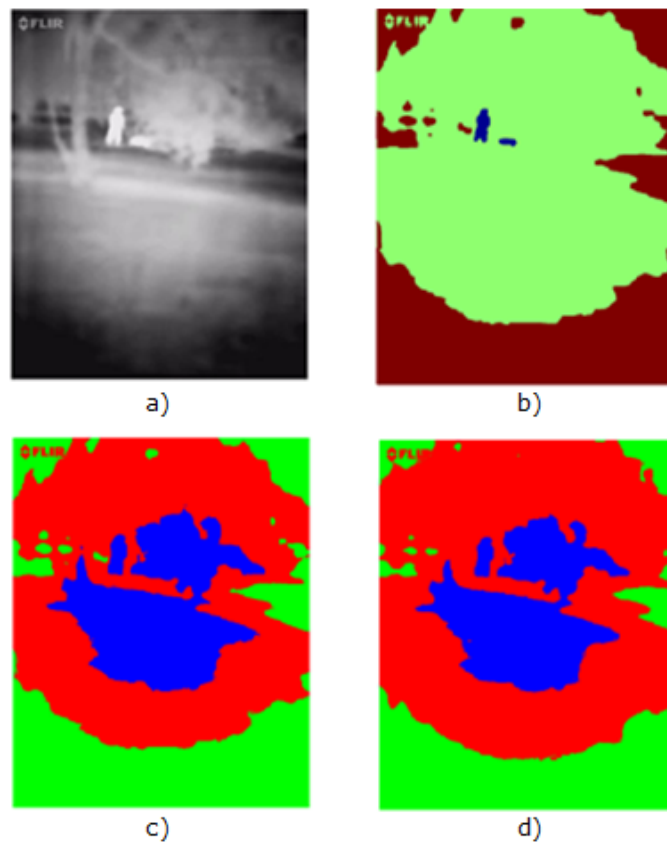


FIGURA 4.23: Comparativa de los resultados tras la segmentación de la imagen b) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

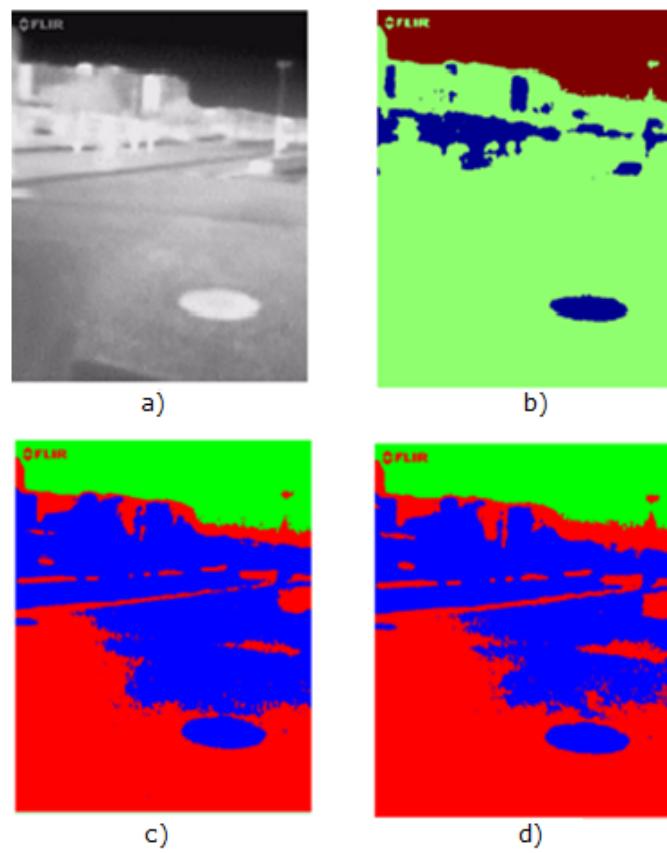


FIGURA 4.24: Comparativa de los resultados tras la segmentación de la imagen c) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

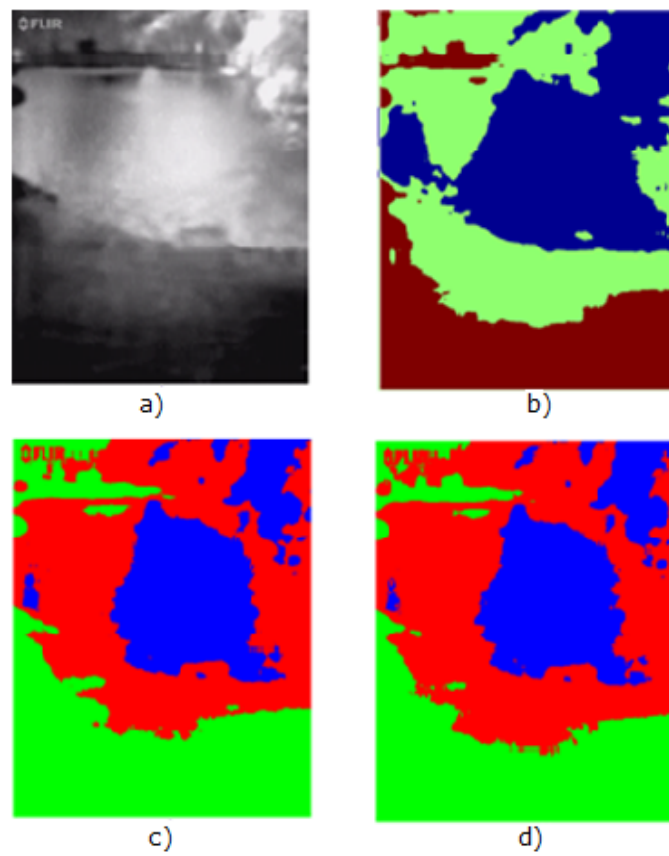


FIGURA 4.25: Comparativa de los resultados tras la segmentación de la imagen d) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

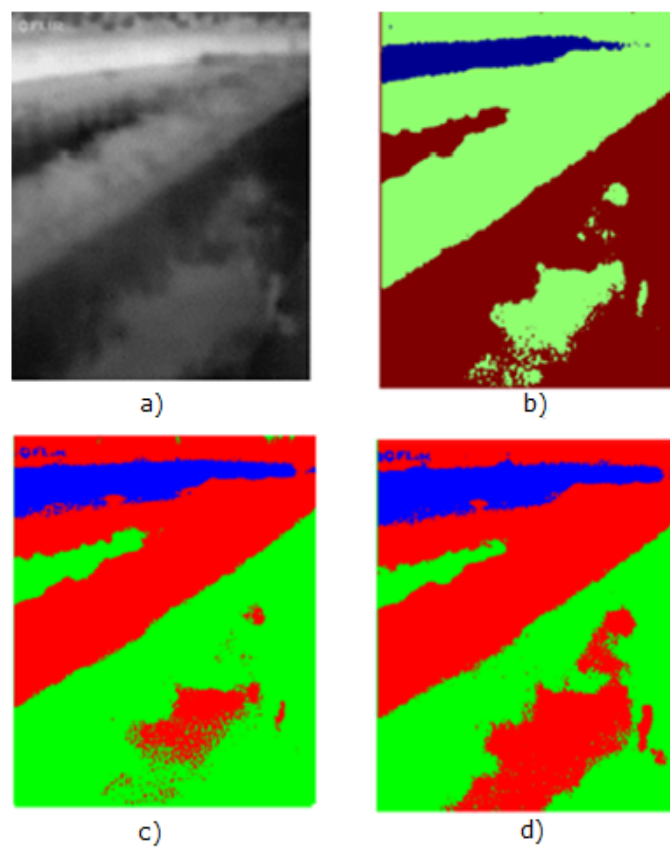


FIGURA 4.26: Comparativa de los resultados tras la segmentación de la imagen e) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

