



UNIVERSIDAD AUTÓNOMA DE YUCATÁN

FACULTAD DE MATEMÁTICAS

ENSAMBLE DE REDES NEURONALES
CONVOLUCIONALES PARA LA SEGMENTACIÓN DE
CÉLULAS PROCESADAS POR ELECTROFORESIS EN
GEL

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

Ingeniero en Computación

PRESENTA:

Daniel Johsue Ruz Suárez

ASESORA DE TESIS:

Dra. Anabel Martín González

Mérida, Yucatán, 2021



A la Facultad de Matemáticas.

Reconocimientos

Quiero agradecer:

- A mis padres, Sergio Ruz y Mayelin Suárez, por haberme apoyado económicamente en todo mi etapa académica y estar siempre al pendiente de mí, pero sobre todo por darme la educación que me permitiera ser la persona que soy hoy en día.
- A mi asesora, la Dra. Anabel Martín, por haber visto el potencial en mí, por darme los conocimientos y herramientas que me permitieron desarrollar la presente tesis, y por hacer que me interese el campo de la inteligencia artificial.
- A mi pareja, Bárbara Pérez, por haberme motivado y apoyado incondicionalmente durante toda mi etapa universitaria, por sus consejos y por ayudarme cuando más lo necesité.

Resumen

La electroforesis en gel de células individuales, también llamada ensayo cometa, es una técnica que permite conocer el daño del Ácido Desoxirribonucleico (ADN) de una única célula. Debido a que este ensayo es económico, simple y fiable, es comúnmente usado en áreas como el biomonitoreo humano, la genotoxicidad y en el estudio de enfermedades degenerativas como el cáncer y la diabetes. La técnica del ensayo cometa, provoca que las células adquieran un aspecto semejante a la de un cometa, integrado por una parte con ADN sin daño y por otra parte conformada por fragmentos de ADN muy dañados, las cuales se denominan cabeza y cola del cometa respectivamente.

El grado de daño de las células, provoca que los cometas, así como sus respectivas cabezas y colas, presenten diversas variaciones en tamaño, longitud y proporción de ADN. Por lo anterior, es imprescindible que sea posible identificar cada célula de manera individual para poder extraer sus características deseadas en su correspondiente aplicación. Por ello y con la finalidad de automatizar la obtención de estos parámetros se han propuesto en los últimos años, sistemas computacionales capaces de segmentar los cometas contenidos en una imagen tomada desde un microscopio.

El presente trabajo, se propone un sistema totalmente automático, basado en un ensamble de redes neuronales convolucionales U-Net, con el objetivo de segmentar las cabezas y colas de los cometas, y además extraer sus características. De igual manera, se implementan técnicas para mejorar la segmentación realizada por el ensamble, así como un algoritmo que sea capaz de dividir los cometas traslapados. Con respecto a la métrica F1-score, los resultados obtenidos por este sistema propuesto indican un mejor desempeño de segmentación que otros sistemas computacionales basados en umbralización.

Índice general

Índice de figuras	XI
Índice de tablas	XV
1. Introducción	17
1.1. Antecedentes	18
1.1.1. Sistemas semi-automáticos	18
1.1.2. Sistemas automáticos	20
1.1.3. Sistemas con aprendizaje automático	22
1.2. Contribuciones	23
1.3. Estructura de la tesis	24
1.4. Publicaciones	24
2. Marco Teórico	25
2.1. Ensayo cometa	25
2.2. Segmentación	27
2.2.1. Segmentación binaria	28
2.2.2. Segmentación multiclase	28
2.2.3. Segmentación por transformación divisoria	29
2.2.4. Evaluación de la segmentación	30
2.2.4.1. Matriz de confusión	30
2.2.4.2. Métricas de desempeño	31
2.3. Redes Neuronales Convolucionales	33
2.3.1. Kernel	34
2.3.2. Convolución	35
2.3.3. Convolución transpuesta	36
2.3.4. Submuestreo	37
2.3.5. Sobremuestreo	37
2.3.6. Función de activación	38
2.3.6.1. ReLU	38
2.3.6.2. Softmax	38
2.3.7. Función de costo	39
2.3.8. Descenso de gradiente	40

ÍNDICE GENERAL

2.3.9. Optimizador Adam	40
2.3.10. Entrenamiento	41
2.3.11. Sobreajuste	42
2.3.11.1. Validación de retención	42
2.3.12. Red neuronal convolucional U-Net	44
2.3.13. Ensamble de redes neuronales convolucionales	45
2.4. Convex Hull	46
2.5. Morfología matemática	48
2.5.1. Dilatación	49
2.5.2. Erosión	49
2.5.3. Extracción de componentes conectados	50
2.5.4. Rellenado de orificios	51
2.5.5. Detección de bordes de regiones etiquetadas	52
3. Metodología	53
3.1. Funcionamiento general del sistema	53
3.2. Base de datos	54
3.2.1. Ajuste y segmentación manual de la base de datos	54
3.3. Arquitectura	56
3.4. Entrenamiento	57
3.5. Módulo de segmentación	59
3.6. Módulo de refinamiento	61
3.6.1. Eliminación de cometas con probabilidad baja	62
3.6.2. Separación de cometas con ligero traslape	63
3.6.3. Eliminación de cabezas de posibles cometas traslapados	64
3.6.4. Aplicación de la envolvente convexa	65
3.6.5. Relleno de orificios en cometas traslapados	66
3.6.6. Eliminación de cometas con áreas anómalas	67
3.7. Módulo de partición y etiquetado	68
3.8. Módulo de extracción de características	70
3.9. Módulo de presentación	71
4. Resultados	73
4.1. Análisis del aprendizaje de las redes neuronales convolucionales	73
4.2. Análisis de la segmentación	75
4.3. Comparación de segmentación con otros sistemas computacionales	86
4.4. Análisis y comparación de los resultados finales	97
4.5. Análisis y comparación de los parámetros calculados	107
5. Conclusiones	113
5.1. Trabajo futuro	113
Bibliografía	115

Índice de figuras

1.1. Partes que componen una célula procesada por el ensayo cometa	17
2.1. Representación de la migración de ADN	25
2.2. Muestras del ensayo cometa	26
2.3. Defectos en muestras del ensayo cometa	26
2.4. Tipos de segmentación	27
2.5. Segmentación binaria	28
2.6. Segmentación multiclase	28
2.7. Codificación one-hot	29
2.8. Funcionamiento general de la transformación divisoria	29
2.9. Algoritmo de la transformación divisoria	30
2.10. Ejemplo de ground truth	30
2.11. Matriz de confusión	31
2.12. Aplicaciones de las redes neuronales convolucionales	33
2.13. Kernel de 3×3 con pesos genéricos	34
2.14. Ejemplos de kernel usados en filtrados espaciales	34
2.15. Proceso de convolución	35
2.16. Convolución con relleno de ceros y bias	36
2.17. Convolución transpuesta	36
2.18. Ejemplo de submuestro de 2×2	37
2.19. Ejemplo de sobremuestro de 2×2	37
2.20. Función de activación ReLU	38
2.21. Ejemplo de una salida de la Softmax	39
2.22. Tipos de ajustes del modelo a los datos	42
2.23. Esquemas de partición de conjuntos de datos	43
2.24. Curvas de aprendizaje	43
2.25. Arquitectura U-Net	44
2.26. Métodos de combinación de salidas de un ensamble	45
2.27. Conjuntos de puntos	46
2.28. Pasos del algoritmo de Quickhull	47
2.29. Formas comunes de elementos estructurantes de 3×3	48
2.30. Operaciones de vecindad con elemento estructurante cuadrado de 3×3	48
2.31. Dilatación con elemento estructurante cuadrado de 3×3	49

ÍNDICE DE FIGURAS

2.32. Erosión con elemento estructurante cuadrado de 3×3	49
2.33. Iteraciones del algoritmo de extracción de componentes conectados . . .	50
2.34. Iteraciones del algoritmo de rellenado de orificios	51
2.35. Iteraciones del algoritmo de detección de bordes	52
3.1. Diagrama general del sistema	53
3.2. Muestra de la base de datos	55
3.3. Proceso de segmentación manual	56
3.4. Diagrama de entrada/salida del ensamble	58
3.5. Proceso de conversión de imagen RGB a escala de grises	59
3.6. Proceso de barrido de segmentación	60
3.7. Proceso del módulo de segmentación	60
3.8. Errores comunes en la segmentación	61
3.9. Técnica de eliminación de cometas con probabilidad baja	62
3.10. Técnica de separación de cometas ligeramente traslapados	63
3.11. Técnica de eliminación de cabezas de posibles cometas traslapados . . .	64
3.12. Técnica de aplicación de envolvente convexa	65
3.13. Técnica de relleno de orificios en cometas traslapados	66
3.14. Técnica de eliminación de cometas con áreas anómalas	67
3.15. Etiquetación de cometas intactos	68
3.16. Etiquetación de cometas traslapados	69
3.17. Etiquetación de cometas	69
3.18. Salida del módulo de presentación	71
4.1. Curvas de aprendizaje de los modelos del ensamble	74
4.2. Segmentación de la imagen de prueba 1 realizada por los módulos de segmentación y refinamiento	76
4.3. Segmentación de la imagen de prueba 2 realizada por los módulos de segmentación y refinamiento	77
4.4. Segmentación de la imagen de prueba 3 realizada por los módulos de segmentación y refinamiento	78
4.5. Segmentación de la imagen de prueba 4 realizada por los módulos de segmentación y refinamiento	79
4.6. Segmentación de la imagen de prueba 5 realizada por los módulos de segmentación y refinamiento	80
4.7. Segmentación de la imagen de prueba 6 realizada por los módulos de segmentación y refinamiento	81
4.8. Segmentación de la imagen de prueba 7 realizada por los módulos de segmentación y refinamiento	82
4.9. Segmentación de la imagen de prueba 8 realizada por los módulos de segmentación y refinamiento	83
4.10. Segmentación de la imagen de prueba 9 realizada por los módulos de segmentación y refinamiento	84

4.11. Segmentación de la imagen de prueba 10 realizada por los módulos de segmentación y refinamiento	85
4.12. Segmentación de la imagen de prueba 1 realizada por el sistema propuesto, OpenComet y HiComet	87
4.13. Segmentación de la imagen de prueba 2 realizada por el sistema propuesto, OpenComet y HiComet	88
4.14. Segmentación de la imagen de prueba 3 realizada por el sistema propuesto, OpenComet y HiComet	89
4.15. Segmentación de la imagen de prueba 4 realizada por el sistema propuesto, OpenComet y HiComet	90
4.16. Segmentación de la imagen de prueba 5 realizada por el sistema propuesto, OpenComet y HiComet	91
4.17. Segmentación de la imagen de prueba 6 realizada por el sistema propuesto, OpenComet y HiComet	92
4.18. Segmentación de la imagen de prueba 7 realizada por el sistema propuesto, OpenComet y HiComet	93
4.19. Segmentación de la imagen de prueba 8 realizada por el sistema propuesto, OpenComet y HiComet	94
4.20. Segmentación de la imagen de prueba 9 realizada por el sistema propuesto, OpenComet y HiComet	95
4.21. Segmentación de la imagen de prueba 10 realizada por el sistema propuesto, OpenComet y HiComet	96
4.22. Resultados de la imagen de prueba 1 obtenidas del sistema propuesto, OpenComet y HiComet	97
4.23. Resultados de la imagen de prueba 2 obtenidas del sistema propuesto, OpenComet y HiComet	98
4.24. Resultados de la imagen de prueba 3 obtenidas del sistema propuesto, OpenComet y HiComet	99
4.25. Resultados de la imagen de prueba 4 obtenidas del sistema propuesto, OpenComet y HiComet	100
4.26. Resultados de la imagen de prueba 5 obtenidas del sistema propuesto, OpenComet y HiComet	101
4.27. Resultados de la imagen de prueba 6 obtenidas del sistema propuesto, OpenComet y HiComet	102
4.28. Resultados de la imagen de prueba 7 obtenidas del sistema propuesto, OpenComet y HiComet	103
4.29. Resultados de la imagen de prueba 8 obtenidas del sistema propuesto, OpenComet y HiComet	104
4.30. Resultados de la imagen de prueba 9 obtenidas del sistema propuesto, OpenComet y HiComet	105
4.31. Resultados de la imagen de prueba 10 obtenidas del sistema propuesto, OpenComet y HiComet	106
4.32. Diagrama de caja del área del cometa	107

ÍNDICE DE FIGURAS

4.33. Diagrama de caja del largo del cometa	107
4.34. Diagrama de caja del contenido de ADN en el cometa	108
4.35. Diagrama de caja de la intensidad promedio en el cometa	108
4.36. Diagrama de caja del área de la cabeza	108
4.37. Diagrama de caja del diámetro de la cabeza	108
4.38. Diagrama de caja del contenido de ADN en la cabeza	109
4.39. Diagrama de caja de la intensidad promedio en la cabeza	109
4.40. Diagrama de caja del porcentaje de ADN en la cabeza	109
4.41. Diagrama de caja del área de la cola	109
4.42. Diagrama de caja del largo de la cola	110
4.43. Diagrama de caja del contenido de ADN en la cola	110
4.44. Diagrama de caja del la intensidad promedio en la cola	110
4.45. Diagrama de caja del porcentaje de ADN en la cola	110
4.46. Diagrama de caja del momento de cola	111
4.47. Diagrama de caja del momento de Olive	111

Índice de tablas

3.1. Arquitectura U-Net usada para el ensamble	57
3.2. Características extraídas por el sistema	70
3.3. Código de colores del módulo de presentación	71
4.1. Desempeño del módulo de segmentación y del módulo de refinamiento .	75
4.2. Comparación por imagen de los resultados del módulo de segmentación y el módulo de refinamiento	75
4.3. Comparación del desempeño del sistema propuesto con el OpenComet y HiComet	86

Los tóxicos ambientales, el cáncer y el envejecimiento, son problemas que impactan directamente en los tejidos y las células de los humanos [1], teniendo como consecuencia la deterioración del Ácido Desoxirribonucleico (ADN). Debido a esto, se han desarrollado técnicas que permitan conocer el daño en el ADN de células individuales, tales como la técnica de reacción en cadena de la polimerasa (PCR), la técnica denominada DNA breakage detection-fluorescence in situ hybridization (DBD-FISH) y la electroforesis en gel de células individuales (SCGE).

La electroforesis en gel de células individuales, comúnmente denominada ensayo cometa, fue introducida por [2], y es usada en diversas aplicaciones, como el biomonitoreo humano, la ecogenotoxicología y en las pruebas de genotoxicidad [3], debido a que es una técnica rápida, sencilla, sensible, fiable y bastante económica para medir el daño del ADN [4]. Esta técnica consiste en suspender en un portaobjetos con gel de agarosa, una pequeña cantidad de células de ADN, para luego ser sometidas a lisis y electroforesis, y finalmente ser teñidas con un tinte fluorescente [5]. Como resultado de esta técnica, las células de ADN toman una apariencia similar a la de un cometa, en donde el núcleo o ADN intacto es conocido como cabeza del cometa, y el ADN con más daño y fragmentado como cola del cometa (ver Figura 1.1).

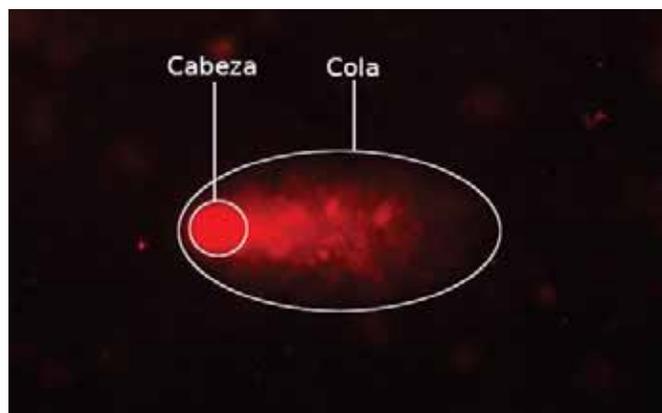


Figura 1.1: Partes que componen una célula procesada por el ensayo cometa

Dentro de los atributos de los cometas, existen algunos que son importantes para la evaluación precisa del daño inducido en el ADN, como, por ejemplo, la determinación de la forma, el tamaño y la cantidad de ADN dentro de los cometas [6]. Por lo tanto, es necesario extraer o calcular parámetros de los cometas obtenidos por el ensayo cometa, ya sea observando la imagen del ensayo cometa a través del microscopio y realizando los cálculos manualmente, o procesando la imagen mediante un sistema computacional capaz de analizar células del ensayo cometa.

1.1. Antecedentes

Con la finalidad de realizar mediciones sobre las imágenes de las células procesadas por el ensayo cometa, se han propuesto varios métodos y sistemas computacionales durante los últimos años. Estos sistemas, tienen el objetivo de segmentar, de forma semi-automática o automática, las regiones de interés de las imágenes del ensayo cometa y extraer automáticamente los parámetros que permitan conocer el grado del daño en el ADN. Dentro de los sistemas propuestos, se distinguen tres categorías, de acuerdo con los métodos usados en el procesamiento del sistema.

- **Sistema semi-automático:** Es necesaria la intervención del usuario en algún punto de su procesamiento.
- **Sistema automático:** No es necesaria la intervención del usuario en el procesamiento.
- **Sistema con aprendizaje automático:** Utiliza modelos de aprendizaje automático en algún punto de su procesamiento.

1.1.1. Sistemas semi-automáticos

Helma & Uhl [7] desarrollaron un sistema semi-automático con una interfaz gráfica amigable para el usuario, en donde se segmentan los cometas por el método de umbralización. Inicialmente, el sistema establece automáticamente un valor de umbral para segmentar los cometas, sin embargo, es posible ajustar el valor de umbral de acuerdo con el criterio del usuario. Para la clasificación de la cabeza, se toma el área que tiene entre el 80 % y 100 % de la intensidad máxima del cometa segmentado. Luego, el usuario selecciona únicamente un cometa para que se calculen los parámetros pertinentes y el sistema abre una ventana nueva en la cual se puede observar el cometa seleccionado junto con sus respectivas mediciones. Si bien, este sistema tiene la desventaja de no ser automático, su punto fuerte radica en darle al usuario cierto control en la segmentación de los cometas. No obstante, el proceso puede resultar bastante tedioso si se quieren analizar varias células y/o imágenes. Además, es posible que un sólo valor de umbral no sea suficiente para segmentar todos los cometas de manera correcta, puesto que las imágenes del ensayo cometa presentan en su mayoría una variación de iluminación.

Otro sistema similar al anterior, es el de Końca et al. [8], quienes presentaron un sistema semi-automático con interfaz gráfica y bastante sencillo de utilizar llamado

CaspLab, que también utiliza el método de umbralización. Cuando se inicia el programa y se carga la imagen, la cual sólo debe contener un cometa, el sistema determina la intensidad máxima de la imagen y la intensidad del nivel de fondo automáticamente, aun así, se le solicita al usuario un valor de umbral que se encuentre entre estos dos valores, para que se pueda segmentar correctamente el cometa. Acto seguido, el sistema procede a clasificar como cabeza el área con los pixeles que estén entre el 80 %-100 % de la intensidad máxima del cometa. Finalmente, si bien el programa calcula los parámetros y los muestra en pantalla, igualmente los exporta en un archivo txt. Este sistema puede resultar más tedioso que [7], puesto que no sólo se tiene definir un valor de umbral para cada imagen, sino que además cada imagen únicamente puede contener un cometa.