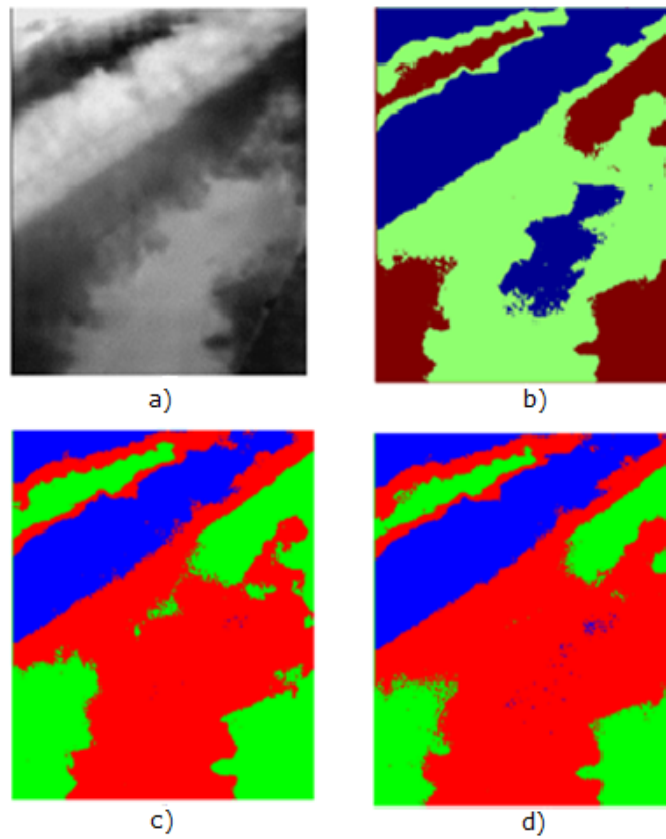


FIGURA 4.27: Comparativa de los resultados tras la segmentación de la imagen f) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

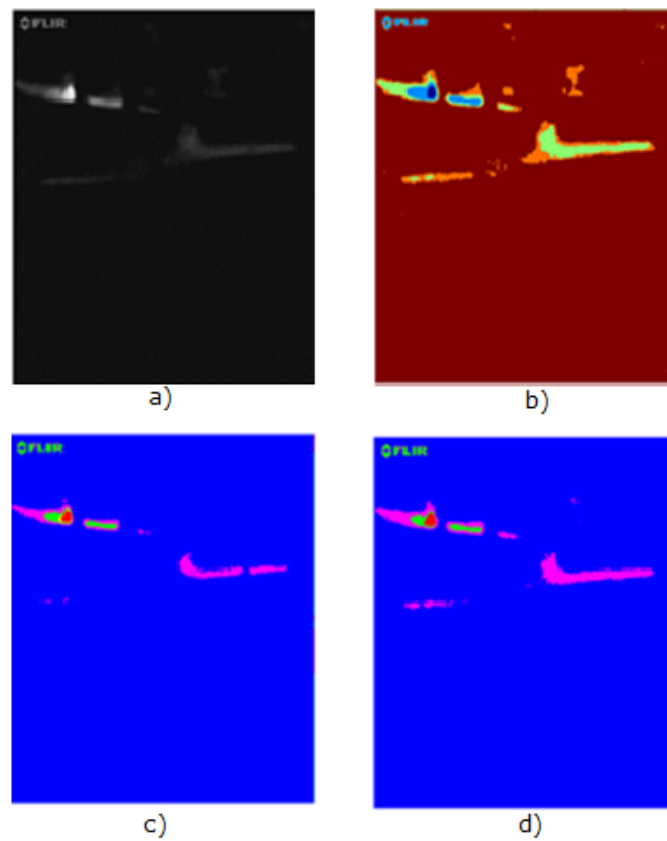


FIGURA 4.28: Comparativa de los resultados tras la segmentación de la imagen g) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

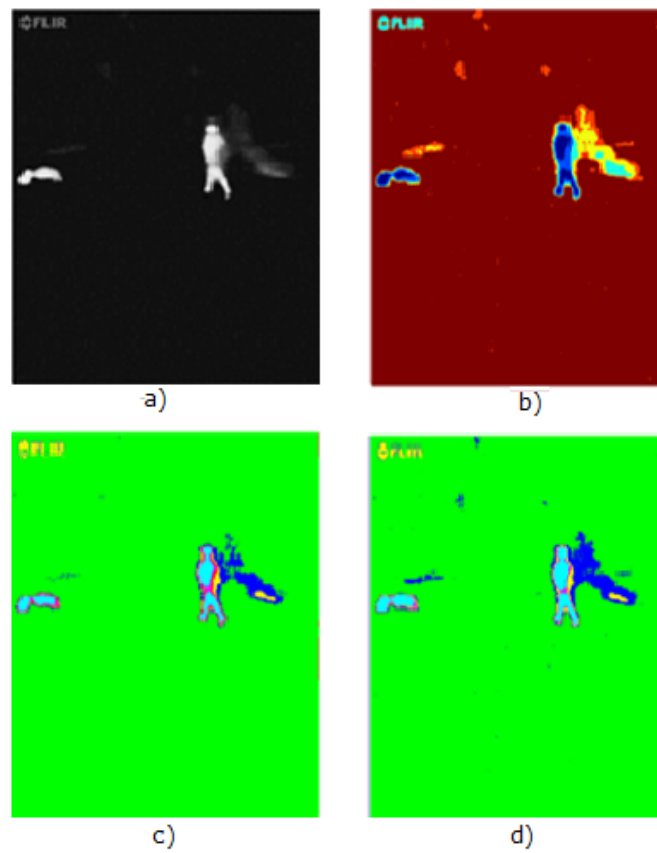


FIGURA 4.29: Comparativa de los resultados tras la segmentación de la imagen h) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