Unos años más tarde, Kiziltan & Yurtcu [9] desarrollaron un software de análisis semi-automático para segmentar las partes de un cometa por medio de un algoritmo de mapeo radial, que, además, con el propósito de obtener mediciones más precisas, se le proporciona al usuario la libertad de ajustar estas segmentaciones a su conveniencia. El software cuenta con una interfaz gráfica en la que se le permite al usuario realizar una calibración métrica, y ajustar el valor de umbral de color y las resoluciones angulares/radiales de la imagen a analizar. Luego, el usuario selecciona visualmente los objetos que considera como cometas y mediante el clic del mouse indica los centros de la cabeza y cola de cada uno. Una vez terminada la selección manual, el sistema segmenta automáticamente la cabeza y cola del cometa a través del algoritmo de mapeo radial. Si el usuario lo desea, tiene la posibilidad de ajustar o modificar los bordes de las segmentaciones manualmente usando las herramientas de remodelado que tiene disponible el software. Los parámetros calculados se muestran en pantalla junto con el cometa segmentado. La principal ventaja de este sistema es el control que se tiene para la corrección o el perfeccionamiento de la segmentación de los cometas, el cual es mucho mayor a sus similares, como los desarrollados por [7] y [8]. Sin embargo, aún puede resultar tedioso si se quiere analizar muchas imágenes con múltiples cometas, y sobre todo si se requiere modificar cada una de las segmentaciones.

El último sistema semi-automático, es el propuesto por Lee et al. [10], el cual lleva el nombre de HiComet y hace uso de una segmentación adaptativa para segmentar los cometas. Una de las características principales de este sistema de cuatro pasos, es su capacidad de particionar cometas traslapados. El primer paso, denominado *Preprocessing*, consiste en suavizar la imagen por medio del filtro de la mediana. Luego, en el paso de *Binarization*, se segmentan los objetos por medio de una umbralización, en el cual el valor de umbral se toma del primer valle del histograma de la imagen. El tercer paso, llamado *Filtering and overlap correction*, se encarga de eliminar las regiones que tocan los bordes de la imagen y asignar las regiones muy pequeñas a los cometas más cercanos. Seguidamente, con el fin de particionar las regiones detectadas como cometas traslapados, se aplican una serie de transformadas: la transformada de distancia, la transformada ondícula y la transformada divisoria. Después, se deben corregir las particiones excesivas que se producen por medio de fusiones horizontales. En el último paso, *Characterization and classification*, se calculan los parámetros con base en si el cometa

se cataloga como normal, en necrosis o en apoptosis, y en un parámetro especificado por el usuario, llamado *head threshold* (HT), cuyo valor determina que fracción de la intensidad máxima es utilizada para definir el tamaño de la cabeza. Este sistema es uno de los pocos que es capaz de particionar cometas con traslape, sin embargo, para que sea posible esto, el traslape se debe dar en cometas que sean simétricos con respecto al eje horizontal. Además, los valores de intensidad de la cabeza pueden variar entre los cometas, por lo que elegir un sólo valor de umbral puede ocasionar que algunas cabezas no se detecten completamente o que unas se definan con un tamaño más grande.

1.1.2. Sistemas automáticos

Böcker et al. [11] desarrollaron un sistema de análisis automatizado que se basa en software y hardware donde la intervención humana es casi nula, y la segmentación se hace por medio de una umbralización con histéresis. El sistema establece tres distintas tareas de software, que son ejecutadas en paralelo y distribuidas en tres componentes de hardware diferentes. La primera tarea llamada *Searching and Focusing*, se encarga de mover el portaobjetos del microscopio con una platina de escaneo automática para luego hacer un enfoque automático en busca de objetos brillantes. La siguiente tarea, *Grabbing and Administration*, ajusta automáticamente la ganancia de la cámara, el nivel de negro y el cálculo del sombreado, con el fin de poder adquirir una imagen en escala de grises de los cometas detectados en la tarea anterior. Además, también se encarga de almacenar las imágenes en una pila. La última tarea lleva el nombre de *Recognition and Analysis*, la cual consiste en usar filtros morfológicos para la eliminación del ruido y la corrección de sombreado. Luego se halla un valor de umbral global que no permita el aumento drástico de píxeles considerados como ruido. Este umbral global será utilizado en una umbralización con histéresis [12], obteniendo así una imagen binaria. Finalmente, se obtienen los marcadores, definidos como áreas dentro de los cometas que exceden el 90 % de la intensidad máxima, esto se hace con la finalidad de reconstruir las cabezas de los cometas y proceder a extraer los parámetros que sean necesarios. Este sistema tiene como resultados una sensibilidad de 95,2 % y una especificidad del 92,7 %, sin embargo, es importante señalar, que sólo se analizan cometas intactos, es decir, el sistema rechaza cometas traslapados o anormales.

Una década después, Gyori et al. [13] desarrollan una herramienta automática de análisis de células procesadas por el ensayo cometa, nombrada OpenComet, el cual segmenta los cometas mediante una umbralización adaptativa. Este sistema tiene un algoritmo que se divide en dos partes, una encargada de detectar los cometas denominada *comet finding* y otra que se ocupa de detectar las cabezas de los cometas llamada *Head finding*. El proceso de *comet finding* comienza con un filtrado para eliminar el ruido de la imagen y una corrección global del fondo, para luego efectuar una umbralización adaptativa por medio del método de Huang [14], el cual elige un valor de umbral minimizando una medida de borrosidad. Una vez obtenida la imagen binaria, se eliminan las regiones pequeñas y las regiones que tocan el borde la imagen. Después, el sistema etiqueta como cometas traslapados a las regiones que no cumplan con una relación mínima de convexidad y a las regiones que no sean simétricas con respecto

al eje horizontal de la imagen, con el fin de rechazar estas regiones en las mediciones. Además, con base en estadísticas de cometas atípicos, se etiquetan a las regiones con una altura o área irregular como cometas atípicos. En el proceso de *head finding*, se establece un método para cometas levemente dañados, que consiste en clasificar como cabeza a los píxeles con mayor intensidad, mientras que para los cometas gravemente dañados se analiza una señal aleatoria del perfil de intensidad del cometa para localizar el borde entre la cabeza y la cola del cometa. Finalmente, se extraen los parámetros de los cometas segmentados y etiquetados como normales y atípicos. Los resultados de este sistema alcanzaron una sensibilidad del 97.6 % y una especificidad del 92.7 %, por lo mismo, es uno de los métodos que más se usan para comparar un nuevo sistema propuesto. Aun así, el área que se designa como cabeza del cometa es mucho mayor a la real, y como consecuencia, los parámetros calculados pueden resultar erróneos. Adicionalmente, se observa que el sistema siempre define la cabeza en el extremo izquierdo, por lo que, si el cometa tiene un cabeza justo en el centro, el área de la cabeza tomará parte de la cola.

Por último, Bocanegra et al. [15] propusieron un sistema automático llamado CometLab, para segmentar cometas por medio de filtros morfológicos y umbralización, que además puede particionar algunos de los cometas que tengan ligeros traslapes. Si bien el sistema segmenta automáticamente, igual tiene la flexibilidad de remover cometas detectados si el usuario así lo considera. En el algoritmo, se establecen cuatro pasos a seguir. El primero de ellos es pre-procesar la imagen haciendo una ecualización de histograma para revelar cometas ocultos, y generar dos imágenes binarias, una llamada cometas subestimados (detección de cometas muy estricto) y otra llamada cometas sobreestimados (detección de cometas permisivo). Para generar los cometas subestimados, se aplicó un filtro de sobel, un filtro de la mediana y una operación closing, luego de eso se remueven las áreas pequeñas y se rellenan algunas regiones. En cambio, para generar los cometas sobreestimados, se aplica un filtro de entropía y se procede a remover las áreas pequeñas para luego terminar con una operación de closing. Después, se lleva a cabo la etapa de segmentación, que consiste en asignar, por medio de la intersección, a cada cometa de la imagen con cometas subestimados un cometa de la imagen con cometas sobreestimados. De esta manera, se separan cometas que estaban muy cercanos o cometas que tenían un pequeño traslape y además se eliminan falsos positivos que se habían detectado en la imagen con cometas sobreestimados. Para finalizar este paso, se aplica una operación morfológica de engrosamiento a la imagen interceptada. En el tercer paso, se realiza la segmentación de cabezas, obteniendo una señal de valores medios de intensidad por cada cometa y localizando el mayor cambio de una intensidad alta a una intensidad baja. En el último paso, se extraen los parámetros de los cometas segmentados automáticamente, sin embargo, el usuario puede rechazar algún cometa de estos cálculos. Si bien, uno de los puntos fuertes del sistema CometLab, es que es capaz de particionar cometas traslapados, el método de traslape es inferior al propuesto en [10], puesto que CometLab sólo particiona los cometas cuando su traslape es muy ligero.

1.1.3. Sistemas con aprendizaje automático

Sansone et al. [16] propusieron un sistema con un algoritmo que se subdivide en dos partes, una para detectar los cometas intactos y otra para segmentar cada una de sus partes, que, a diferencia de los demás sistemas, este sistema identifica tres partes del cometa (cabeza, cola y halo) a través de un modelo de aprendizaje automático no supervisado, llamado fuzzy c-means. Para detectar los cometas, inicialmente se calcula el gradiente de la imagen y se filtra con un filtro gaussiano; seguido de eso, se umbraliza la imagen con un valor de $1/5$ de la intensidad máxima con el fin de obtener una imagen binaria. Después, mediante operadores morfológicos, se rechazan cometas con una intensidad baja, cometas pequeños, cometas traslapados y cometas con núcleos anormales. Con respecto a la segunda parte del proceso, se segmentan la cabeza, cola y halo mediante el algoritmo de agrupación difusa fuzzy c-means [17]. Los resultados de la detección de cometas indicaron una buena sensibilidad, puesto que el porcentaje de verdaderos positivos fue superior al 93 %, mientras que el porcentaje de falsos negativos fue inferior al 7 %, a pesar de esto, el sistema solo detecta cometas intactos y rechaza los traslapados.

Pocos años después, Ganapathy et al. [18] implementaron una herramienta totalmente automatizada, denominada CometQ, que usa un modelo de aprendizaje supervisado llamado SVM, y un modelo de aprendizaje no supervisado denominado fuzzy c-means. En este sistema se emplean cuatro diferentes etapas: clasificación, segmentación, partición y cuantificación. En la primera etapa, la imagen se clasifica, por medio de clasificadores SVM, el tipo tinción, y nuevamente se clasifica dependiendo del grado de daño presentado en los cometas de la imagen. Por tanto, una imagen puede ser clasificada en una de las cuatro categorías, en donde se tiene un método de segmentación diferente para cada una. *Método 1.* El primer método de segmentación está dirigido a las imágenes con tinción argéntica que contienen cometas con leve o moderado daño, y consiste en remover el efecto de sombreado y hacer un filtrado homomórfico para mejorar el contraste de la imagen [19], luego se umbraliza la imagen por el método de Otsu [20] y se finaliza con la eliminación de ruido por medio de operaciones morfológicas. *Método 2.* Para segmentar las imágenes con tinción argéntica que contienen cometas muy dañados, se remueve el efecto de sombreado y se realiza el método de sustitución de fondo [21], para terminar con una umbralización por el método DRLSE [22]. *Método 3.* Con relación a las imágenes con tinción fluorescente que contienen cometas con leve o moderado daño, se remueve el efecto de sombreado y se efectúa el método de sustitución de fondo, seguido de eso, se umbraliza a través del método de Otsu y se acaba con la eliminación de ruido por medio de operaciones morfológicas. *Método 4.* En el caso de las imágenes con tinción fluorescente que contienen cometas muy dañados, el método consiste en remover el efecto de sombreado y mejorar el contraste con el método MAGCWD [23], después se segmenta con el algoritmo fuzzy c-means y se elimina el ruido con operadores morfológicos. Es importante señalar, que después de cada método de segmentación se eliminan las regiones que tocan el borde la imagen y las regiones que se consideren como cometas traslapados. En la etapa de partición, se recorta la imagen segmentada en recuadros que contienen solo un cometa, se corrige su orientación y se

clasifican las partes de cabeza, cola y halo con el algoritmo fuzzy *c*-means. Luego de esto, para tener regiones mejor definidas, se emplean operaciones morfológicas para cada clase. Finalmente, en la etapa de cuantificación, la región perteneciente al halo se asigna a la región de cola, y se calculan los parámetros que se desean conocer. La selección automática del método de segmentación, de acuerdo con las características de la imagen, conllevan a buenos resultados para distinguir los cometas del fondo. Pese a esto, la clasificación de las partes del cometa por medio del algoritmo de agrupamiento difuso, no es del todo buena, puesto que los bordes de las regiones de cada clase tienden a ser muy irregulares.

Recientemente, Hong et al. [24] han propuesto un sistema de aprendizaje profundo para la detección, clasificación y extracción de parámetros de células procesadas por el ensayo cometa llamado DeepComet. Este sistema usa una red neuronal convolucional denominada Mask R-CNN para realizar la detección de los cometas. Para entrenar el modelo de aprendizaje profundo, se utilizaron 837 imágenes que contienen alrededor de 6,690 cometas. Por cada imagen se hizo un enmascarado manual en el que se delimita el área del cometa y se colocan dos puntos críticos para indicar el centro de la cabeza y el límite entre la cabeza y la cola del cometa. Además, cada cometa se etiqueta como fantasma, en caso de que el cometa tenga una cabeza pequeña y una cola ancha, y como no fantasma, cuando se distingue perfectamente el área perteneciente a la cola y a la cabeza. Por último, se tiene un etiquetado más, donde se señala el estado del cometa, es decir, si el cometa está traslapado, intacto o parcial como consecuencia de estar en el borde de la imagen. Una vez entrenado el modelo Mask R-CNN, ya es posible obtener automáticamente una imagen enmascarada donde se identifiquen los cometas y se obtengan las etiquetas anteriormente descritas. Como en la mayoría de las imágenes del ensayo cometa la intensidad del fondo se distribuye de manera irregular, el sistema extrae cada cometa detectado en un cuadro delimitador y se le resta a la intensidad de brillo del cometa la intensidad del color del fondo local. Luego, se corrige la orientación del cometa en caso de que éste no esté simétrico horizontalmente, y se separa la cabeza y cola de acuerdo con la predicción de la ubicación de los puntos críticos. Finalmente, el sistema calcula los parámetros de los cometas e indica si se clasificó como un cometa fantasma o como un cometa no fantasma. Los autores de este sistema, comparan los parámetros extraídos con los parámetros extraídos de los sistemas Comet Assay IV, OpenComet [13] y HiComet [10], e indican que su sistema propuesto presenta una mejor exactitud y está más cerca del Ground truth. No obstante, se observa en sus imágenes que no existen una variedad de cometas, y en su mayoría la cabeza del cometa siempre se presenta en el extremo izquierdo.

1.2. Contribuciones

- La principal contribución del presente trabajo es el desarrollo de un sistema completamente automático basado en un ensamble de redes neuronales convolucionales U-Net para segmentar células individuales procesadas por electroforesis en gel y poder extraer sus características.

- De igual manera, se realizan una serie de algoritmos, que tienen como objetivo mejorar el desempeño de la segmentación realizada por el ensamble de redes neuronales convolucionales U-Net.
- Finalmente, se implementa un algoritmo computacional para particionar los cometas traslapados, con el fin de obtener los parámetros para cada cometa de manera individual.

1.3. Estructura de la tesis

En esta sección, se describe brevemente el contenido de cada capítulo que integra este trabajo.

Capítulo 1.- Introducción. Se menciona el contexto del problema que se quiere solucionar, y además se describen los sistemas computacionales a fines al problema que fueron desarrollados en años anteriores al presente trabajo.

Capítulo 2.- Marco teórico. Se exponen los conceptos, algoritmos y métodos computacionales usados en el desarrollo del presente proyecto.

Capítulo 3.- Metodología. Se describe la base de datos, la arquitectura y la forma de entrenamiento del ensamble de redes neuronales convolucionales, y además se explican las técnicas utilizadas en el sistema desarrollado.

Capítulo 4.- Resultados. Se muestran los resultados de la segmentación y la extracción de características del sistema desarrollado, y adicionalmente, se comparan los resultados con otros sistemas del estado del arte.

Capítulo 5.- Conclusiones. Se presenta un análisis acerca del desempeño del sistema desarrollado, y las potenciales mejoras que se pueden implementar en un futuro.

1.4. Publicaciones

En este capítulo se definirán y explicarán los conceptos, métodos y técnicas usadas en el desarrollo del presente trabajo de tesis. En un principio, se explicará en que consiste el ensayo cometa y se expondrá la definición y el objetivo del problema segmentación. Posteriormente se detallarán las partes que constituyen una red neuronal convolucional enfocada a este problema. De último, se describen los algoritmos morfológicos que fueron de ayuda en el post-procesamiento.

2.1. Ensayo cometa

En [1] se describe al ensayo cometa como una técnica de electroforesis en microgel desarrollada por [2]. Singh et al., mencionan que para realizar esta técnica, las células, que comúnmente se tiñen de un color fluorescente, se deben incrustar en un portaobjetos con gel de agarosa, y posteriormente son lisadas con detergentes y sal. Luego, las células se someten a electroforesis durante un período corto en condiciones neutrales, de modo que, al aplicar la diferencia de potencial, los fragmentos con mayor daño genético serán mayormente desplazados desde el núcleo hasta el ánodo o polo positivo, formando la apariencia de cometa (ver Figura 2.1).

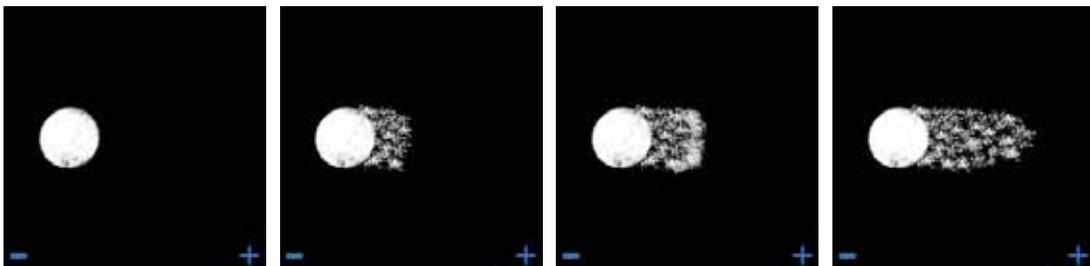


Figura 2.1: Representación de la migración de ADN

En el resultado del ensayo cometa, se puede distinguir dos elementos importantes que constituyen un cometa. El primero, es el núcleo o cabeza del cometa, el cual consiste en los fragmentos con menor o ningún daño y que no fueron desplazados por la electroforesis hacia el ánodo. El otro elemento, es denominado cola del cometa, y está

2. MARCO TEÓRICO

compuesto por los fragmentos de ADN con mayor daño y que fueron desplazados desde el núcleo hacia el ánodo.

Es importante resaltar que no todas las células de ADN presentan el mismo daño genético, en consecuencia, los cometas pueden tomar diferentes formas y tamaños dependiendo del daño que presenten las células de ADN (ver Figura 2.2).