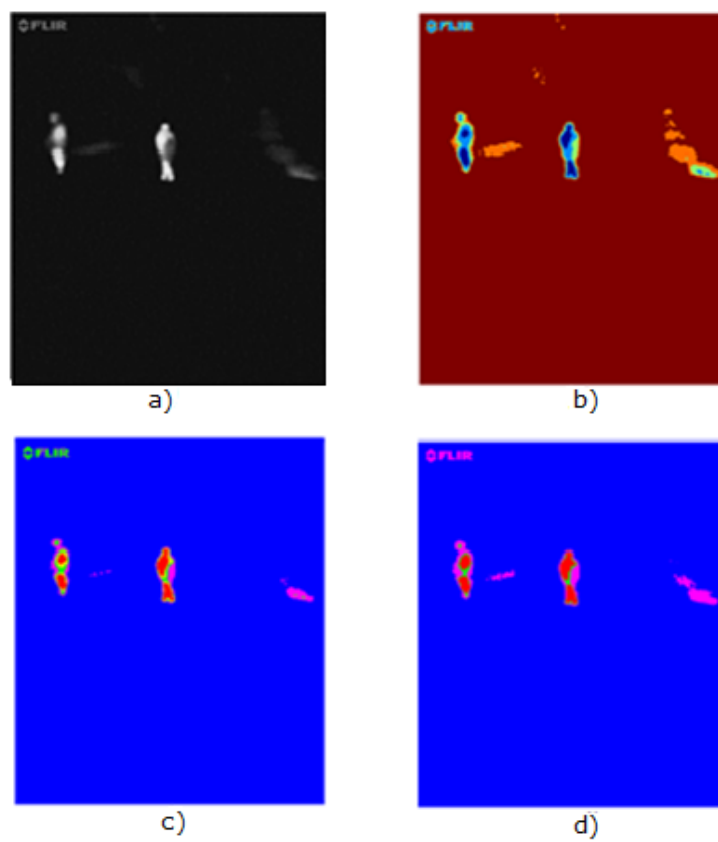


FIGURA 4.30: Comparativa de los resultados tras la segmentación de la imagen i) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

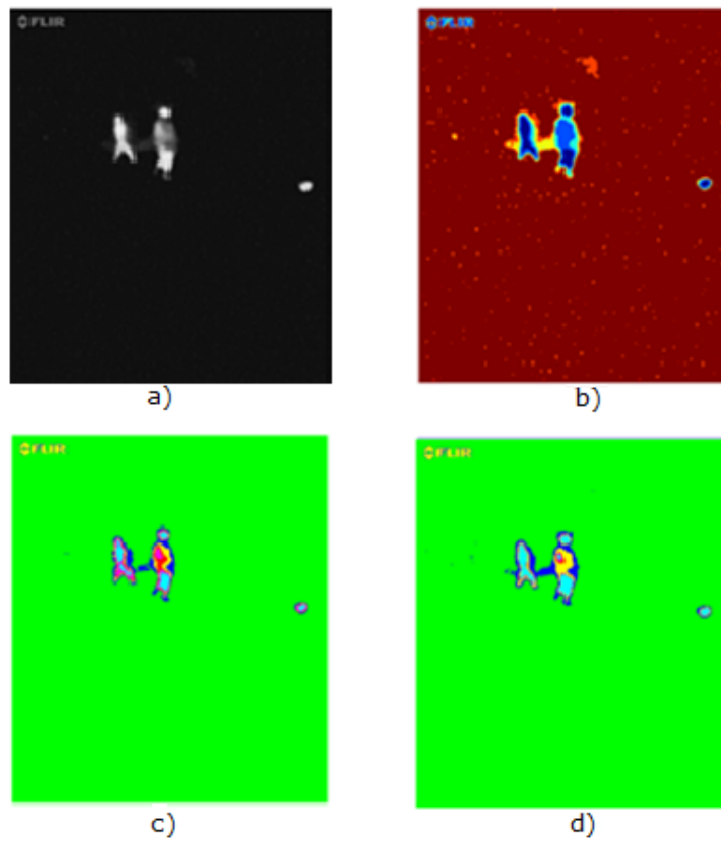


FIGURA 4.31: Comparativa de los resultados tras la segmentación de la imagen j) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

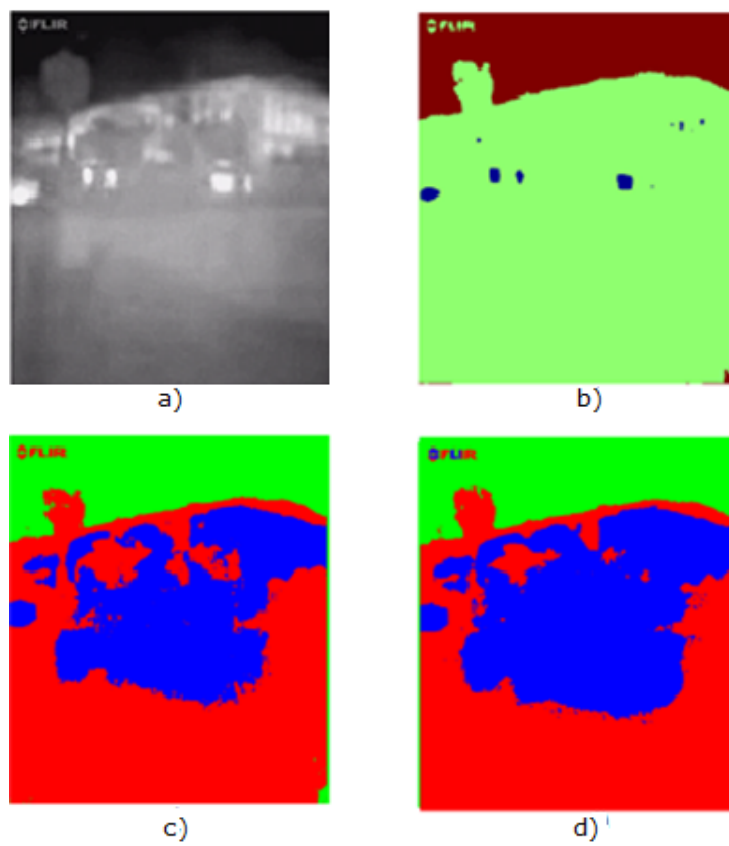


FIGURA 4.32: Comparativa de los resultados tras la segmentación de la imagen k) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

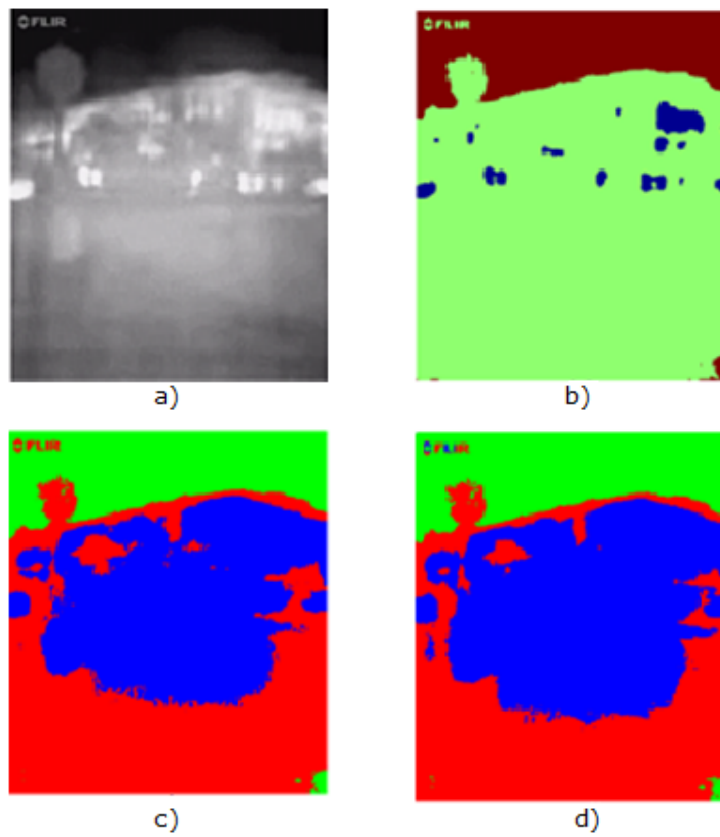


FIGURA 4.33: Comparativa de los resultados tras la segmentación de la imagen 1) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

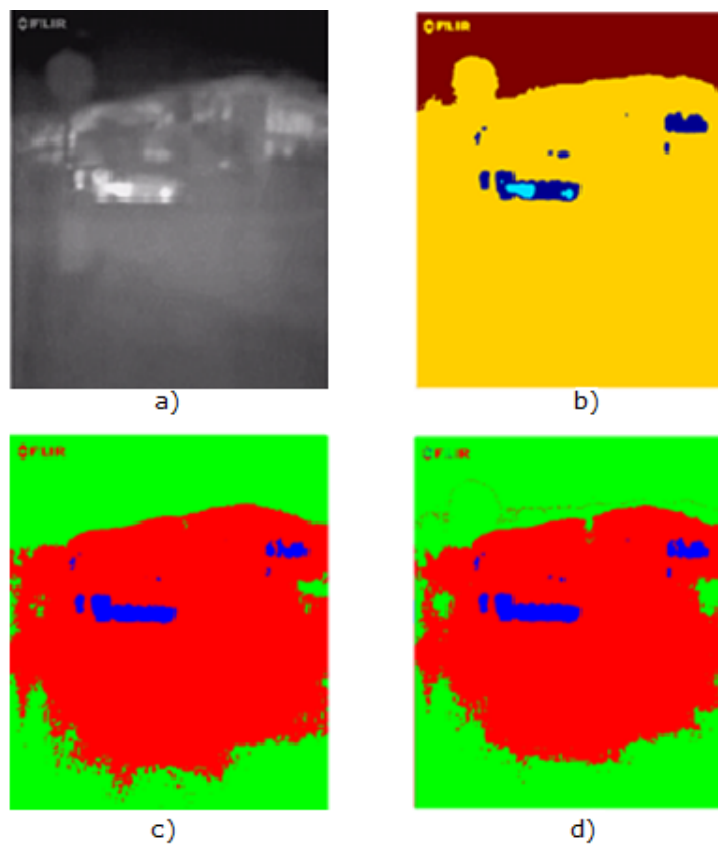


FIGURA 4.34: Comparativa de los resultados tras la segmentación de la imagen m) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

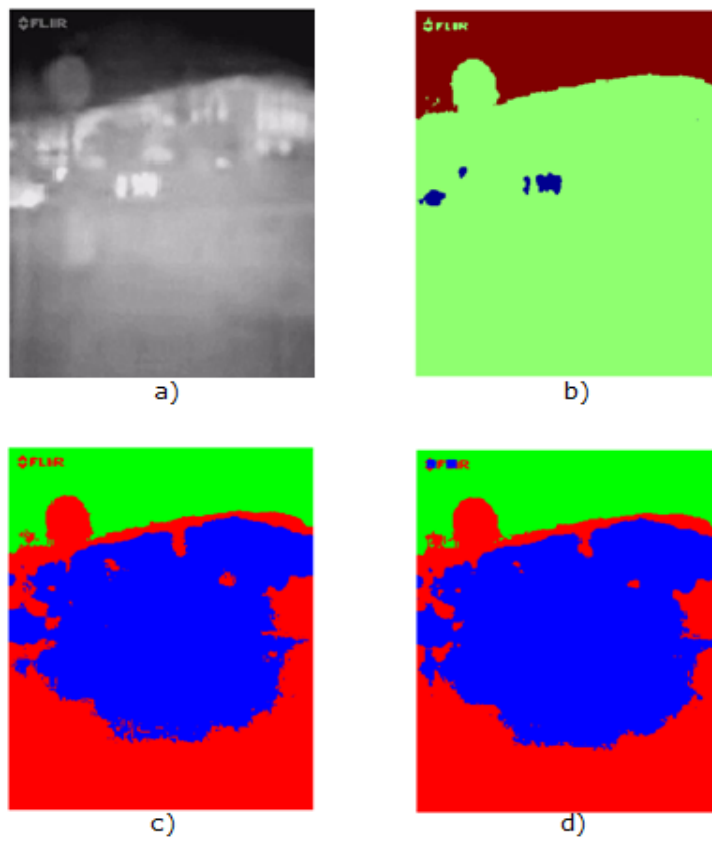


FIGURA 4.35: Comparativa de los resultados tras la segmentación de la imagen n) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

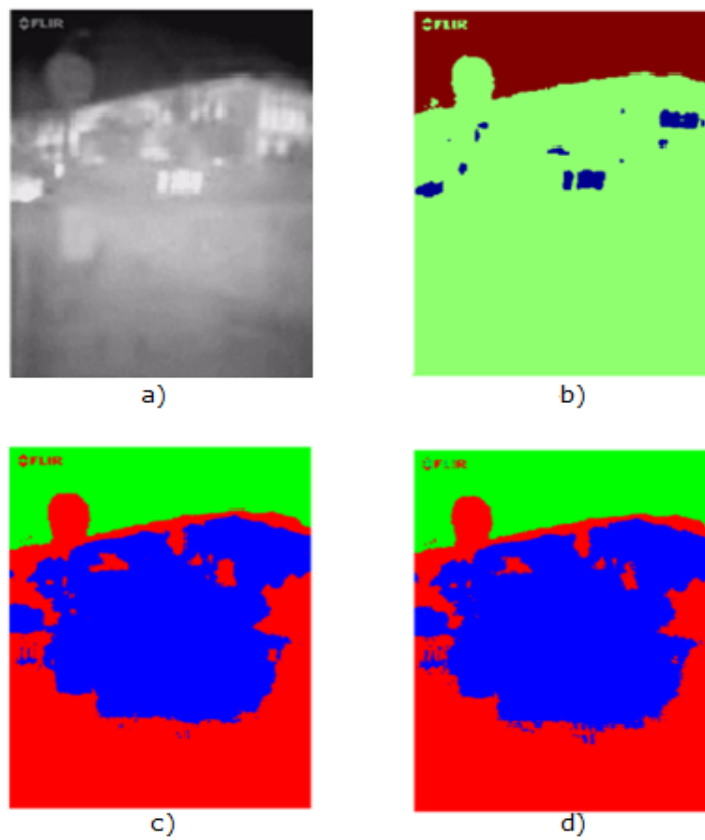


FIGURA 4.36: Comparativa de los resultados tras la segmentación de la imagen o) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