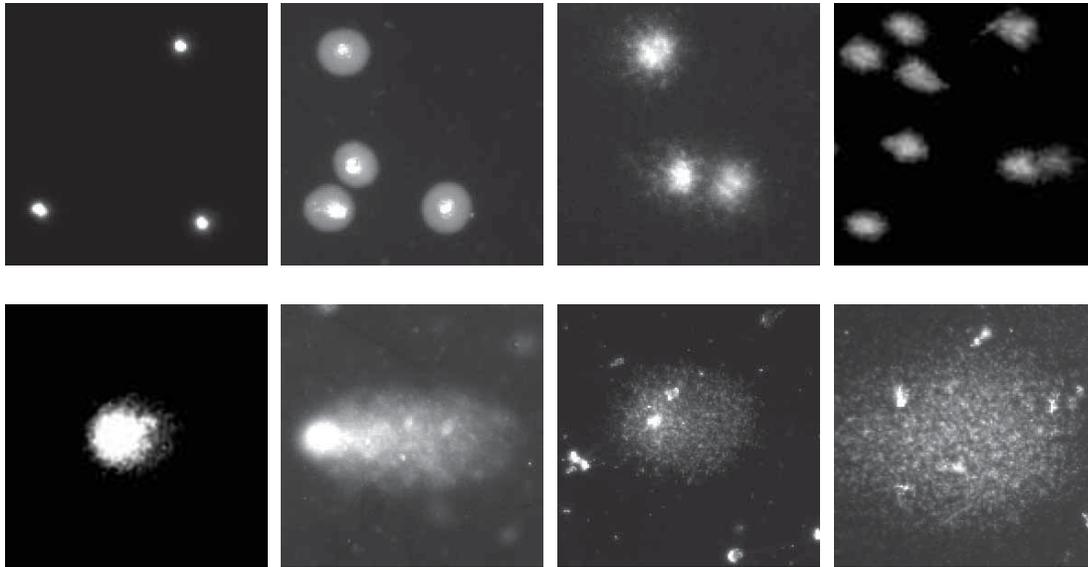
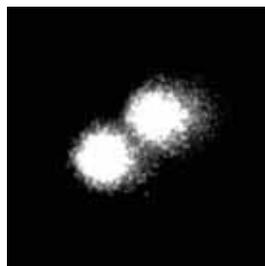
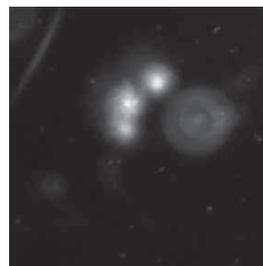


Figura 2.2: Muestras del ensayo cometa

Algunos defectos que pueden ocasionar problemas para el análisis de las células del ensayo cometa son: el traslape de cometas y la aparición de partículas que no son cometas. Con respecto al traslape de cometas, este se da cuando el ADN que está migrando al ánodo colisiona con otro cometa, provocando que se mezcle el contenido genético de cada cometa (ver Figura 2.3a). En cambio, la aparición de partículas que no son cometas, se da por una ligera contaminación ya sea en la obtención de la muestra de las células de ADN o en la instrumentación utilizada (ver Figura 2.3b).



(a) Cometas traslapados



(b) Partículas diferentes a cometas

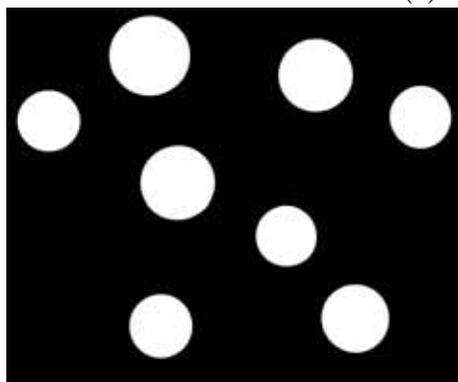
Figura 2.3: Defectos en muestras del ensayo cometa

2.2. Segmentación

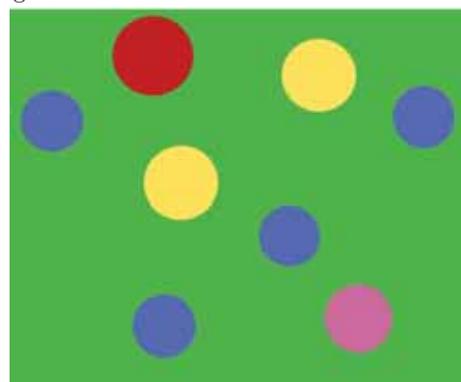
La segmentación es un método que divide una imagen en sus partes u objetos constituyentes [25], con el fin de distinguir los objetos de interés que contiene la imagen y separarlos del fondo y/o de otros objetos. La segmentación puede ser binaria (ver Figura 2.4b) o multiclase (ver Figura 2.4c). En el primer tipo de segmentación, únicamente se separa una clase de objeto del fondo. En el caso de la segmentación multiclase, se separan varias clases de objetos del fondo, y se distinguen los límites de cada objeto en caso de que se traslapen. Para realizar este método existen diversas técnicas como la umbralización, region growing y la técnica de división y fusión. Si bien, la mayoría de estas técnicas requieren que se determine un valor de umbral u homogeneidad de manera manual, se han propuesto algunas técnicas adicionales para hallar este valor de forma automática. No obstante, existen modelos matemáticos de aprendizaje profundo, denominados redes neuronales convolucionales, que permiten la segmentación binaria y multiclase de manera automática.



(a) Imagen original



(b) Segmentación binaria



(c) Segmentación multiclase

Figura 2.4: Tipos de segmentación

2.2.1. Segmentación binaria

La segmentación binaria permite la identificación de una clase de objeto en una imagen. En este tipo de segmentación, normalmente, se identifica un píxel del objeto con un valor de 1, y un píxel del fondo con un valor de 0 (ver Figura 2.5).

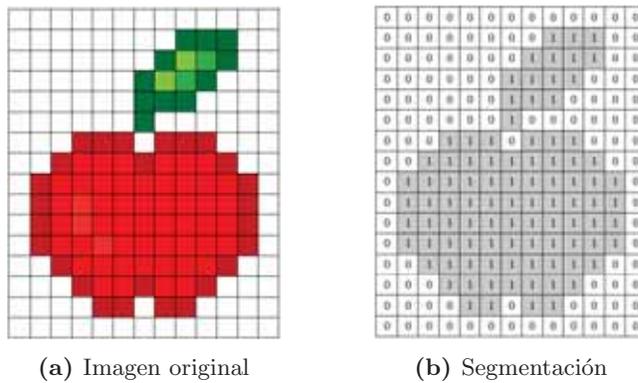


Figura 2.5: Segmentación binaria

2.2.2. Segmentación multiclase

En la segmentación multiclase se distinguen varias clases de objetos en una imagen. Para este tipo de segmentación, se puede identificar cada píxel de una clase de objeto con un número entero (ver Figura 2.6), o bien, usar la codificación one-hot (ver Figura 2.7), en donde el resultado de la segmentación tiene el mismo número de canales de profundidad que el número de clases, por lo tanto, cada canal está asociado a una clase, y en él se identifica con un 1 si el píxel pertenece a la clase asociada y un 0 si no pertenece a esa clase.

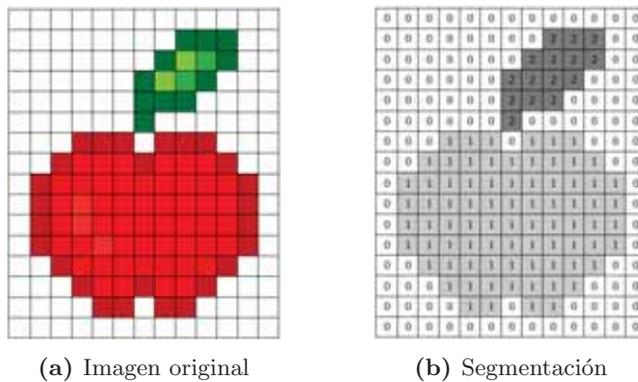


Figura 2.6: Segmentación multiclase

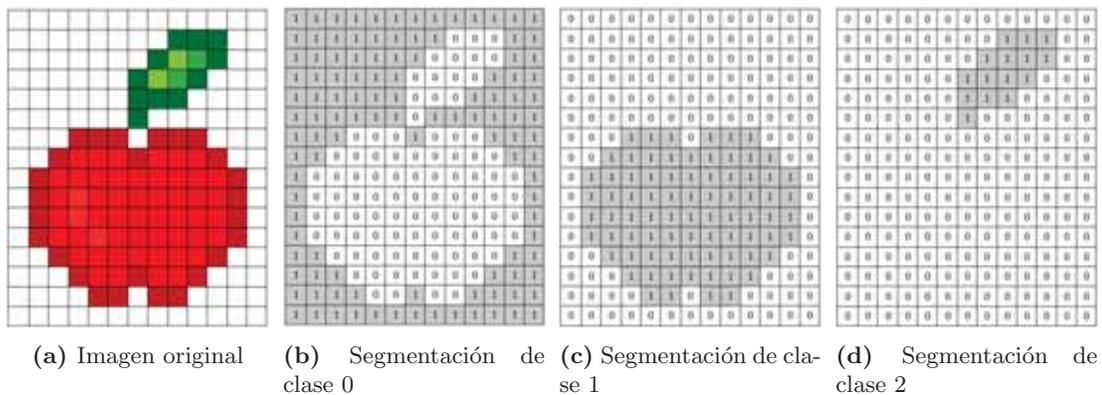


Figura 2.7: Codificación one-hot

2.2.3. Segmentación por transformación divisoria

La transformación divisoria o watershed transform es un método de segmentación, introducido por [26], dirigido a objetos que se traslapan. En [27] se presenta a la transformación divisoria como un método que es capaz de visualizar una imagen como un paisaje topográfico (ver Figura 2.8b), con agujeros perforados en los mínimos locales, y que se está sumergiendo en un lago. Inicialmente, las cuencas del paisaje se llenarán de agua, hasta el punto de que el agua proveniente de diferentes cuencas se encuentra, y se construye ahí una presa. Al finalizar el proceso, el paisaje estará dividido en regiones o cuencas separadas por presas (ver Figura 2.8c).

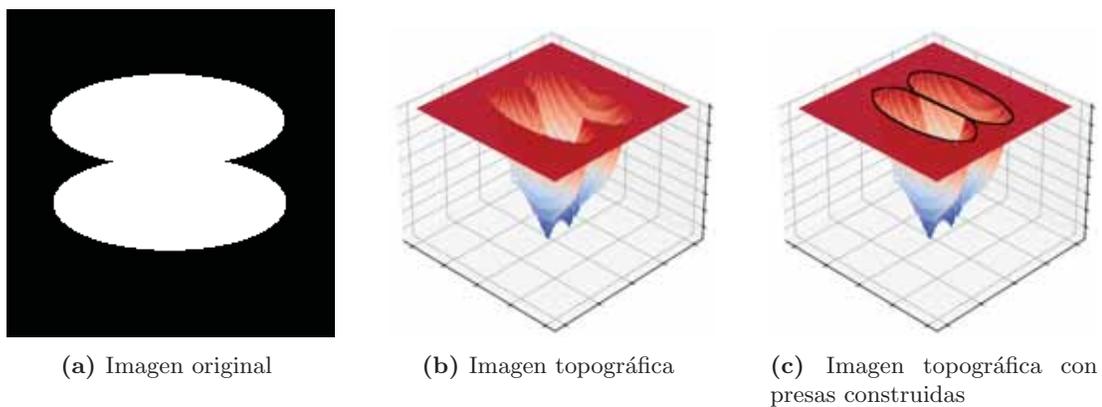


Figura 2.8: Funcionamiento general de la transformación divisoria

Para realizar el algoritmo de watershed de una imagen binaria (ver Figura 2.9a), primero se halla su transformada de distancia. El resultado de lo anterior, se denomina mapa de distancias y se multiplica por -1 , con el fin de obtener la imagen con cuencas (ver Figura 2.9b). Luego, se determinan los marcadores, ya sea de manera manual, o de forma automática, hallando los máximos locales del mapa de distancias original (ver

Figura 2.9c). Finalmente, los píxeles del mapa de distancia se asignan al marcador más cercano, obteniendo así, la imagen segmentada (ver Figura 2.9d).

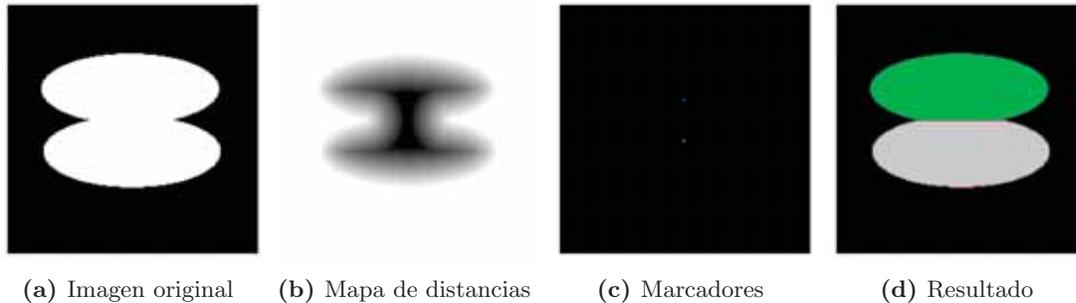


Figura 2.9: Algoritmo de la transformación divisoria

2.2.4. Evaluación de la segmentación

Independientemente de la técnica o el modelo matemático que se use para segmentar una o varias clases de objetos de una imagen, es común evaluar su desempeño con respecto a los resultados esperados. Los resultados esperados, consisten en segmentar manualmente la imagen o las imágenes que se procesaron con la técnica o el modelo. A estas imágenes que se segmentan manualmente con el fin de evaluar el desempeño de la técnica o el modelo, se denominan Ground Truth. En la Figura 2.10, se muestra los resultados esperados en una segmentación de calculadoras.

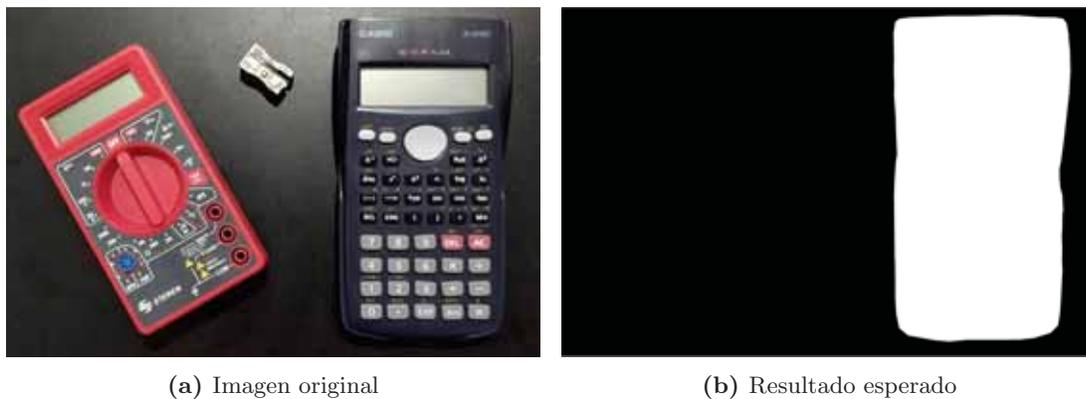


Figura 2.10: Ejemplo de ground truth

2.2.4.1. Matriz de confusión

Las métricas usadas para evaluar el desempeño de una segmentación, compara cada uno de los píxeles contenidos en la imagen segmentada con su respectivo píxel en el Ground Truth. Estos píxeles comparados pueden agruparse en 4 grupos, los cuales

están determinados por la matriz de confusión (ver Figura 2.11), y se describen a continuación:

		Valor del Ground Truth	
		Positivo (1)	Negativo (0)
Valor clasificado	Positivo (1)	Verdadero positivo (TP)	Falso positivo (FP)
	Negativo (0)	Falso negativo (FN)	Verdadero negativo (TN)

Figura 2.11: Matriz de confusión

- **Verdadero positivo (TP):** Son los píxeles que fueron clasificados correctamente como clase del objeto.
- **Verdadero negativo (TN):** Son los píxeles que fueron clasificados correctamente como no perteneciente a la clase del objeto.
- **Falso positivo (FP):** Son los píxeles que fueron clasificados como clase del objeto, pero realmente no pertenecen al objeto.
- **Falso negativo (FN):** Son los píxeles que fueron clasificados como no perteneciente a la clase del objeto, pero en realidad si pertenecían.

2.2.4.2. Métricas de desempeño

En esta sección, se describen algunas de las métricas usadas para evaluar el desempeño de las segmentaciones.

- **Exactitud (Accuracy):** La métrica de exactitud mide la proporción de predicciones correctas sobre el número total de instancias evaluadas [28]. A pesar de ser una de las métricas más usadas, es posible que la métrica favorezca a la clase mayoritaria, en un conjunto de datos extremadamente no balanceado.

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.1)$$

- **Precisión (Precision):** La precisión se utiliza para medir la fracción de verdaderos positivos que se predicen correctamente a partir de todos los elementos positivos predichos [28], en otras palabras, indica que proporción de todos los píxeles clasificados como clase de objeto, realmente si pertenecen a esa clase.

$$P = \frac{TP}{TP + FP} \quad (2.2)$$

- **Exhaustividad (Recall):** La exhaustividad se utiliza para medir la fracción de verdaderos positivos que se clasifican correctamente [28], es decir, que proporción de todos los píxeles de la clase del objeto fueron clasificados correctamente.

$$R = \frac{TP}{TP + FN} \quad (2.3)$$

- **F1 score:** El F1-Score es la media armónica de la precisión y exhaustividad. Esta métrica es la más favorable para problemas de clasificación con clases raras [29], o lo que es, un conjunto de datos con clases extremadamente no balanceadas.

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (2.4)$$

- **F1 score Average:** Esta métrica es usada para evaluar segmentación multiclase, la cual únicamente consiste en el promedio del F1 score de cada clase.

$$F1_{avg} = \frac{1}{C} \cdot \sum_{k=1}^C F1_k \quad (2.5)$$

2.3. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN) son un modelo matemático inspirado en el mecanismo de percepción visual natural de los seres vivos [30], los cuales tienen la tarea de resolver problemas de aprendizaje automático, por medio de funciones de activación y capas convolucionales con kernels capaces de extraer patrones. Estos modelos han tenido resultados innovadores en una variedad de campos relacionados con el reconocimiento de patrones, como el de procesamiento de imágenes [31]. Para que una red neuronal convolucional aprenda a reconocer estos patrones, es necesario entrenar el modelo enseñándole datos históricos con su esperada salida, y determinar una función de costo que establezca una trayectoria para llegar al objetivo deseado, el cual es predecir correctamente un resultado dada una entrada.

Algunas de las aplicaciones de las redes neuronales convolucionales en el campo de procesamiento de imágenes, son la detección (ver Figura 2.12b), la segmentación (ver Figura 2.12c) y la clasificación (ver Figura 2.12d).

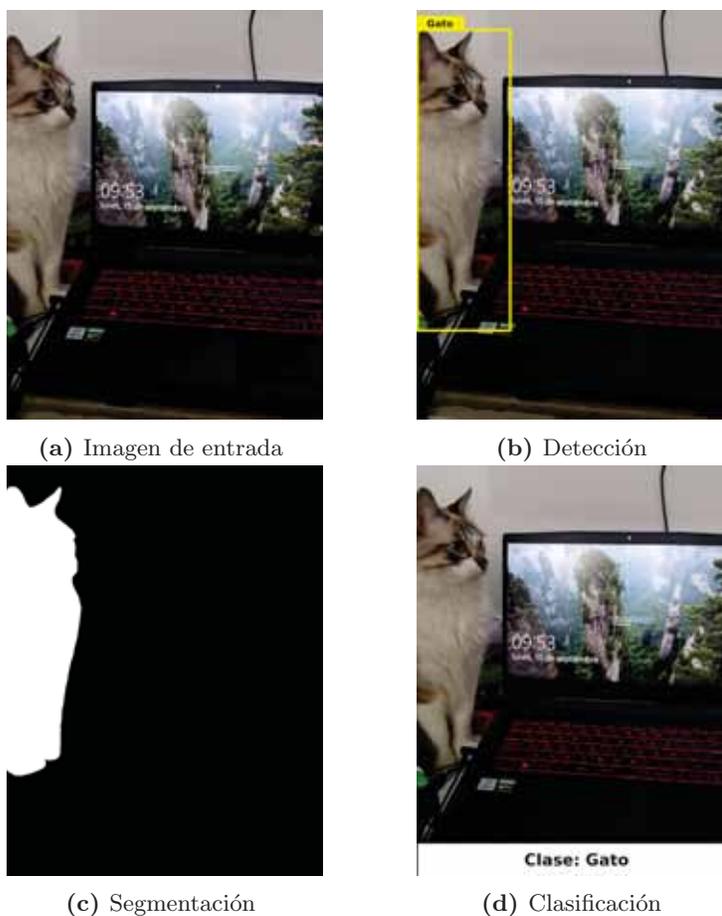


Figura 2.12: Aplicaciones de las redes neuronales convolucionales

2.3.1. Kernel

El kernel es una matriz cuyo tamaño define una vecindad para realizar una operación y cuyos coeficientes o pesos determinan la naturaleza del filtro [25]. Estas matrices, también denominadas filtros, son usadas en operaciones de convolución o filtrados espaciales. En la Figura 2.13, se ilustra un ejemplo de un kernel de 3×3 con coeficientes genéricos.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Figura 2.13: Kernel de 3×3 con pesos genéricos

En el campo de procesamiento de imágenes, existen filtros con coeficientes que mediante filtrados espaciales, permiten el suavizado de imágenes, como el kernel de promedio (ver Figura 2.14a), o la detección de bordes, como el kernel de sobel [32] (ver Figura 2.14b) o el kernel de prewitt [33] (ver Figura 2.14c).

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

(a) Kernel de promedio

-1	0	1
-2	0	2
-1	0	2

(b) Kernel de Sobel

1	0	-1
1	0	-1
1	0	-1

(c) Kernel Prewitt

Figura 2.14: Ejemplos de kernel usados en filtrados espaciales

Por otra parte, en el campo de aprendizaje automático, los kernel son usados en las capas convolucionales de las redes neuronales convolucionales. En estos modelos, los valores de los pesos del kernel, son determinados por el aprendizaje del modelo, con el fin de reconocer patrones. No obstante, los pesos de los kernel en una capa convolucional, pueden definirse de manera inicial a través de inicializadores, como el Glorot Uniform [34]. En esta inicialización, los valores de los pesos W , son extraídos de manera aleatoria dentro de los límites de 2.6, donde n_j es el número de coeficientes de los kernel de entrada y n_{j+1} es el número de coeficientes de los kernel de salida.

$$W \sim U \left[-\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}} \right] \quad (2.6)$$

2.3.2. Convolución

La convolución de una imagen I de tamaño $M \times N$ con un kernel w de tamaño $m \times n$ viene dado por 2.7, donde x y y varían de modo que el centro (origen) del kernel w visita cada píxel de I una vez [25].

$$g(x, y) = (w * I)[x, y] = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (2.7)$$

En el campo de aprendizaje profundo, se conoce al resultado de la convolución como mapa de características. En la Figura 2.15, se puede observar el proceso que se lleva a cabo al realizar una convolución. El valor resultante para cada píxel del mapa de características es una combinación lineal de los pesos del kernel y los valores de los píxeles de entrada. Además, se observa que el tamaño de la imagen resultante disminuye.

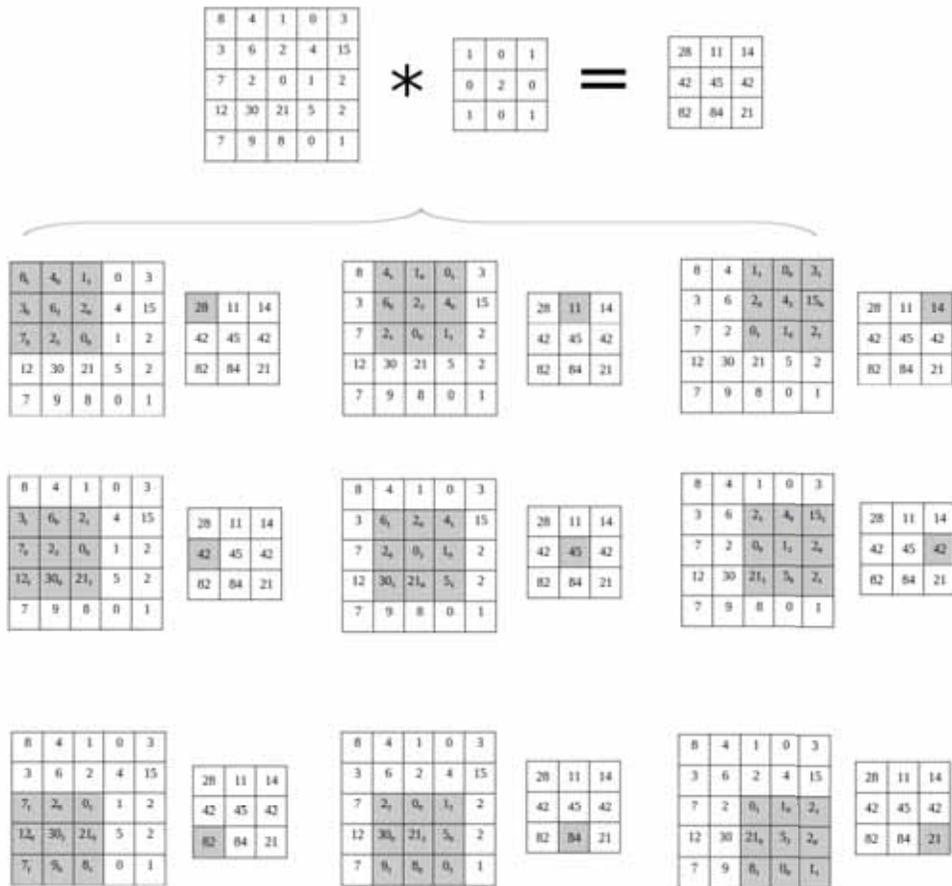


Figura 2.15: Proceso de convolución

En las capas convolucionales de las redes neuronales convolucionales, comúnmente se añade al resultado de la convolución, un vector denominado bias (ver 2.8). El bias, tiene el objetivo de mover fuera del origen la combinación lineal producida por la convolución.

$$g(x, y) = (w * I)[x, y] + b \tag{2.8}$$

Adicionalmente, un método usado para que la salida de la convolución sea del mismo tamaño que la entrada, es el relleno de ceros. Este método consiste en aumentar el tamaño de la imagen de entrada agregando ceros en los bordes, de modo que, al realizar la convolución, el tamaño de la salida sea exactamente igual al tamaño de entrada. Un ejemplo de una convolución con relleno de ceros y uso del bias se muestra en la Figura 2.16.

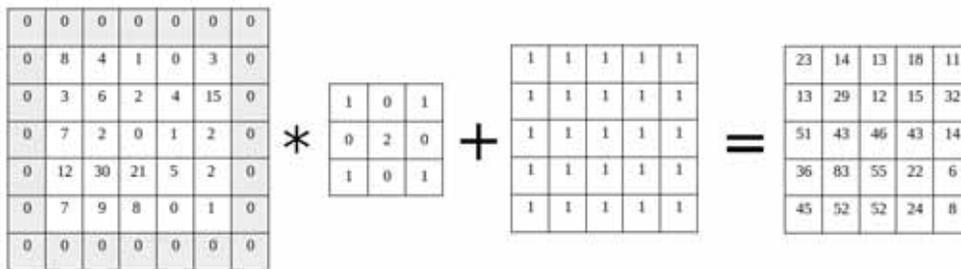


Figura 2.16: Convolución con relleno de ceros y bias

2.3.3. Convolución transpuesta

La convolución transpuesta es una operación matemática, que, a diferencia de las convoluciones, que conectan múltiples entradas dentro de un kernel a un solo resultado, estas asocian una sola entrada con múltiples salidas [35]. Estas operaciones, normalmente son usadas para aumentar el tamaño de los mapas de características obtenidos al realizar una convolución. En la Figura 2.17, se muestra un ejemplo del proceso de una convolución transpuesta, donde se hace un corrimiento de kernel de 2 píxeles.

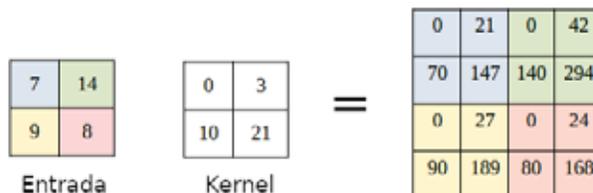


Figura 2.17: Convolución transpuesta

2.3.4. Submuestreo

El submuestreo o pooling se realiza subdividiendo un mapa de características en un conjunto de regiones pequeñas (normalmente 2×2), llamadas vecindades de submuestreo, y reemplazando todos los elementos en dicha vecindad por un solo valor [25]. Algunos tipos de submuestreos utilizados en las redes neuronales convolucionales, es el max pooling, el cual reemplaza los valores de la vecindad por su máximo valor (ver Figura 2.18b); y el average pooling, que reemplaza los valores de la vecindad por su promedio (ver Figura 2.18c). Estas operaciones regularmente se usan para reducir el tamaño de los mapas de características de una red neuronal convolucional y así poder calcular convoluciones con más números de filtros en menor tiempo de procesamiento.

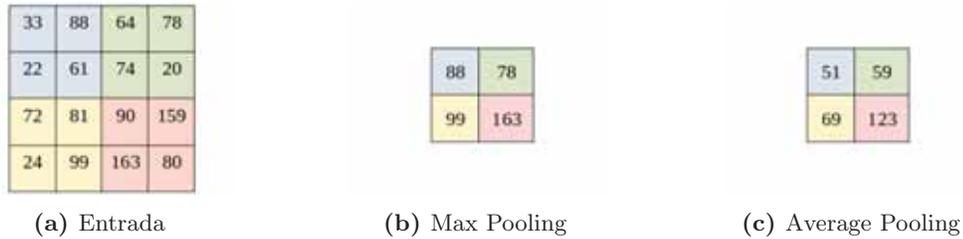


Figura 2.18: Ejemplo de submuestreo de 2×2

2.3.5. Sobremuestreo

El sobremuestreo o unpooling, es una operación contraria al submuestreo, la cual se realiza multiplicando un valor del mapa de características por cada uno de los valores contenidos en una matriz de tamaño n (normalmente 2×2). Un tipo de sobremuestreo utilizado en las redes neuronales convolucionales, es el sobremuestreo con el vecino más cercano, el cual consiste en multiplicar el valor tomado del mapa de características por una matriz de unos (ver Figura 2.19b). Otro tipo de sobremuestreo, es el bilineal, en el que se multiplica el valor tomado del mapa de características por una matriz con valores uniformemente espaciados, que van desde 1 a un factor de escalamiento seleccionado (ver Figura 2.19c).

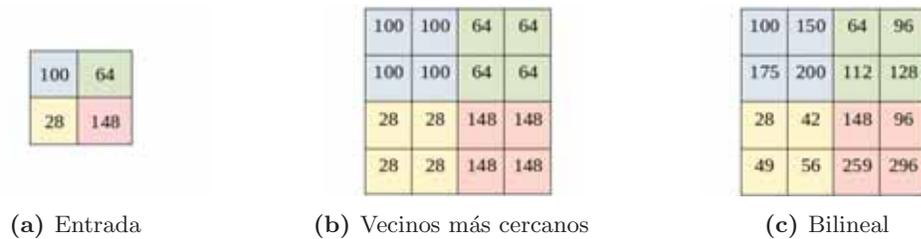


Figura 2.19: Ejemplo de sobremuestreo de 2×2

2.3.6. Función de activación

La función de activación, es una relación que asocia un elemento del mapa de características a un valor de salida definido por la función de activación usada. En 2.9, se describe de manera general una función de activación.

$$f_A(z) = f_A(g(x, y)) = f_A((w * I)[x, y] + b) \quad (2.9)$$

2.3.6.1. ReLU

Una de las funciones de activación más usadas en las capas convolucionales, es la Rectified linear unit (ReLU) [36]. ReLU es una función lineal por partes que reduce la parte negativa a cero y retiene la parte positiva [30]. La función definida por 2.10, se muestra gráficamente en la Figura 2.20.

$$R(z) = \max(0, z) \quad (2.10)$$

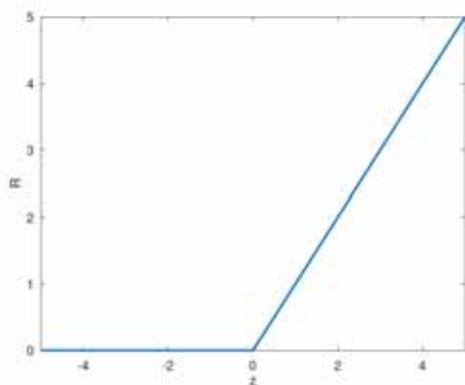


Figura 2.20: Función de activación ReLU

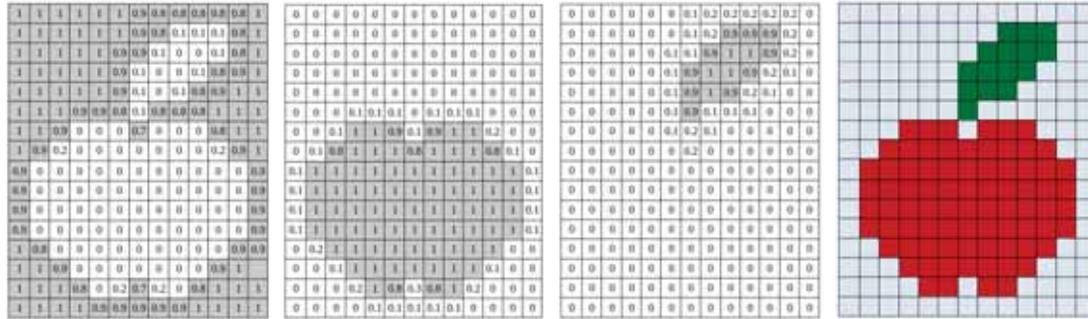
2.3.6.2. Softmax

Una función de activación, habitualmente usada en la última capa de convolución, cuyo mapa de activación es la predicción o resultado final del modelo, es la softmax (ver 2.11), la cual satisface $0 \leq \sigma(z)_j \leq 1$ y $\sum_j \sigma(z)_j = 1$ [37].

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{para } j = 1, 2, \dots, k \quad (2.11)$$

En problemas de clasificación o segmentación, los valores de salida de la softmax representan la probabilidad de pertenecer a una clase. Por lo tanto, la salida de la softmax es un vector o una matriz de probabilidades. La Figura 2.21 muestra un ejemplo

de lo que sería una predicción determinada por la softmax, para un modelo de segmentación de manzanas y sus tallos. Adicionalmente, en la Figura 2.21d, se muestra como quedaría la segmentación, si se clasificara de acuerdo con la probabilidad más alta.



(a) Predicción para clase fondo (b) Predicción para clase manzana (c) Predicción para clase tallo (d) Segmentación

Figura 2.21: Ejemplo de una salida de la Softmax

2.3.7. Función de costo

La función de costo o pérdida, sirve como una heurística para el aprendizaje de las redes neuronales convolucionales, es decir, determina que tan bien o mal están las predicciones realizadas por el modelo de acuerdo con los datos históricos y dicta el camino a seguir para mejorar estas predicciones.

La función de costo cross-entropy es usada para la clasificación binaria. Esta función se muestra en 2.12, donde y es el valor real, \hat{y} es el valor predicho y m es el número de salidas. En ella, cada probabilidad predicha se compara con el valor real de salida de la clase (0 o 1) y se calcula una puntuación que penaliza la probabilidad en función de la distancia al valor esperado [38]. Puntuaciones cercanas a cero indican una buena predicción, mientras que puntuaciones más altas, significan un mal desempeño.

$$L = -\frac{1}{m} \sum_{i=1}^m y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log(1 - \hat{y}_i) \tag{2.12}$$

De igual manera, la función de costo cross-entropy puede ser usada para clasificación multiclase (ver 2.13) Para poder calcular la función cross-entropy, los valores reales de las clases deben estar codificados en one-hot-vector, lo que significa que hay una característica binaria para cada valor de clase y las predicciones deben tener probabilidades predichas para cada una de las clases [38].

$$L = -\sum_{i=1}^m y_i \cdot \log \hat{y}_i \tag{2.13}$$

2.3.8. Descenso de gradiente

El aprendizaje de una red neuronal convolucional, depende de una constante actualización de sus pesos, que al evaluarse en la función de costo, se consigan valores de costo cada vez más bajos. Un método usado para realizar este objetivo, es el descenso de gradiente. Este método se centra en optimizar el modelo matemático y la función de costo, lo que significa modificar los pesos de las capas convolucionales para reducir el valor de la función de costo [39]. Por lo tanto, sea $J(\theta)$ una función de costo, la actualización del vector de pesos θ de la red está dada por 2.14, donde α es la tasa de aprendizaje, y su valor indica que tan lento o rápido se realiza el descenso de gradiente.

$$\theta_{nueva} = \theta_{previa} - \alpha \frac{\partial}{\partial \theta_i} J(\theta) \quad (2.14)$$

Se muestra, el algoritmo general de descenso de gradiente en 2.1.

Algoritmo 2.1: Algoritmo del descenso de gradiente

Entrada: α : Tasa de aprendizaje

Entrada: θ_0 : Pesos iniciales

Entrada: $J(\theta)$: Función de costo con pesos iniciales θ

$t \leftarrow 0$ (Inicializa número de iteración)

mientras θ_t no converja **hacer**

$t \leftarrow t + 1$

$g_t \leftarrow \nabla_{\theta} J(\theta_{t-1})$ (Computa el gradiente)

$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot g_t$ (Actualiza pesos)

fin

devolver θ_t

2.3.9. Optimizador Adam

Adaptive Moment Estimation (Adam), es un método de optimización estocástica, para actualizar los pesos de un modelo. Este método calcula tasas de aprendizaje adaptativas individuales para diferentes pesos a partir de las estimaciones del primer y segundo momento de los gradientes de la función de costo [40]. El algoritmo de ADAM hace uso de dos hiperparámetros: β_1 , definido como la tasa de decaimiento exponencial para la estimación del primer momento; y β_2 , determinado como la tasa de decaimiento exponencial para la estimación del segundo momento. El algoritmo de ADAM se muestra en 2.2.