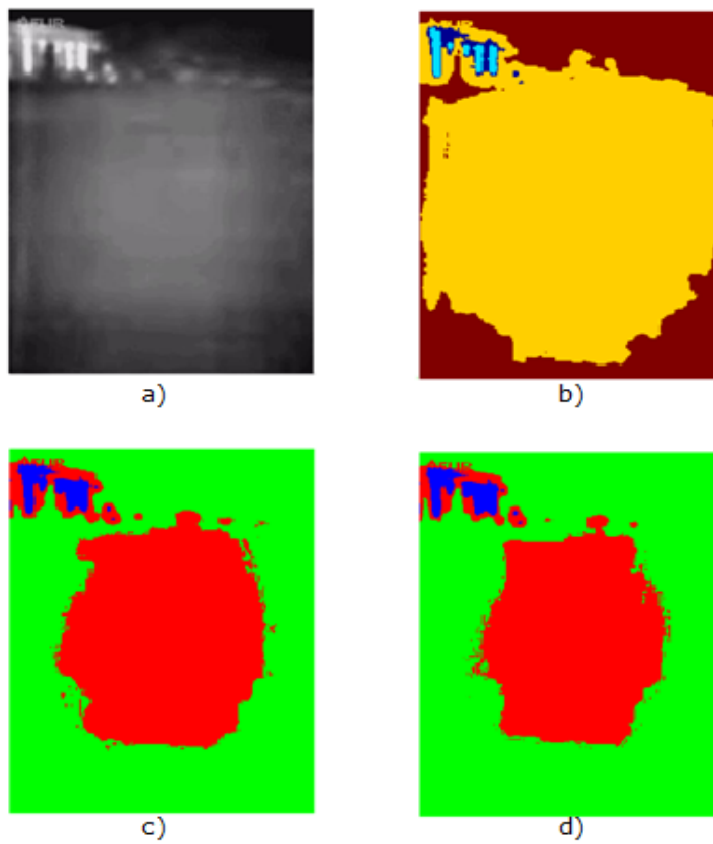


FIGURA 4.37: Comparativa de los resultados tras la segmentación de la imagen p) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

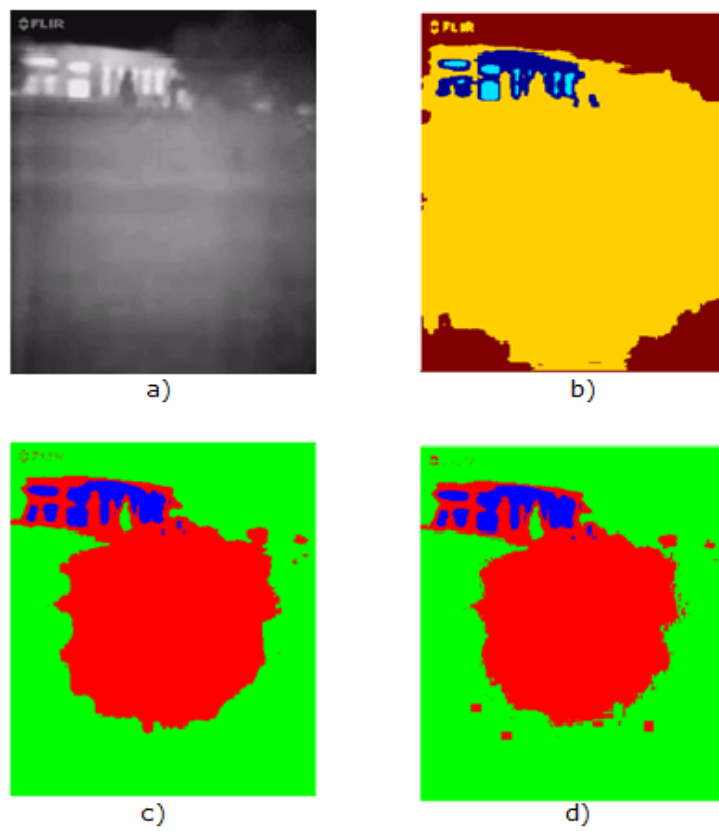


FIGURA 4.38: Comparativa de los resultados tras la segmentación de la imagen q) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano.

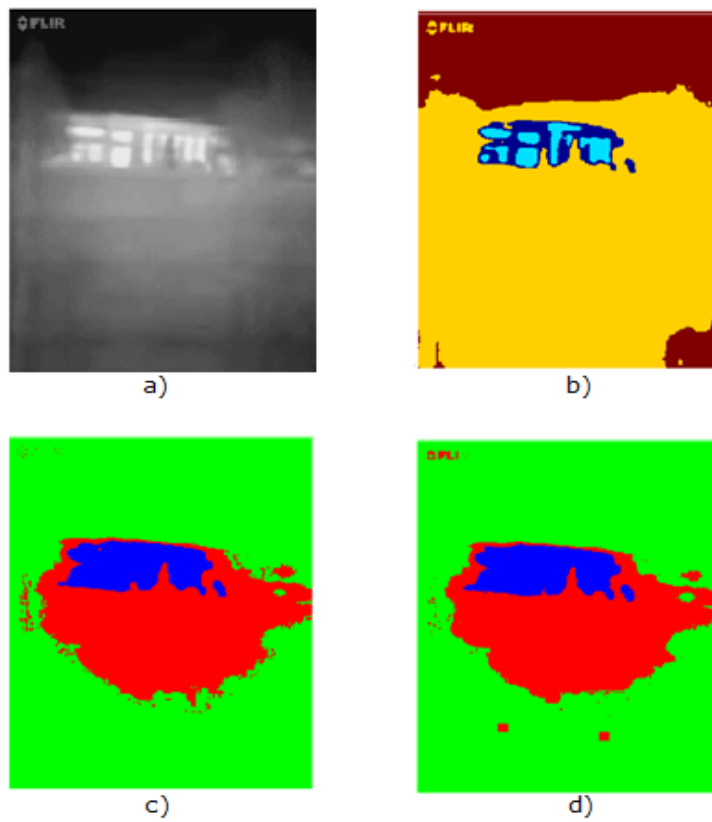


FIGURA 4.39: Comparativa de los resultados tras la segmentación de la imagen r) obtenidos por el algoritmo propuesto y por los clasificadores Fuzzy y Bayesiano. a) Imagen original, b) Algoritmo propuesto, c) Clasificador Fuzzy, d) Clasificador Bayesiano

4.3. Comparativa cuantitativa. Cálculo de errores en la segmentación.

Tras el análisis cualitativo de los resultados, la valoración de éstos mediante la definición de una métrica de evaluación numérica que refleje el error cometido por cada uno de los métodos de segmentación empleados ayudaría a esclarecer las conclusiones a extraer en lo referente a la eficacia de cada método para realizar una correcta segmentación.