Algoritmo 2.2: Algoritmo del descenso de gradiente

Entrada: α : Tasa de aprendizaje
Entrada: $\beta_1, \beta_2 \in [0, 1)$ Tasas de decaimiento exponencial
Entrada: θ_0 : Pesos iniciales
Entrada: $J(\theta)$: Función de costo con pesos iniciales θ
 $m_0 \leftarrow 0$ (Inicializa vector del primer momento)
 $v_0 \leftarrow 0$ (Inicializa vector del segundo momento)
 $t \leftarrow 0$ (Inicializa número de iteración)
mientras θ_t *no converja* **hacer**
 $t \leftarrow t + 1$
 $g_t \leftarrow \nabla_{\theta} J(\theta_{t-1})$ (Computa el gradiente)
 $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Actualiza estimación del primer momento)
 $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Actualiza estimación del segundo momento)
 $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Corrige sesgo del primer momento)
 $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Corrige sesgo del segundo momento)
 $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Actualiza pesos)
fin
devolver θ_t

2.3.10. Entrenamiento

Después de definir el modelo de la red neuronal convolucional, y elegir la función de costo correcta para el problema que se desea solucionar, así como el método de optimización para actualizar los pesos, se procede a entrenar el modelo. Para esto, se definen unos componentes necesarios en el proceso de entrenamiento:

- **Conjunto de datos de entrenamiento:** Subconjunto de datos usado para actualizar los pesos del modelo.
- **Conjunto de datos de validación:** Subconjunto de datos utilizado para ajustar hiperparámetros del modelo.
- **Conjunto de datos de prueba:** Subconjunto para evaluar el modelo entrenado.
- **Batch size o tamaño de lote:** El número de muestras utilizadas para estimar el gradiente antes de actualizar los pesos del modelo [38].
- **Época:** El número de pases completos a través del conjunto de datos de entrenamiento antes de que finalice el proceso de entrenamiento [38].
- **Iteración:** El número de veces que se actualizan los pesos del modelo por época.
- **Tasa de aprendizaje:** Hiperparámetro que indica la cantidad que se actualiza de los pesos del modelo por iteración del algoritmo de optimización [38].

2.3.11. Sobreajuste

En cada época del entrenamiento de los modelos de las redes neuronales convolucionales, se actualizan los pesos para tener un mejor ajuste a los datos. Un modelo con muy poco ajuste a los datos no puede aprender el problema (ver Figura 2.22a), mientras que un modelo con demasiada capacidad puede aprenderlo demasiado bien y sobreajustarse al conjunto de datos usado en entrenamiento (ver Figura 2.22c), en ambos casos, se da como resultado un modelo que no se generaliza bien [38]. Con la finalidad de evitar el sobreajuste, es necesario hacer lo que se denomina como parada anticipada. El término anterior, se refiere a parar el entrenamiento del modelo de aprendizaje automático, antes de que los datos se sobreajusten al conjunto de datos.

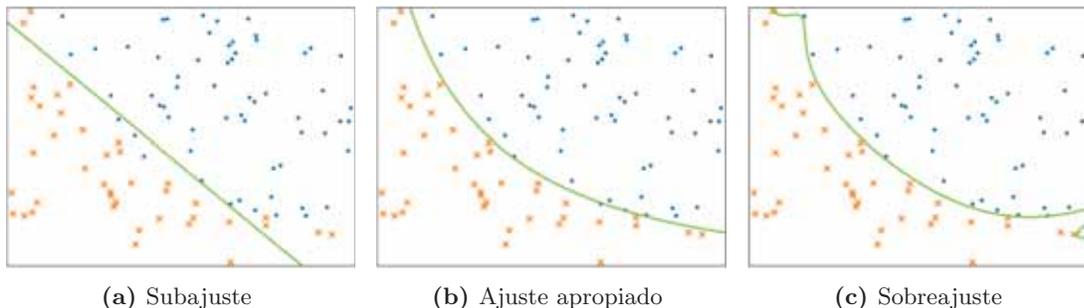


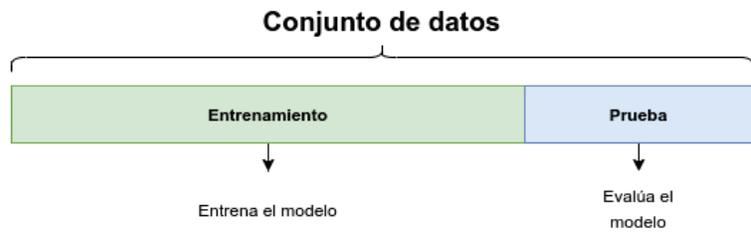
Figura 2.22: Tipos de ajustes del modelo a los datos

2.3.11.1. Validación de retención

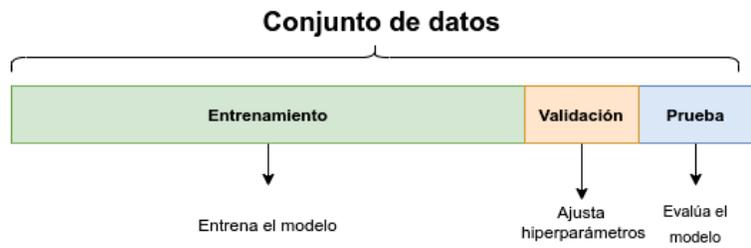
La validación de retención, es un método que puede ser usado para hacer una parada anticipada en el entrenamiento de una red neuronal convolucional, y evitar el sobreajuste. En este método se retiene una parte para probar y se entrena el modelo utilizando la parte restante de los datos [41]. Para ello, el conjunto de datos se puede dividir en dos subconjuntos, entrenamiento y prueba (ver Figura 2.23a), o bien, en tres subconjuntos, entrenamiento, validación y prueba (ver Figura 2.23b). Cabe aclarar, que la división de los subconjuntos no tiene que ser exactamente igual, y normalmente es dividido en un 70 % para el entrenamiento y un 30 % sólo para prueba y/o validación.

El rendimiento del modelo en un subconjunto de validación, se puede monitorear durante el entrenamiento y el entrenamiento se detiene cuando el error de generalización comienza a aumentar [38].

Las curvas de aprendizaje (ver Figura 2.24), pueden describir el comportamiento de aprendizaje de una red neuronal convolucional. Al término de cada época de entrenamiento, se evalúan tanto el subconjunto de entrenamiento, como el subconjunto de validación en la función de costo. Si la curva de aprendizaje del entrenamiento tiende a cero y la curva de validación aumenta su valor, se dice que el modelo se está sobreajustando.



(a) Esquema de dos subconjuntos



(b) Esquema de tres subconjuntos

Figura 2.23: Esquemas de partición de conjuntos de datos

El punto en donde se obtiene el valor más bajo de la curva de validación, es llamado punto de retención, y es el punto óptimo para tener un modelo con ajuste apropiado.

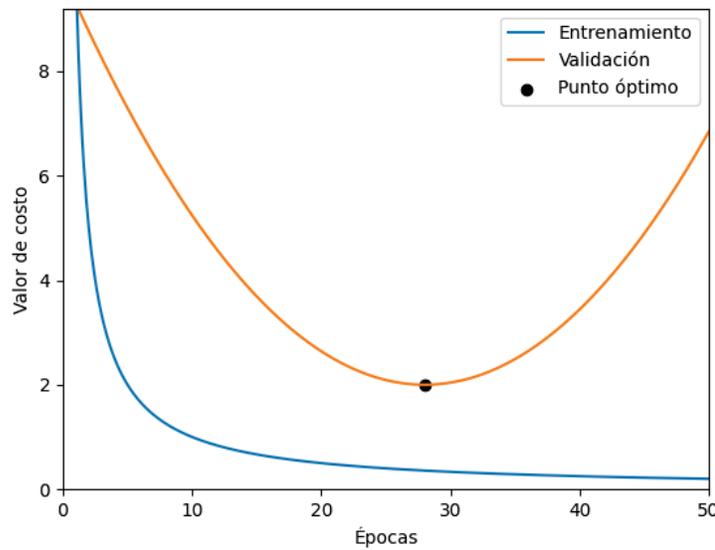


Figura 2.24: Curvas de aprendizaje

2.3.12. Red neuronal convolucional U-Net

La red neuronal convolucional U-Net [42] está compuesta por dos partes, una llamada ruta de contracción y otra denominada ruta de expansión.

Con respecto a la primera ruta, ésta consiste en cinco bloques, que contienen dos convoluciones de 3×3 con función de activación ReLU, seguida de un submuestreo de 2×2 . Además, en cada bloque de convoluciones, se duplica el número de filtros.

En el caso de la ruta expansiva, consiste en cuatro bloques, en donde se hace uso de un sobremuestreo seguido de una convolución de 2×2 que reduce a la mitad la cantidad de filtros, después se concatena con el mapa de características correspondiente de la ruta de contracción, para finalmente realizar dos convoluciones de 3×3 con función de activación ReLU.

Es necesario recalcar, que al realizar cada convolución de 3×3 , el mapa de características, pierde un borde con un ancho de 2 píxeles, por lo que, se tiene que cortar el mapa de características de la ruta de contracción antes de concatenarlo al mapa de características de la ruta de expansión, con la finalidad de que ambos tengan la misma dimensión.

En la capa final, se realiza una convolución con tamaño de kernel de 1×1 y número de filtros correspondientes al número de clases. La Figura 2.25 muestra la arquitectura U-Net.

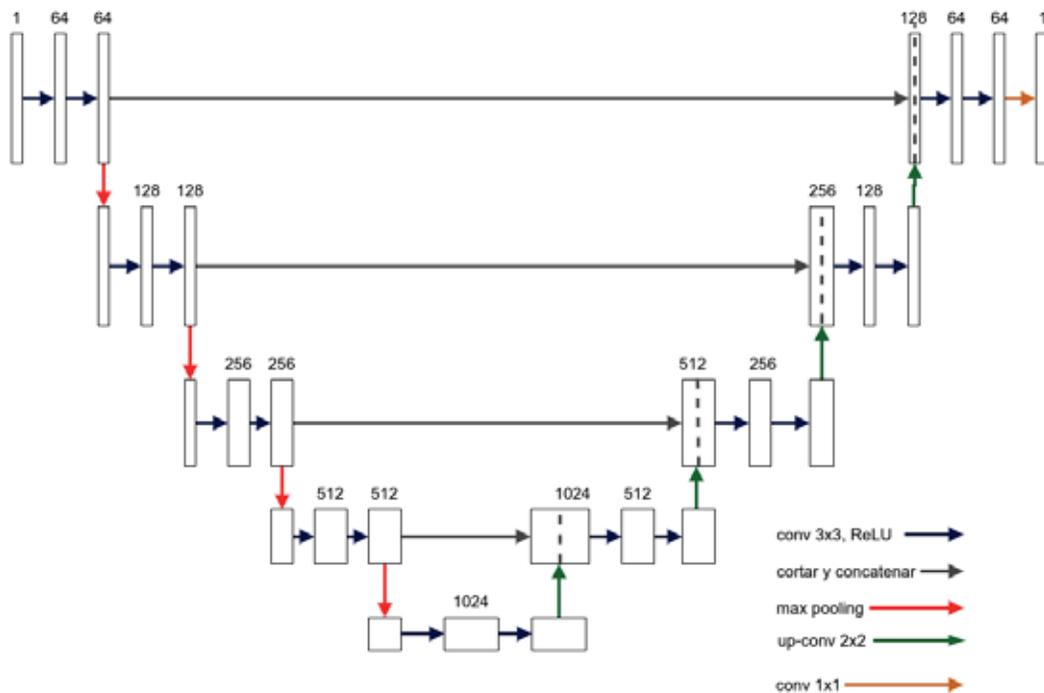
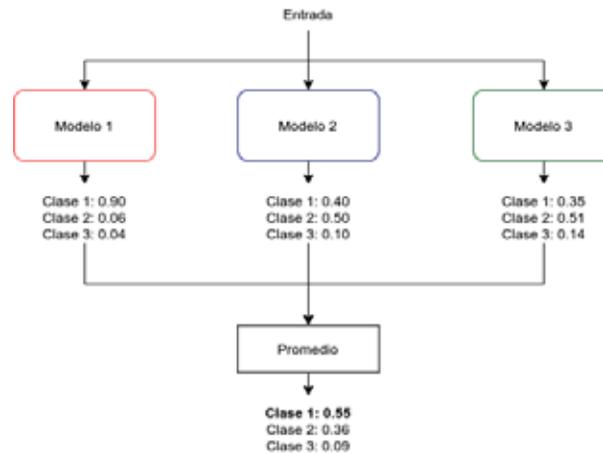


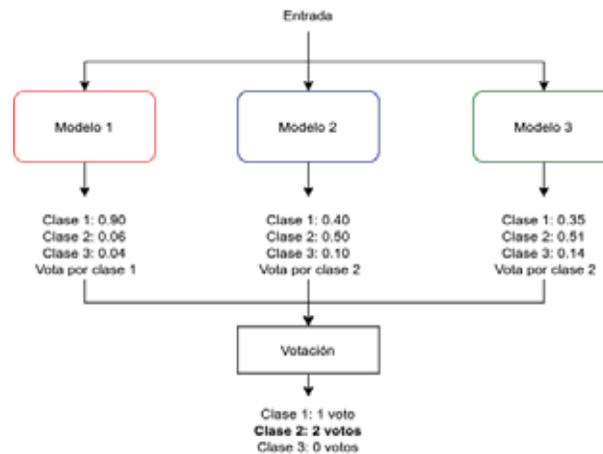
Figura 2.25: Arquitectura U-Net

2.3.13. Ensamble de redes neuronales convolucionales

Un enfoque exitoso para reducir el error de generalización de los modelos de redes neuronales es entrenar varios modelos en lugar de un solo modelo y combinar las predicciones de estos modelos [38]. Lo anterior, recibe el nombre de ensamble de redes neuronales convolucionales, y consiste en un determinado número de modelos de redes neuronales convolucionales entrenados para resolver un mismo problema. Debido a esto, la entrada de cada modelo será exactamente la misma, y como resultado se tendrá una salida por cada modelo, las cuales deberán ser combinadas para tener una única salida. Dos métodos comunes para combinar la salida de un ensamble de redes neuronales convolucionales, es el promedio (ver Figura 2.26a) y la votación (ver Figura 2.26b) .



(a) Método de promedio



(b) Método de votación

Figura 2.26: Métodos de combinación de salidas de un ensamble

2.4. Convex Hull

Un conjunto de puntos S , es convexo si y sólo si un segmento de línea recta que une dos puntos cualesquiera en S se encuentra completamente dentro de S [25] (ver Figura 2.27). Mientras que, la envolvente convexa o convex hull se refiere al conjunto convexo más pequeño que puede contener al conjunto de puntos S .

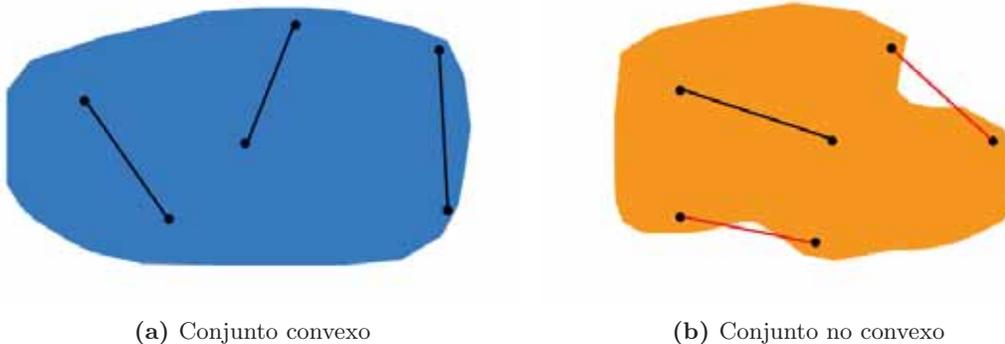


Figura 2.27: Conjuntos de puntos

Un algoritmo para computar la envolvente convexa de un conjunto de puntos es el QuickHull. Este algoritmo, se puede aplicar a N-dimensiones [43], sin embargo, inicialmente fue propuesto por [44] para dos dimensiones. El algoritmo para dos dimensiones (ver Figura 2.28), es aplicable en imágenes binarias, y consta de la siguiente serie de pasos:

1. Se determinan dos vértices primarios, donde uno es el punto con la coordenada en y más alta y el otro con la coordenada en y más baja.
2. Se define un segmento con los dos vértices primarios. Este segmento será la línea base de un triángulo.
3. Se elige el punto situado a la mayor distancia perpendicular del segmento anterior, y se forma un triángulo con los dos puntos del segmento.
4. Se descartan los puntos del interior del triángulo.
5. Se repite el proceso de los pasos 3 y 4, con los dos lados del triángulo e ignorando el segmento base, hasta que no queden puntos.
6. Los puntos seleccionados, serán los que constituyen la envolvente convexa.

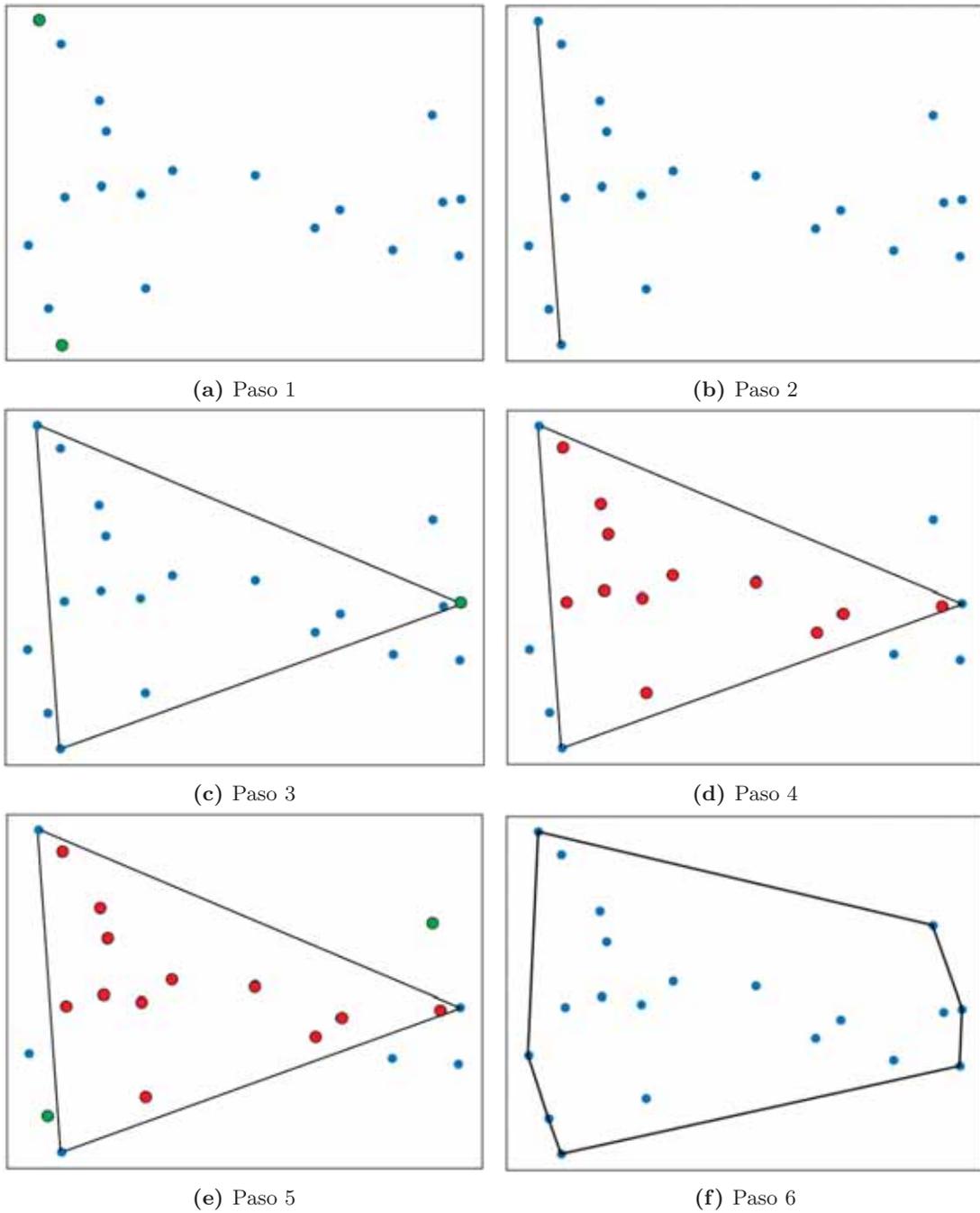


Figura 2.28: Pasos del algoritmo de Quickhull

2.5. Morfología matemática

La morfología matemática es una herramienta para extraer componentes de la imagen que son útiles en la representación y descripción de la forma de la región [25]. Dentro de la morfología matemática, existen, los llamados operadores morfológicos, como la erosión y dilatación, que consisten en trasladar un elemento estructurante (ver Figura 2.29) por toda una imagen binaria, de manera similar a un filtrado espacial. Este elemento estructurante, es una matriz que puede ser de cualquier tamaño y su función es definir la vecindad de operación.