Para ello, se han seleccionado 11 imágenes del conjunto de las 18 utilizadas previamente y se han generado de manera sintética los *ground truths* de ellas. Dicha selección se efectuó bajo la premisa de escoger imágenes donde los elementos a detectar fueran evidentes. La Figura 4.40 muestra dichos mapas de evaluación los cuales están etiquetados según las imágenes de las que provengan. Por ejemplo, para las imágenes a) y b) los elementos a detectar son el individuo y el perro que pasean por el parque, para las imágenes k) - o) las personas y vehículos que circulan por la escena, y así sucesivamente para el resto.

Existen muchas métricas de evaluación de resultados en la literatura [29], [30], [31], [32], como son: la sensibilidad (ecuación 4.1) que indica la relación entre el número total de verdaderos positivos (TP) y el número total de positivos (TP + FN), siendo FN los falsos negativos; la especificidad (ecuación 4.2) que indica la relación entre los verdaderos negativos (TN) y el número total de negativos (TN + FP), siendo FP los falsos positivos; la precisión (ecuación 4.3) que muestra la relación entre los verdaderos positivos y todos los píxeles que se han marcado como positivos (TP + FP); el ratio de falsos negativos (ecuación 4.4) que muestra la relación entre los falsos negativos cometidos y la cantidad total de posibles candidatos, es decir, todos los positivos; el ratio de falsos positivos (ecuación 4.5) que muestra la relación entre los falsos positivos cometidos y la cantidad total de posibles candidatos, es decir, todos los negativos, y por último, la exactitud (ecuación 4.6) que muestra la relación entre los píxeles que han sido correctamente clasificados (TP + TN) entre el número total de píxeles en la imagen (TP + TN + FP + FN).

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4.1)$$

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (4.2)$$

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4.3)$$

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}} \quad (4.4)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (4.5)$$

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (4.6)$$

Sin embargo, ninguna de ellas recoge la posibilidad de analizar el total de errores cometidos en la clasificación de positivos y negativos, es decir, la sumatoria de FP + FN entre el número total de píxeles que se desea clasificar correctamente, es decir, los positivos. A lo largo de esta exposición se hace referencia como conjunto positivo al número de píxeles marcados con un 1 en el *ground truth*, es decir, los objetos de interés, y como negativo al resto de elementos del entorno o *background*. Debido a ello, surge la necesidad de definir una nueva métrica de evaluación denominada *Error de Clasificación* que nos da una idea del tanto por ciento del error cometido por cada método de segmentación para definir correctamente los límites de la región o regiones que engloban los elementos a detectar. Es decir, los *ground truths* son la referencia del área abarcada por dichos elementos, los cuales podrán estar definidos por una o varias regiones siempre y cuando éstas no se salgan de los márgenes definidos por los mapas de evaluación ya que lo contrario se considerará un error en la segmentación. Un ejemplo claro podría verse en los resultados obtenidos por el MOM para la imagen b) puesto que la región que engloba al individuo y a su perro abarca también parte del césped y del árbol situado en primer plano. Estas partes del *background* englobadas dentro de la misma región que los elementos a detectar se consideran FP lo que aumentará el error en la clasificación. Por lo tanto, la nueva métrica de evaluación se defino como:

$$\text{Error}(\%) = \frac{\text{FP} + \text{FN}}{\sum_{i=1}^N (\text{GT}_i = 1)} \quad (4.7)$$

siendo FP los falsos positivos (número de píxeles del *background* englobados dentro de la misma región que los elementos a detectar), FN son los falsos negativos (número de píxeles de los elementos a detectar clasificados dentro de otras regiones), N representa el número total de píxeles de la imagen bajo análisis y $\sum_{i=1}^N (\text{GT}_i = 1)$ es la sumatoria de los píxeles del *ground truth* a detectar y que se han considerado etiquetados con el número 1. Mayores valores de error para el mismo número de píxeles a detectar, significa mayor número de FP y/o FN cometidos, lo que implica peores resultados en la clasificación.

La Tabla 4.1 muestra los valores de errores cometidos por cada algoritmo de segmentación para las 11 imágenes de test seleccionadas. En el caso del MOM también se especifica el número de niveles de umbral utilizado. Por simplicidad, dichos valores se han representado sobre el gráfico recogido en la Figura 4.41. Se puede observar, cómo los clasificadores Fuzzy y Bayesiano son los que mayores errores cometen de manera general para todas las imágenes consideradas, mostrando el método de Fuzzy una ligera tendencia a menores errores que el método Bayesiano. No obstante, para las imágenes j), h), i), j) y m) los errores cometidos por dichos métodos son considerablemente menores que para el resto de imágenes. Ello es debido a que, como ya se ha comentado con anterioridad, el comportamiento de los métodos de clasificación supervisados está estrechamente ligado al conjunto de datos de entrenamiento que para estos casos se ha escogido de manera aleatoria utilizando para ello 5 imágenes distintas del conjunto de datos disponible para cada caso de estudio. Finalmente, el MOM y el método propuesto presentan un comportamiento bastante estable con errores bastante bajos para todos los casos, presentando el método propuesto incluso menores valores de error que el MOM (obsérvese cómo la curva de color violeta representativa del método propuesto queda por debajo que la curva de color azul representativa del MOM para casi todos los casos).

Dicha apreciación podría parecer contradictoria tras el análisis cualitativo efectuado anteriormente puesto que a simple vista el MOM parece delimitar mejor los bordes de los elementos a detectar. Ello se debe a que el MOM considera muchos más elementos del entorno dentro de las regiones correspondientes con los elementos a detectar, generando así mayores valores de FP. Quizás otra combinación de *ground*

truths arrojaría resultados distintos. No obstante, no cabe duda que los resultados de la segmentación arrojados por el método propuesto y por el MOM son bastante aceptables, por lo que se puede concluir que el método propuesto es un buen estimador del número de regiones mínimas necesarias para definir los elementos mas importantes de una escena.