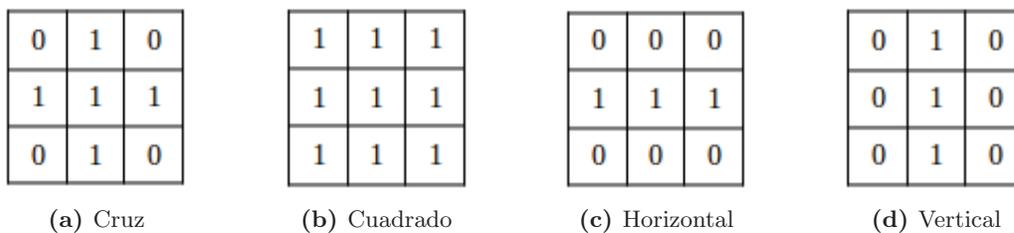


Figura 2.29: Formas comunes de elementos estructurantes de 3×3

Cuando se está trasladando el elemento estructurante por la imagen, los operadores morfológicos hacen uso de algunas operaciones sobre la vecindad del elemento estructurante (ver Figura 2.30), las cuales se definen como:

- **Hit:** Algunos de los elementos con valor 1 del elemento estructurante, coincide con uno o más píxeles con valor 1 de la imagen binaria.
- **Fit:** Todos los elementos con valor 1 del elemento estructurante, coinciden con píxeles con valor 1 de la imagen binaria.
- **Miss:** Ninguno de los píxeles con valor 1 del elemento estructurante, coincide con algún píxel con valor 1 de la imagen binaria.

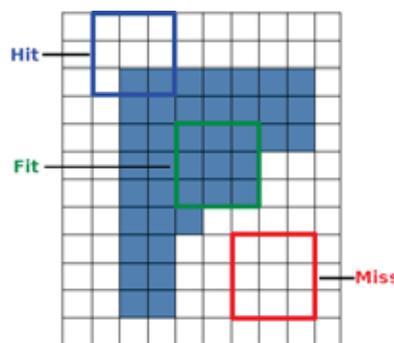


Figura 2.30: Operaciones de vecindad con elemento estructurante cuadrado de 3×3

2.5.1. Dilatación

La dilatación es un operador morfológico, que posiciona el centro del elemento estructurante S en todos los píxeles (x, y) de la imagen I y efectúa la siguiente operación:

$$g(x, y) = I \oplus S = \begin{cases} 1, & \text{si } S \text{ hits } I \\ 0, & \text{Otro caso} \end{cases} \quad (2.15)$$

En la Figura 2.31, se observa que la operación morfológica de dilatación, tiene el objetivo de expandir la estructura geométrica contenida en la imagen.

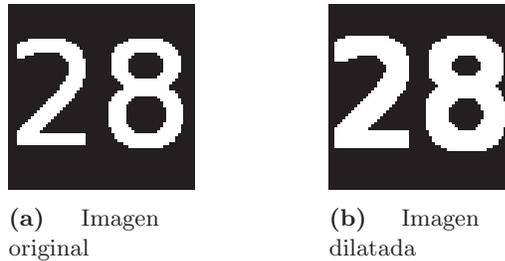


Figura 2.31: Dilatación con elemento estructurante cuadrado de 3×3

2.5.2. Erosión

La erosión es otro tipo de operador morfológico, el cual también posiciona el centro del elemento estructurante S en todos los píxeles (x, y) de la imagen I y realiza la operación:

$$g(x, y) = I \ominus S = \begin{cases} 1, & \text{si } S \text{ fits } I \\ 0, & \text{Otro caso} \end{cases} \quad (2.16)$$

En la Figura 2.32, se muestra que al aplicar la operación morfológica de erosión, la estructura geométrica contenida en la imagen se reduce.

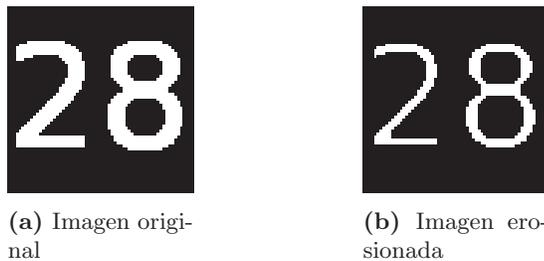


Figura 2.32: Erosión con elemento estructurante cuadrado de 3×3

2.5.3. Extracción de componentes conectados

La extracción de componentes conectados, consiste en etiquetar con un valor único las diferentes regiones con píxeles conectados dentro de una imagen. Dos píxeles están conectados cuando son vecinos y tienen el mismo valor. Si bien, existen varios métodos para realizar esta función, en [25] se indica que para extraer una región de la imagen, se debe elegir un píxel X_0 contenido en la región e iterar 2.17 hasta que $X_{k-1} = X_k$, y donde B es un elemento estructurante con conectividad 8, lo cual significa que es un elemento estructurante cuadrado de 3×3 .

$$X_k = (X_{k-1} \oplus B) \cap I \quad k = 1, 2, 3, \dots \quad (2.17)$$

Con la finalidad de extraer más de un componente conectado, se debe iterar 2.17 para un punto X_0 de cada región. En la Figura 2.33, se muestra algunas iteraciones del algoritmo para extraer dos regiones contenidas en una imagen binaria.

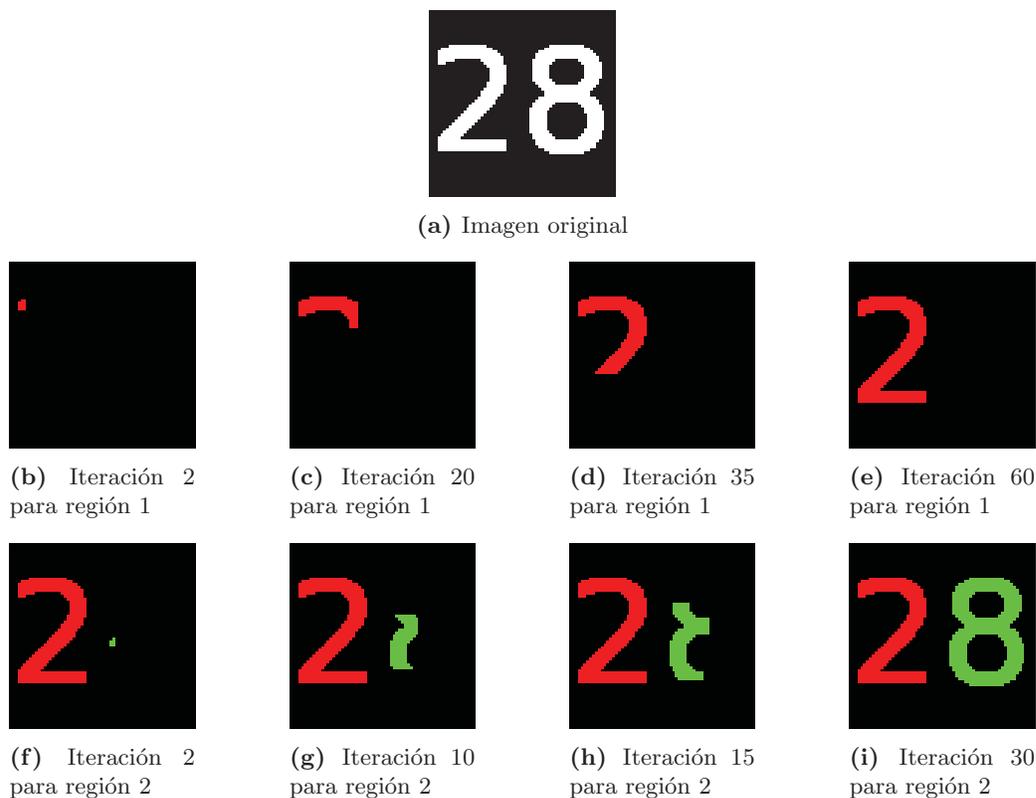


Figura 2.33: Iteraciones del algoritmo de extracción de componentes conectados

2.5.4. Rellenado de orificios

Este método consiste en rellenar los orificios en el interior de las regiones de una imagen binaria, con píxeles de valor 1. Para realizar esto, en [25] se describe una manera semi-automática para el rellenado de orificios, la cual consiste en definir a X_0 como una selección de píxeles dentro de los orificios de las regiones que se desea rellenar. Luego, se procede a iterar 2.18, hasta que $X_{k-1} = X_k$, y donde B es un elemento estructurante con conectividad 4, es decir, un elemento estructurante de cruz de 3×3 . Finalmente, para obtener la imagen con orificios rellenados, X_k se une con la imagen binaria con orificios.

$$X_k = (X_{k-1} \oplus B) \cap I^c \quad k = 1, 2, 3, \dots \quad (2.18)$$

En cambio, la manera automática del rellenado de orificios (ver Figura 2.35), consiste en realizar una invasión del complemento de la imagen binaria, desde el límite exterior de la imagen. Por tanto, X_0 se define como un marco que rodea el límite exterior de una imagen con ceros. Posteriormente, de forma análoga a la manera semi-automática, se itera 2.18, hasta que $X_{k-1} = X_k$, y donde B es un elemento estructurante con conectividad 4. Por último, para obtener la imagen con orificios rellenados, se obtiene el complemento X_k .

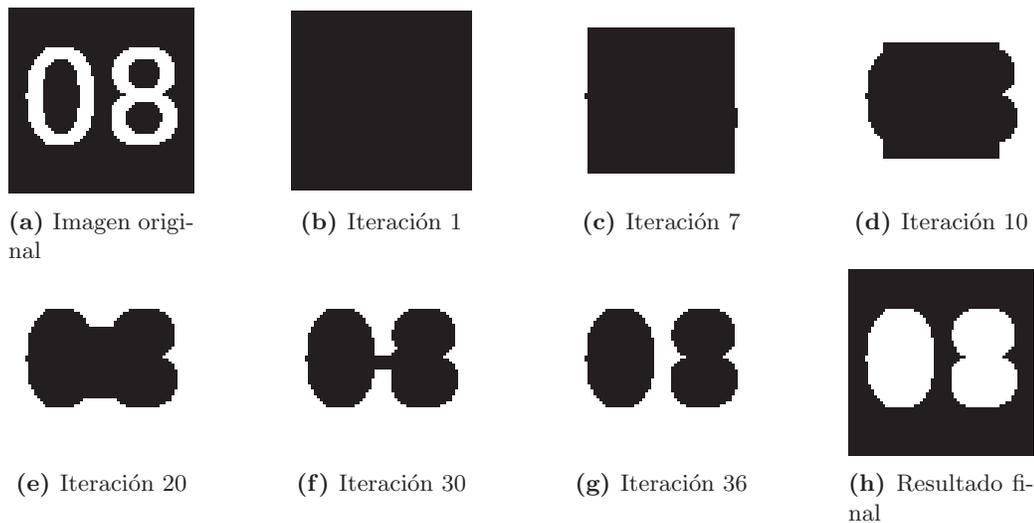


Figura 2.34: Iteraciones del algoritmo de rellenado de orificios

2.5.5. Detección de bordes de regiones etiquetadas

El objetivo de este método es encontrar los límites de cada región etiquetada por algún algoritmo de análisis de componentes conectados. Como el valor de todos los píxeles de una región etiquetada es un número entero único, se considera que, un píxel es un píxel de límite si alguno de sus vecinos tiene un valor diferente.

En consecuencia, el algoritmo para detectar los bordes, funciona como un filtrado morfológico, en donde un elemento estructurante de conectividad 4, se traslada en cada posición (x, y) de la imagen y se hace una operación parecida a la del hit. Esta operación verifica si algún píxel ubicado en las posiciones, donde el elemento estructurante tiene valor de 1, es diferente al valor del píxel ubicado en la posición central del elemento estructurante. Si algún valor es diferente, significa que el píxel central es un píxel de borde (ver Figura 2.35).

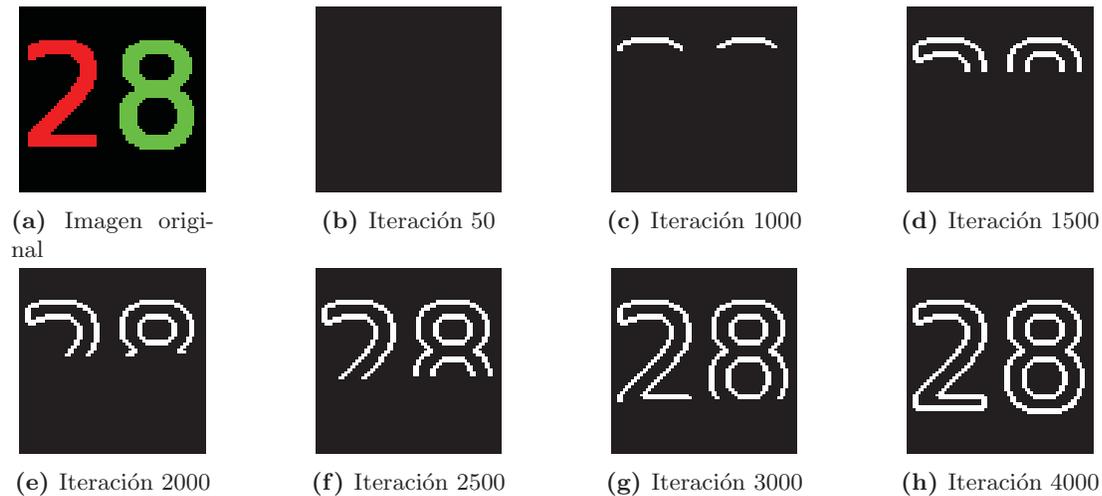


Figura 2.35: Iteraciones del algoritmo de detección de bordes

En el presente capítulo se describen los métodos que se emplearon en el desarrollo del presente proyecto. Se detalla el procedimiento para adecuar la base de datos de imágenes del ensayo cometa a conjuntos de datos para segmentación. Luego, se expone la arquitectura usada en el ensamble redes neuronales convolucionales, y posteriormente como se entrenó. Finalmente, se profundiza en el desarrollo de cada módulo que integra el sistema propuesto.

3.1. Funcionamiento general del sistema

El sistema propuesto para la segmentación de células procesadas por electroforesis en gel se compone de cinco módulos, cada uno con una tarea en específico. La entrada del sistema es una imagen en escalas de grises o una imagen RGB, mientras que, en la salida se obtiene la misma imagen de entrada, pero con contornos añadidos delimitando los cometas y sus respectivas cabezas, al igual que líneas sobrepuestas, que representan las distancias del largo de la cola y el radio de la cabeza. Además, el sistema genera un archivo con extensión .csv con los parámetros calculados de los cometas identificados. En la Figura 3.1, se puede observar un diagrama representativo del sistema desarrollado.

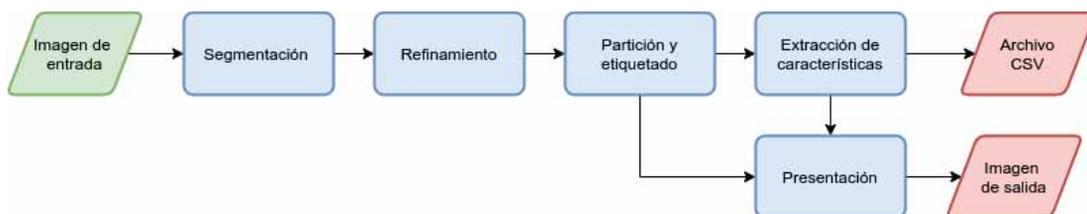


Figura 3.1: Diagrama general del sistema

De manera general, se explica el objetivo de cada módulo del sistema:

- **Segmentación:** Identifica por medio de un ensamble de redes neuronales convolucionales, la región de la cola y la cabeza de los cometas contenidos en la imagen de entrada.

- **Refinamiento:** Aplica diversas técnicas con base a probabilidad y morfología matemática, para eliminar imperfecciones, regiones anómalas y falsos positivos de la segmentación.
- **Partición y etiquetado:** Establece los límites de cada cometa en una región de traslape, y etiqueta con un número entero cada uno de los cometas.
- **Extracción de características:** Para cada cometa identificado y etiquetado, se calculan varias características de interés y se guardan en un archivo .csv, en donde se asocia un número a cada cometa e indica el nombre de la imagen en donde está contenido.
- **Presentación:** Sobrepone contornos, líneas y puntos a la imagen de entrada representando los límites de los cometas y características como el largo de la cola, radio de la cabeza y los centroides del cometa y cabeza. Además, asigna un número a cada cometa correspondientemente al número asociado en el archivo con extensión csv.

3.2. Base de datos

La base de datos original constó de 140 imágenes con células procesadas por el ensayo cometa, en escala de grises, con un tamaño de 1388×1038 píxeles, se muestra un ejemplo en la Figura 3.2. Las células contenidas en todas las imágenes presentan diferentes grados de daño, formas y tamaños, por lo que se considera que se tiene cierta diversidad de cometas. Para entrenar cada uno de los modelos de redes neuronales convolucionales, se agruparon las 140 imágenes en tres conjuntos:

- **Conjunto de entrenamiento:** Sirvió para ajustar los parámetros de los modelos de redes neuronales convolucionales y constó de 116 imágenes.
- **Conjunto de validación:** Ayudó a ajustar hiperparámetros y evitar el sobreajuste de los modelos al conjunto de entrenamiento. Este conjunto se compuso de 14 imágenes.
- **Conjunto de prueba:** Se usó para evaluar el desempeño del sistema desarrollado y se compuso de 10 imágenes.