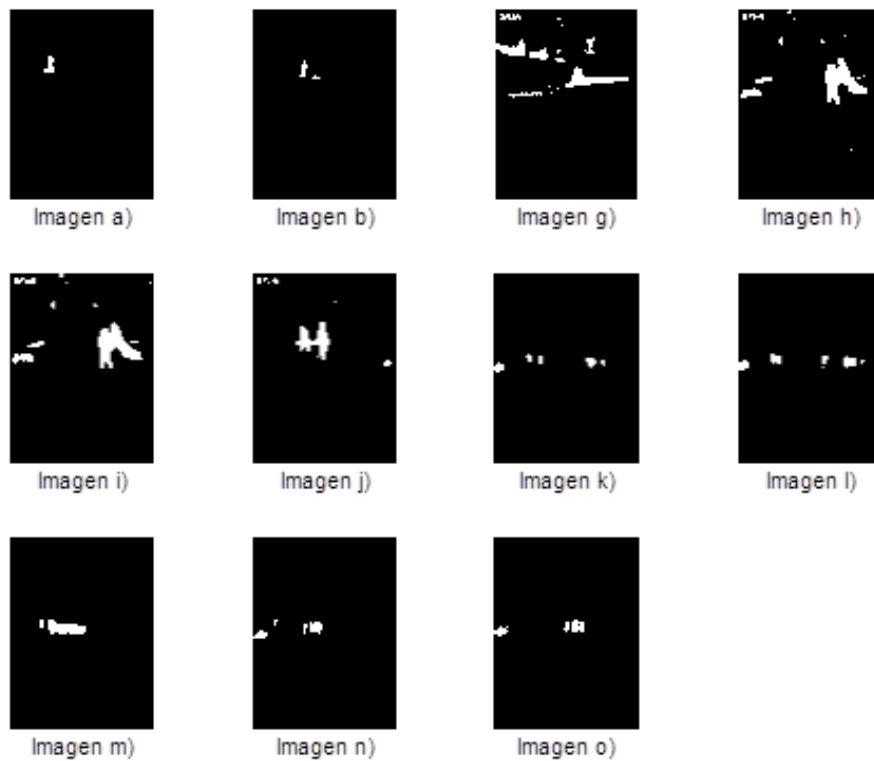


FIGURA 4.40: Ground-truths de 11 imágenes del data set.

	Imagen a)	Imagen b)	Imagen g)	Imagen h)
	Error (%)	Error (%)	Error (%)	Error (%)
Método Otsu Multiumbral	1,9197 (6 niveles)	11,093 (6 niveles)	0,1504 (6 niveles)	0,1287 (6 niveles)
Clasificador Fuzzy	11,9514	37,205	0,7744	0,2953
Clasificador Bayesiano	14,6567	36,018	0,6750	0,1432
Método Propuesto	0,1935	0,0548	0,0333	0,0167

	Imagen i)	Imagen j)	Imagen k)	Imagen l)
	Error (%)	Error (%)	Error (%)	Error (%)
Método Otsu Multiumbral	1,3754 (6 niveles)	0,0377 (6 niveles)	3,9261 (5 niveles)	3,3550 (5 niveles)
Clasificador Fuzzy	1,2593	0,1933	18,1630	23,2923
Clasificador Bayesiano	1,3241	0,1108	23,9578	25,1094
Método Propuesto	1,4067	0,6057	0,4603	1,2125

	Imagen m)	Imagen n)	Imagen o)
	Error (%)	Error (%)	Error (%)
Método Otsu Multiumbral	2,6740 (5 niveles)	3,6528 (7 niveles)	3,5310 (7 niveles)
Clasificador Fuzzy	0,3758	41,8378	38,5758
Clasificador Bayesiano	0,5119	46,4250	42,4107
Método Propuesto	1,5002	0,1959	0,8468

TABLA 4.1: Tabla con los errores en la clasificación (%) cometidos por los distintos algoritmos para 11 de las imágenes del data set.

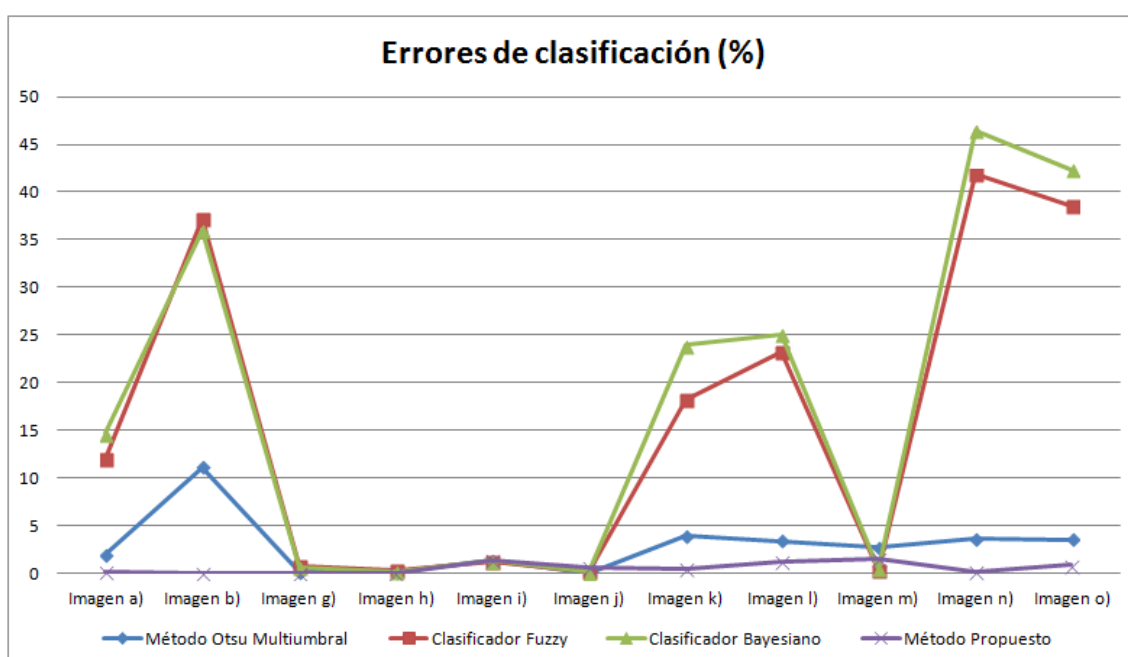


FIGURA 4.41: Gráfico con los errores en la clasificación (%) cometidos por los distintos algoritmos para 11 de las imágenes del data set.

Capítulo 5

Conclusiones y líneas futuras

Los resultados de la evaluación muestran la viabilidad del algoritmo propuesto como método de segmentación multinivel no supervisado y totalmente autónomo. Para la mayoría de las imágenes analizadas, especialmente las 11 imágenes seleccionadas para el análisis cuantitativo, el método propuesto ha sido capaz de estimar el número de regiones necesarias para definir a todos los objetos de interés presentes en ella, cuyos bordes quedan definidos con mayor o menor precisión dependiendo de las características de la imagen bajo análisis. Además, se comprueba que el algoritmo propuesto ha sido también capaz de abordar con éxito la condición marcada de agrupar todos aquellos píxeles del fondo o *background* con características similares en una única región común con el objetivo final de destacar los objetos de interés y definir el mínimo número de regiones realmente necesarias para dicho fin.

A simple vista, los bordes de los elementos de interés parecen mejor definidos con el MOM para un número de regiones a definir mayor en términos generales que el definido por nuestro método. Sin embargo, se ha comprobado que a medida que el número de niveles de umbral aumenta, más elementos son identificados pero también muchas otras regiones que no muestran información nueva, generándose mapas de segmentación algo más engorrosos con múltiples zonas que representan lo mismo y que por tanto, podrían fusionarse con otras vecinas. Por el contrario, la tendencia del método propuesto es definir el mayor número de objetos de interés cuyos niveles de grises difieran significativamente del resto de elementos que más predominan en la escena, bajo la premisa de generar el menor número de regiones posibles. Por tanto, los mapas de segmentación resultado son más fáciles de interpretar puesto que

dividen la imagen en un número de zonas suficiente como para detallar correctamente los elementos de mayor interés.