3.2.1. Ajuste y segmentación manual de la base de datos

El entrenamiento de los modelos de redes neuronales convolucionales requiere de bastantes recursos computacionales, por lo que, con el fin de evitar fallas debido a la insuficiente memoria o un tiempo de procesamiento excesivo, se decidió hacer recortes, con un tamaño de 288×288 píxeles, en las imágenes de los conjuntos de datos de entrenamiento y validación. El número de recortes que se hicieron a cada imagen original fue variando en función de la cantidad de cometas contenidos en la imagen, y la cantidad

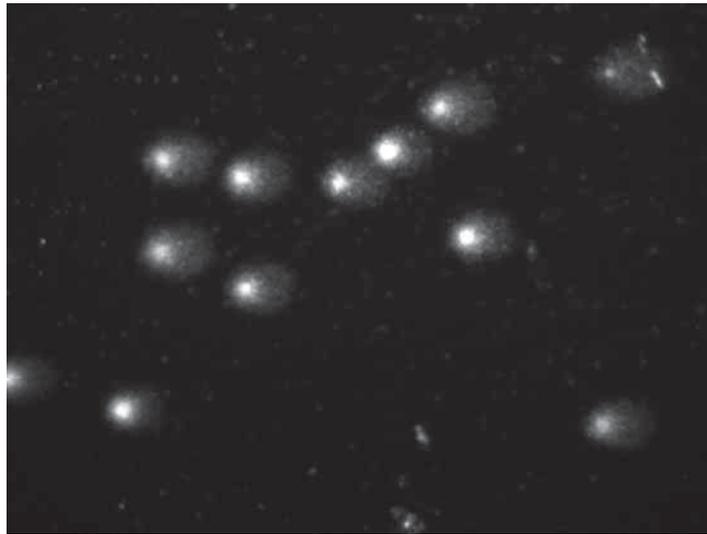


Figura 3.2: Muestra de la base de datos

de cometas contenidos en un recorte varió en función del tamaño de los cometas de la imagen original. Adicionalmente, se hizo una segmentación manual, tanto en los recortes de los conjuntos de entrenamiento y validación, como en el conjunto de prueba. En esta segmentación manual se marcaron tres clases:

- **Clase 0:** Fondo de la imagen
- **Clase 1:** Cola del cometa
- **Clase 2:** Cabeza del cometa

El procedimiento para recortar y segmentar manualmente, realizado con el software libre GIMP y octave, se describe a continuación:

1. Contener en un recuadro de dimensión 288×288 , cometas que no hayan sido segmentados anteriormente.
2. Abarcar mínimo el 90 % del área de un cometa.
3. Tratar de que los cometas estén centrados.
4. Segmentar la imagen recortada, con valor de intensidad de 165 en escala de grises, la cabeza del cometa y dejar el fondo restante de color negro.
5. Segmentar la imagen recortada, con valor intensidad de 90 en escala de grises, la cola del cometa y dejar el fondo restante de color negro.
6. Sumar las dos imágenes segmentadas.

3. METODOLOGÍA

En la Figura 3.3 se puede observar de manera gráfica como se realizó el procedimiento descrito. Al finalizar este proceso, se obtuvieron 664 recortes de imágenes para el conjunto de entrenamiento, y 87 recortes de imágenes para el conjunto de validación.

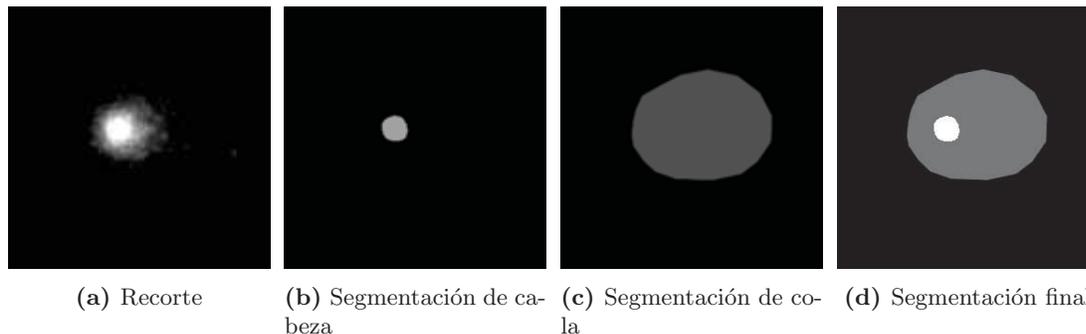


Figura 3.3: Proceso de segmentación manual

Por último, con la librería `numpy` de `python`, se codificaron los recortes segmentados manualmente en el formato `one-hot`, siendo el primer canal de profundidad perteneciente al fondo, el segundo a la cola, y el tercero a la cabeza. El resultado de este proceso, se guardó en archivos `.npy`, los cuales fueron usados para entrenar los modelos de redes neuronales convolucionales del ensamble.

3.3. Arquitectura

El ensamble de redes neuronales convolucionales está integrado por cinco modelos de la red neuronal convolucional para imágenes biológicas, denominado U-Net [42], el cual fue ligeramente modificado. La primera modificación que se realizó fue rellenar con ceros en los límites de los mapas de características, de manera que, después de cada convolución, se mantenga el tamaño de entrada. Como resultado de lo anterior, el tamaño de la imagen de entrada del modelo U-Net, es igual al tamaño de la imagen de salida, difiriendo únicamente en la profundidad. De igual forma, se efectuó una modificación en la sección de decoder del modelo U-Net, específicamente, se reemplazó el llamado “up-convolution”, por una convolución transpuesta de 2×2 con stride de 2 píxeles, provocando así, que se calculen los mejores parámetros para aumentar de tamaño los mapas de activación. La última modificación realizada fue aumentar el tamaño del kernel en la convolución final, pasando de un kernel de 1×1 a un kernel de 3×3 , de este modo, se toman más píxeles vecinos para clasificar el píxel final. De acuerdo con las modificaciones realizadas, la arquitectura de cada modelo U-Net del ensamble, se describe en el Tabla 3.1.

Arquitectura U-Net				
Entrada				$288 \times 288 \times 1$
Convolución 1.1	3×3	64	ReLU	$288 \times 288 \times 64$
Convolución 1.2	3×3	64	ReLU	$288 \times 288 \times 64$
Maxpooling				$144 \times 144 \times 64$
Convolución 2.1	3×3	128	ReLU	$144 \times 144 \times 128$
Convolución 2.2	3×3	128	ReLU	$144 \times 144 \times 128$
Maxpooling				$72 \times 72 \times 128$
Convolución 3.1	3×3	256	ReLU	$72 \times 72 \times 256$
Convolución 3.2	3×3	256	ReLU	$72 \times 72 \times 256$
Maxpooling				$36 \times 36 \times 256$
Convolución 4.1	3×3	512	ReLU	$36 \times 36 \times 512$
Convolución 4.2	3×3	512	ReLU	$36 \times 36 \times 512$
Maxpooling				$18 \times 18 \times 512$
Convolución 5.1	3×3	1024	ReLU	$18 \times 18 \times 1024$
Convolución 5.2	3×3	1024	ReLU	$18 \times 18 \times 1024$
Convolución transpuesta	2×2	512	ReLU	$36 \times 36 \times 512$
Concatenación con 4.2				$36 \times 36 \times 1024$
Convolución 6.1	3×3	512	ReLU	$36 \times 36 \times 512$
Convolución 6.2	3×3	512	ReLU	$36 \times 36 \times 512$
Convolución transpuesta	2×2	256	ReLU	$72 \times 72 \times 256$
Concatenación con 3.2				$72 \times 72 \times 512$
Convolución 7.1	3×3	256	ReLU	$72 \times 72 \times 256$
Convolución 7.2	3×3	256	ReLU	$72 \times 72 \times 256$
Convolución transpuesta	2×2	128	ReLU	$144 \times 144 \times 128$
Concatenación con 2.2				$144 \times 144 \times 256$
Convolución 8.1	3×3	128	ReLU	$144 \times 144 \times 128$
Convolución 8.2	3×3	128	ReLU	$144 \times 144 \times 128$
Convolución transpuesta	2×2	64	ReLU	$288 \times 288 \times 64$
Concatenación con 1.2				$288 \times 288 \times 128$
Convolución 9.1	3×3	64	ReLU	$288 \times 288 \times 64$
Convolución 9.2	3×3	64	ReLU	$288 \times 288 \times 64$
Convolución 10	3×3	3	Softmax	$288 \times 288 \times 3$

Tabla 3.1: Arquitectura U-Net usada para el ensamble

Con respecto a las entradas y salidas del ensamble, este usa exactamente la misma entrada para cada uno de los cinco modelos U-Net, por lo cual, al terminar el procesamiento, se tendrán cinco mapas de probabilidades. Para obtener la salida del ensamble, se calcula el promedio de los mapas de probabilidad resultantes. Un diagrama de lo anterior se ilustra en la Figura 3.4.

3.4. Entrenamiento

El entrenamiento de los cinco modelos U-Net se realizó de manera independiente, por medio de la plataforma Google Colab. Los entrenamientos de los modelos se hicieron con los mismos conjuntos de datos de entrenamiento y validación, sin embargo, para obtener modelos con diferentes pesos finales, la inicialización de los pesos de los kernel

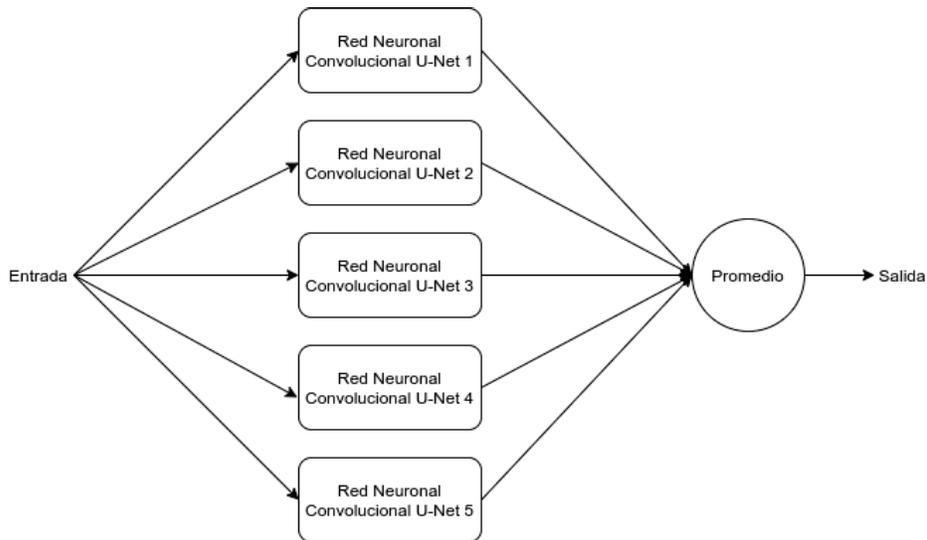


Figura 3.4: Diagrama de entrada/salida del ensemble

se generó aleatoriamente, de acuerdo con la inicialización uniforme de Glorot, o también llamada inicialización uniforme de Xavier [34]. En relación con la función de costo, se estableció la función categorical cross-entropy, puesto que se trata de un problema de segmentación multiclase.

Los hiperparámetros y configuraciones que se hicieron para entrenar cada modelo U-Net del ensemble, se indican a continuación:

- **Optimizador:** Adam
- **Tasa de aprendizaje:** 1×10^{-4}
- **Factor β_1 :** 0.9
- **Factor β_2 :** 0.999
- **Tamaño de lote:** 16 imágenes
- **Épocas de entrenamiento:** 50

Con la finalidad de evitar el sobreajuste de los modelos a los datos de entrenamiento, se usó el método de validación de retención, es decir, al término de cada época, se evaluaba el modelo con la función de costo y los datos de validación, si el resultado era menor a la anterior época, se guardaban los pesos del modelo, en caso contrario, no se sobrescribían los pesos.

3.5. Módulo de segmentación

El módulo de segmentación del sistema está diseñado con la finalidad de obtener el mapa de probabilidades generado por el ensamble de redes neuronales convolucionales, en donde se indique para cada píxel, la probabilidad de pertenecer a una clase, y de acuerdo con eso, crear la imagen segmentada. La arquitectura de cada modelo de red neuronal convolucional del ensamble requiere que la entrada sea de un tamaño de $288 \times 288 \times 1$, por lo que, para realizar segmentaciones de imágenes de cualquier tamaño, es necesario ejecutar dos procesos adicionales.

El primero es reconocer por medio del número de canales de profundidad, si la imagen de entrada está en escala de grises o a color (RGB). Si se reconoce que es una imagen en escala de grises, es decir, se tiene un canal de profundidad, no se realiza ninguna operación. Por otra parte, si es una imagen a color, se suman los valores de intensidad para cada canal, y se escoge el canal que obtuvo el valor más alto. Por ende, en ambos casos, como resultado se obtiene una imagen con solo un canal de profundidad, tal como se requiere con la arquitectura de los modelos del ensamble. En la Figura 3.5, se puede visualizar el resultado de una conversión RGB a escala de grises.

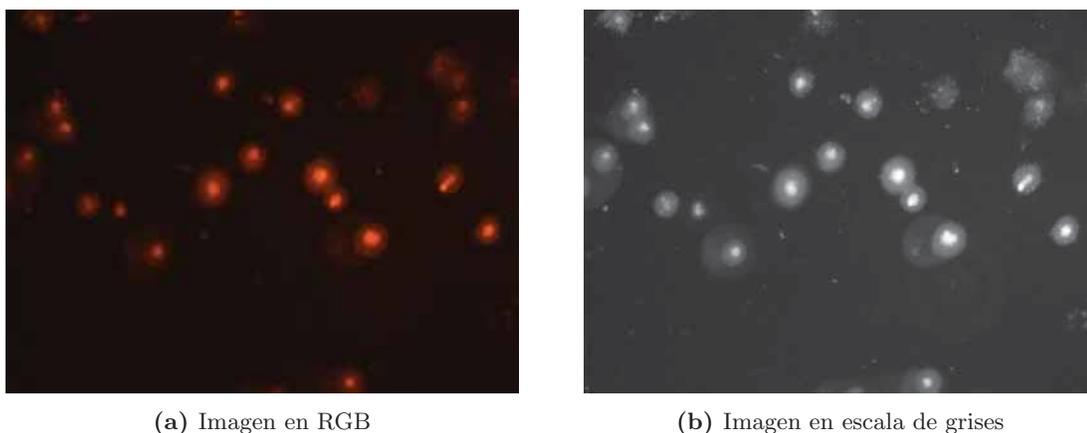


Figura 3.5: Proceso de conversión de imagen RGB a escala de grises

El otro proceso consiste en tomar porciones de 288×288 de la imagen a procesar. Para esto, se sigue una variante de la estrategia Overlap-Tile descrita en [42]. Inicialmente, se agregan bordes que reflejen el contenido de la imagen, con el fin de que la dimensión de la imagen sea un múltiplo de 128. Luego, con una ventana de 288×288 se hacen un barrido, que inicia desde la esquina superior izquierda y termina en la esquina inferior derecha. En este barrido, se toma una porción de 288×288 píxeles de la imagen, se procesa con un modelo del ensamble y se obtiene un mapa de probabilidad de $288 \times 288 \times 3$, aun así, únicamente se asigna un cuadro de $128 \times 128 \times 3$ píxeles ubicados en el centro de la imagen procesada al resultado de la salida. En la Figura 3.6, se puede observar una imagen representativa de este proceso. Cabe aclarar, que este proceso se realiza por cada uno de los cinco modelos del ensamble.

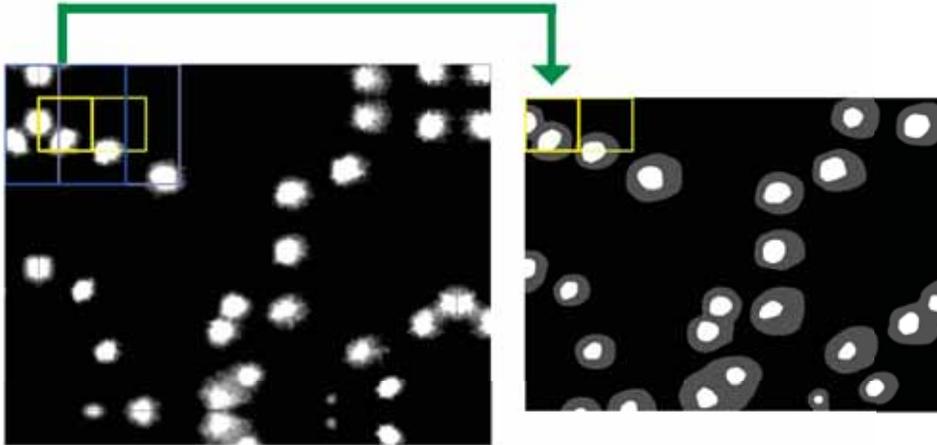


Figura 3.6: Proceso de barrido de segmentación

Al concluir el anterior proceso, se consiguen cinco mapas de probabilidades con tres canales de profundidad (uno para cada clase), de los cuales, para determinar el mapa de probabilidades de salida del ensamble, se calcula el promedio para cada píxel. De esta forma, se obtiene solamente un mapa de probabilidad de profundidad tres. Un ejemplo de esto se muestra en la Figura 3.7c., en donde, se puede observar que los píxeles clasificados como fondo tienen colores con tonalidades rojas, los píxeles de la cola con tonalidades verdes y los píxeles de la cabeza con tonalidades azules. Las tonalidades más bajas representan una menor probabilidad, por lo que se puede decir, que las probabilidades en los límites de cada clase, son menores a las probabilidades del interior de la región de una clase.

Por último, de acuerdo con el mapa de probabilidades generado por el ensamble, se halla para cada píxel, el canal de profundidad con probabilidad más alta, por consiguiente, se obtiene la imagen segmentada, en donde cada píxel solo puede obtener un valor entre 0, 1 o 2. Se muestra un ejemplo en la Figura 3.7d.

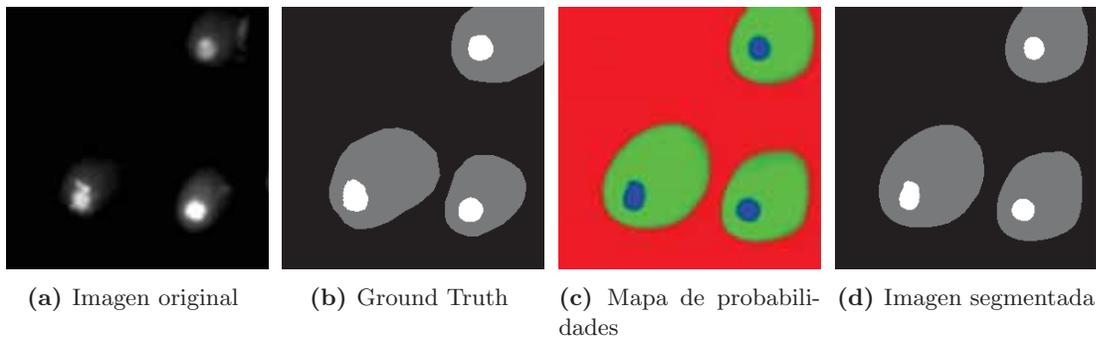


Figura 3.7: Proceso del módulo de segmentación

3.6. Módulo de refinamiento

La segmentación realizada por el ensamble de redes neuronales convolucionales puede contener algunos errores de identificación, algunos de ellos se pueden visualizar en la Figura 3.8. Uno de estos errores, es que el ensamble identifica como cometa a algunas pequeñas partículas o suciedad que está contenida en la imagen (ver Figura 3.8b). Asimismo, puede ocurrir que, en la segmentación realizada por el modelo, los bordes de las regiones identificadas como cometas, tengan una forma muy irregular (ver Figura 3.8d). Los errores escritos anteriormente, pueden ocasionar un grave problema para la obtención de las características que se desean extraer, por lo tanto, el presente módulo tiene la tarea de perfeccionar la segmentación realizada por el ensayo de redes neuronales convolucionales, mediante seis técnicas aplicadas secuencialmente a la imagen segmentada, las cuales se describirán a continuación.