Los clasificadores Fuzzy C-means y Bayesiano no han podido generar unos modelos de segmentación para el mismo número de regiones estimadas por el método propuesto capaces de dividir la imagen de manera eficiente como para delimitar los elementos de interés presentes en ellas. Se puede concluir una vez más que el método propuesto es el único capaz de generar unos resultados de segmentación aceptables con un número de regiones mínimo a definir para la mayoría de las imágenes a analizar. También se ha demostrado que el comportamiento de los métodos de clasificación supervisados está estrechamente ligado al conjunto de datos de entrenamiento. En las pruebas realizadas dichos datos se escogieron de manera aleatoria utilizando para ello 5 imágenes distintas del conjunto de datos disponible para cada caso bajo estudio. Otro conjunto de datos quizás hubiera arrojado resultados diferentes, lo que demuestra la escasa robustez de estos métodos para asegurar un correcto funcionamiento en otras aplicaciones con imágenes reales.

Una de las principales desventajas del método propuesto es que es algo sensible al ruido, como se mostró con la imagen de test j). Una posible solución sería estudiar la posibilidad de encontrar un factor de parada adicional durante el proceso de extracción de los elementos más oscuros de la imagen. En vez de que dicho proceso finalice cuando el vector $\mathbf{level}_{\mathbf{final}}$ o la matriz $\mathbf{IMG}_{\mathbf{reduc}}$ estén vacías, una posible línea de investigación futura sería estudiar la posibilidad de finalizar el proceso cuando dos umbrales contenidos en $\mathbf{level}_{\mathbf{final}}$ estén lo suficientemente próximos.

Otra posible línea futura de mejora sería solucionar el problema de redondeo al cambiar de escala los umbrales contenidos en $\mathbf{level}_{\mathbf{final}}$, con lo que ningún píxel disperso quedaría sin etiquetar y por tanto, no sería necesaria ejecutar el último paso del algoritmo propuesto donde dichos píxeles se clasifican según el valor de la moda de sus píxeles vecinos.

Finalmente, indicar que otra posible línea futura de investigación podría centrarse en encontrar otra condición de evaluación de umbrales para determinar si es necesario continuar estudiando menores o menor número de regiones del histograma, que no implique el cálculo del factor de decisión y su comparación con el ratio de varianzas entre la iteración actual y la anterior. El objetivo sería encontrar otra métrica de evaluación que quizás ayude a delimitar mejor los bordes de los elementos de interés encontrados.

Referencias

- [1] Mehmet Sezgin et al. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, 13(1):146–168, 2004.
- [2] CK Chow and T Kaneko. Automatic boundary detection of the left ventricle from cineangiograms. *Computers and biomedical research*, 5(4):388–410, 1972.
- [3] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
- [4] Hangzhou Sun, Yixin Ma, Huawei Wu, Liwen Miao, Xinyi Chen, Xiaojun Ji, and Ping Cai. An improved otsu’s method for ct image boundary contour extraction. In *Imaging Systems and Techniques (IST), 2016 IEEE International Conference on*, pages 493–497. IEEE, 2016.
- [5] Mohamed Cheriet, Joseph N Said, and Ching Y Suen. A recursive thresholding technique for image segmentation. *IEEE transactions on image processing*, 7(6):918–921, 1998.
- [6] Oliver Nina, Bryan Morse, and William Barrett. A recursive otsu thresholding method for scanned document binarization. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 307–314. IEEE, 2011.
- [7] Ping-Sung Liao, Tse-Sheng Chen, Pau-Choo Chung, et al. A fast algorithm for multilevel thresholding. *J. Inf. Sci. Eng.*, 17(5):713–727, 2001.
- [8] Rafael Guillermo Gonzalez Acuña, Junli Tao, and Reinhard Klette. Generalization of otsu’s binarization into recursive colour image segmentation. In *Image and Vision Computing New Zealand (IVCNZ), 2015 International Conference on*, pages 1–6. IEEE, 2015.

-
- [9] Rolf Adams and Leanne Bischof. Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence*, 16(6):641–647, 1994.
- [10] Jianping Fan, Guihua Zeng, Mathurin Body, and Mohand-Said Hacid. Seeded region growing: an extensive and comparative study. *Pattern recognition letters*, 26(8):1139–1156, 2005.
- [11] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.
- [12] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.
- [13] James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3):191–203, 1984.
- [14] M-S Yang. A survey of fuzzy clustering. *Mathematical and Computer modelling*, 18(11):1–16, 1993.
- [15] Mohamed S Kamel and Shokri Z Selim. New algorithms for solving the fuzzy clustering problem. *Pattern recognition*, 27(3):421–428, 1994.
- [16] James C Bezdek, Institute of Electrical, Electronics Engineers (New York). Neural Networks Council, and Sankar K Pal. *Fuzzy models for pattern recognition: methods that search for structures in data*. IEEE press, 1992.
- [17] Shokri Z Selim and MS Kamel. On the mathematical and numerical properties of the fuzzy c-means algorithm. *Fuzzy sets and Systems*, 49(2):181–191, 1992.
- [18] Geoffrey H Ball and David J Hall. Isodata, a novel method of data analysis and pattern classification. Technical report, Stanford research inst Menlo Park CA, 1965.
- [19] Pajares Gonzalo and M de La Cruz Jesús. Visión por computador: Imágenes digitales y aplicaciones. *Editorial Alfaomega. Segunda edición. USA*, page 740, 2008.
- [20] Eamonn J Keogh and Michael J Pazzani. Learning augmented bayesian classifiers: A comparison of distribution-based and classification-based approaches. In *AISTats*, 1999.

-
- [21] Vladimir Naumovich Vapnik and Vladimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.
- [22] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. Wiley, New York, 1973.
- [23] M Sanz G Pajares and J De La Cruz. Aprendizaje automático un enfoque práctico. *RA-MA SA Editorial y Publicaciones. ISBN-10: 8499640117*, 2010.
- [24] BW Sillverman, MC Jones, E Fix, and JL Hodges. An important contribution to nonparametric discriminant analysis and density estimation. *International Statistical Review*, 57(3):233–247, 1951.
- [25] Songbo Tan. Neighbor-weighted k-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, 28(4):667–671, 2005.
- [26] Charles Boncelet. Handbook of image and video processing, chapter image noise models, 2005.
- [27] Jun Ohta. *Smart CMOS image sensors and applications*. CRC press, 2007.
- [28] FLIR ONE ®. <http://www.flir.com/flirone/>, 2017.
- [29] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.
- [30] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2011.
- [31] Stephen V Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, 62(1):77–89, 1997.
- [32] Kai Ming Ting. Confusion matrix. In *Encyclopedia of machine learning*, pages 209–209. Springer, 2011.