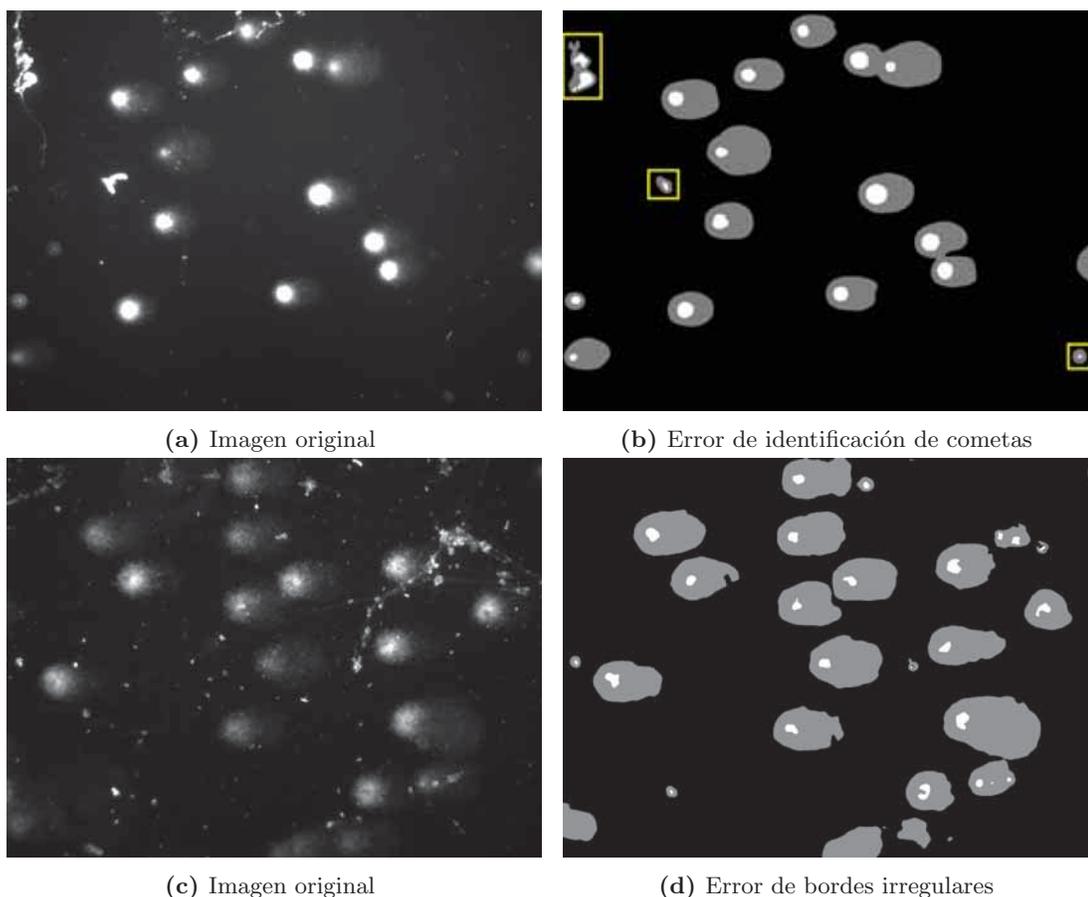
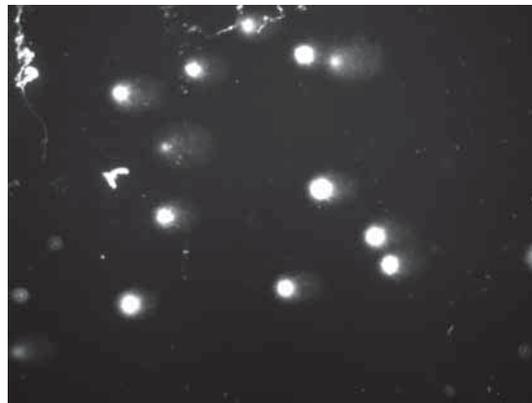


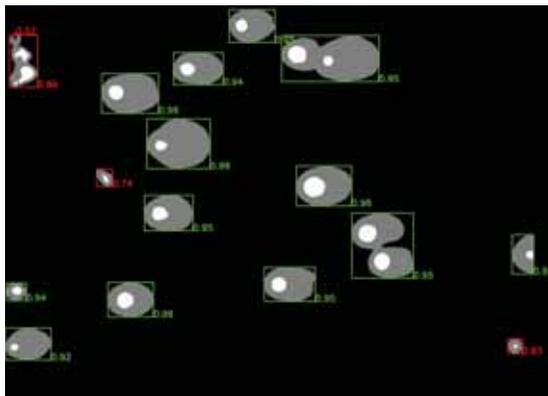
Figura 3.8: Errores comunes en la segmentación

3.6.1. Eliminación de cometas con probabilidad baja

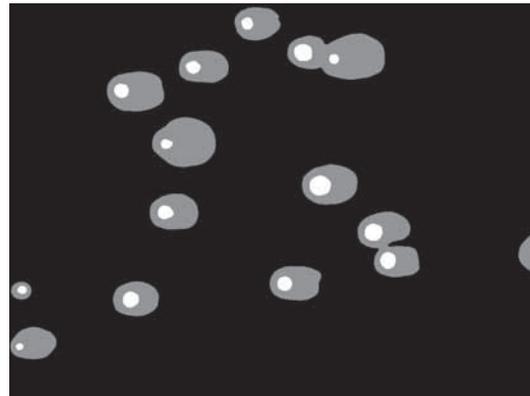
Las imágenes obtenidas por el ensayo cometa no están exentas de tener pequeñas partículas que no sean cometas, pero que su forma sea parecida a la de un cometa, en consecuencia, existe la probabilidad de que el ensamble prediga estas partículas como cometas. Sin embargo, si se identifica una partícula erróneamente como cometa, la probabilidad calculada por el ensamble para la mayoría de los píxeles que conforma la región de esta partícula es menor que la mayoría de las probabilidades de los píxeles de una región que en realidad si es cometa. Por esa razón, esta técnica tiene el objetivo de eliminar las regiones que tengan un promedio de probabilidad menor a 0.90. Para realizar esto, primeramente, se obtiene la probabilidad de ser cometa para cada píxel, esto es, la suma de la probabilidad de la clase cola con la probabilidad de la clase cabeza. Luego, se calcula el promedio de las probabilidades de ser cometa de cada píxel contenido en una región marcada como cometa, y si el resultado obtenido es menor a 0.90, la región que había sido identificada como cometa, se marcará como clase fondo (Ver Figura 3.9).



(a) Imagen original



(b) Probabilidad de los cometas

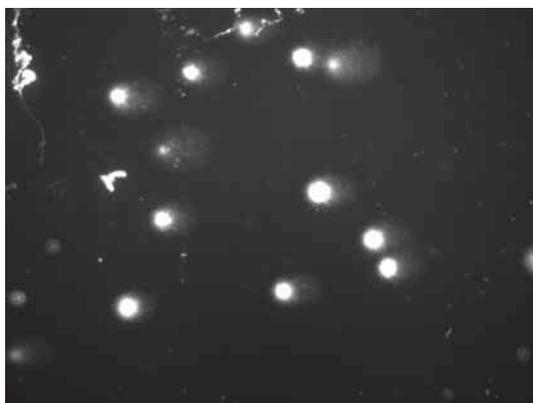


(c) Resultado de la técnica

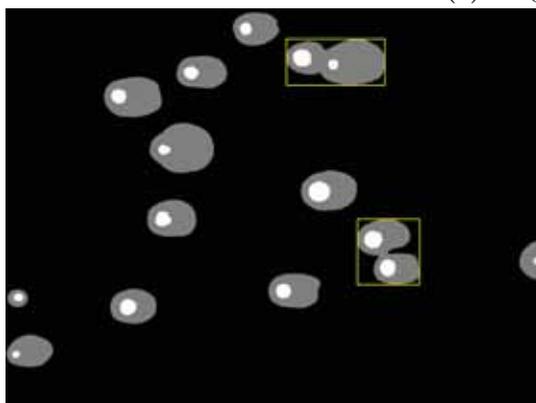
Figura 3.9: Técnica de eliminación de cometas con probabilidad baja

3.6.2. Separación de cometas con ligero traslape

Al aplicar el proceso de electroforesis a las células de ADN, es posible que varios fragmentos de una célula viajen hacia otra célula cercana y colisionen entre sí, ocasionando el traslape de cometas. Como resultado, puede que el ensamble identifique dos o más cometas ligeramente traslapados o que estén demasiado cerca sin tocarse, como un único cometa con varias cabezas en su interior. Por lo tanto, para solventar el problema planteado, el sistema identifica como cometas traslapados, aquellas regiones que en su interior tengan una o más cabezas, y debido a que las probabilidades de los píxeles en los bordes de los cometas son menores en comparación al resto, para cada una de las regiones identificadas, se eliminan los píxeles con probabilidad menor a 0.90. Si como resultado se obtienen dos o más regiones separadas, significa que el proceso fue exitoso y se procede a actualizar la imagen segmentada, con estas regiones separadas. De lo contrario, si no se obtienen más regiones, el proceso no fue exitoso, por lo que no se actualiza la imagen segmentada. En la Figura 3.10, se puede observar que al aplicar esta técnica, el proceso fue exitoso para las dos regiones identificadas.



(a) Imagen original



(b) Identificación de cometas traslapados



(c) Resultado de la técnica

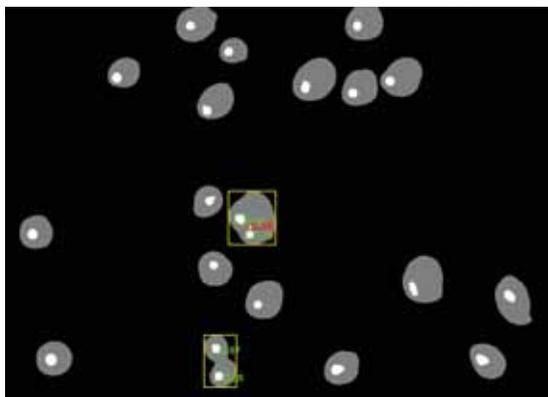
Figura 3.10: Técnica de separación de cometas ligeramente traslapados

3.6.3. Eliminación de cabezas de posibles cometas traslapados

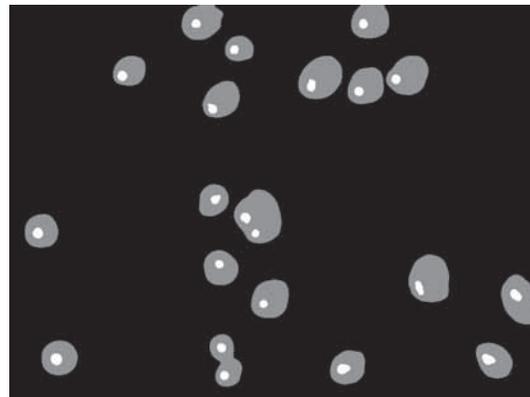
Algunas regiones identificadas como cometas por el ensamble de redes neuronales convolucionales pueden contener dos o más cabezas en su interior. Esto puede significar una región de traslape de cometas, o bien, un cometa con dos o más cabezas identificadas erróneamente. Con el fin de eliminar errores como el último mencionado, esta técnica tiene el propósito de remover aquellas cabezas con un promedio de probabilidad menor a 0.7, únicamente en regiones con más de una cabeza en su interior. Inicialmente, se identifican las regiones que contengan en su interior dos o más cabezas. Después, se calcula el promedio de las probabilidades de clase cabeza de cada píxel contenido en una región clasificada como cabeza, y si el resultado obtenido es menor a 0.7, la cabeza es eliminada y los píxeles se marcan como clase cola. En la Figura 3.11, se muestra que al aplicar esta técnica, se removió una cabeza (marcada en rojo) de las tres identificadas en un cometa.



(a) Imagen original



(b) Probabilidades de las cabezas de posibles cometas traslapados



(c) Resultado de la técnica

Figura 3.11: Técnica de eliminación de cabezas de posibles cometas traslapados

3.6.4. Aplicación de la envolvente convexa

En algunas ocasiones, el fondo de las imágenes del ensayo cometa pueden tener variación de luminosidad y mucho ruido, lo que puede ocasionar que, en la segmentación realizada por el ensamble de redes neuronales convolucionales, los cometas presenten bordes muy irregulares, provocando que se pierda la forma ovalada que se desea tener, para extraer características como el área y largo del cometa. Por lo mismo, se calcula la envolvente convexa para todos los píxeles contenidos en una región identificada como cometa no traslapado o intacto, de esta manera, se obtendrán cometas con formas más ovaladas y bordes menos irregulares. Adicionalmente, se rellenan todos los posibles orificios dentro de la región de la envolvente convexa. De modo semejante, se calcula la envolvente convexa para todos los píxeles contenidos en una región clasificada como cabeza, por tanto, se obtienen cabezas más redondas, facilitando el cálculo del diámetro de la cabeza. En la Figura 3.12, se puede visualizar el resultado al usar esta técnica.

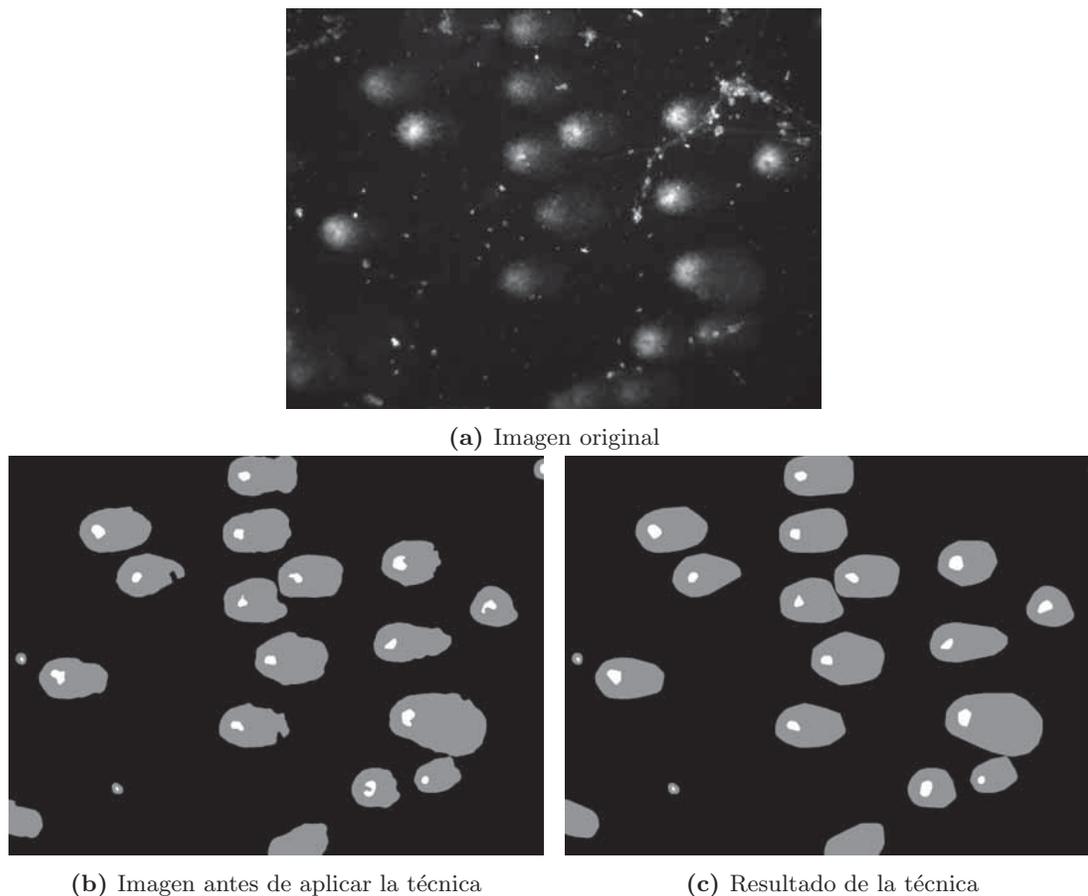
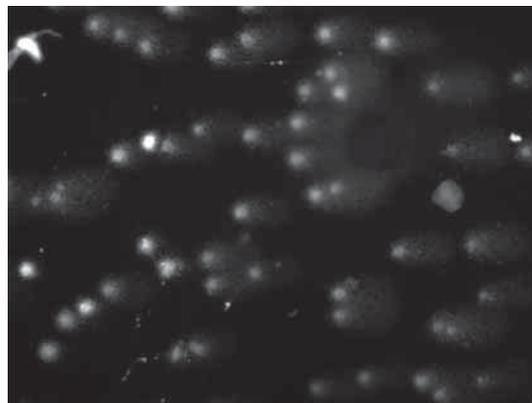


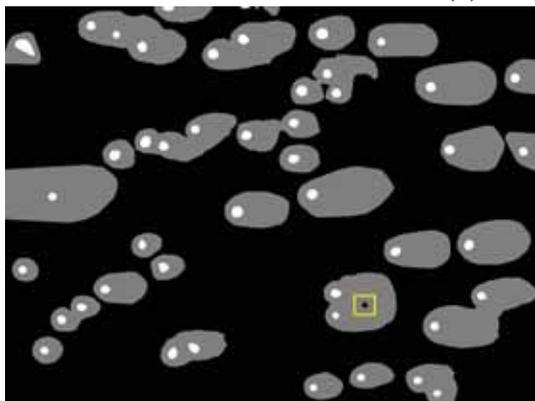
Figura 3.12: Técnica de aplicación de envolvente convexa

3.6.5. Relleno de orificios en cometas traslapados

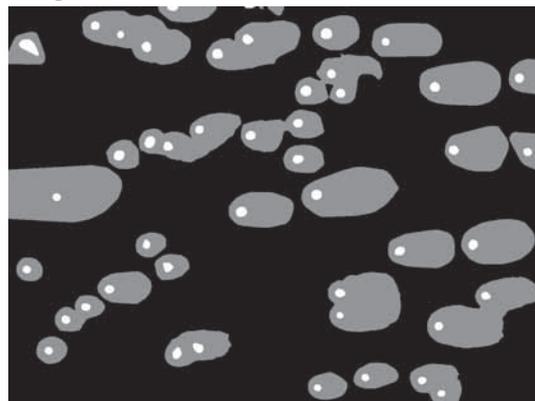
Debido a los residuos o partículas no pertenecientes a las células del ensayo cometa, que se introducen en el interior de un cometa, el ensamble puede que clasifique esas partículas como fondo, provocando la aparición de orificios en el interior de los cometas. En este punto del refinamiento, el problema ya fue resuelto en la anterior técnica para los cometas intactos, por lo tanto, para resolver el problema en los cometas traslapados, primero se convierte la imagen segmentada en una imagen binaria, donde 0 es fondo y 1 es cometa. Luego, se halla el complemento de la imagen binaria y posteriormente se utiliza un algoritmo de morfología matemática, que consiste en invadir el complemento de la imagen binaria desde el límite exterior de la imagen, mediante dilataciones con un elemento estructurante de 3×3 con conectividad 4. El algoritmo rodea las regiones identificadas como cometas y no invade dentro de ellas, ocasionado que en su interior no existan orificios. De manera similar, el algoritmo es usado para las cabezas. Al final, se halla nuevamente el complemento de ambas imágenes y se suman. Un ejemplo del relleno de un orificio se muestra en la Figura 3.13.



(a) Imagen original



(b) Identificación de orificios

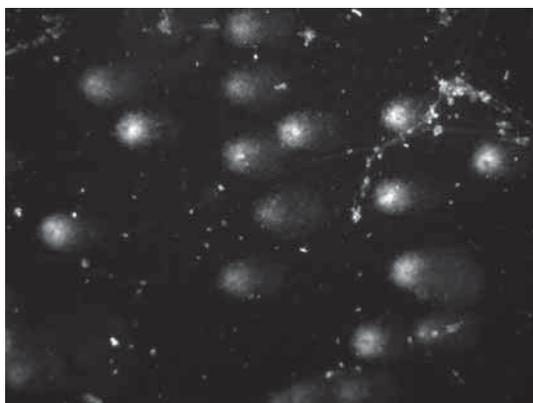


(c) Resultado de la técnica

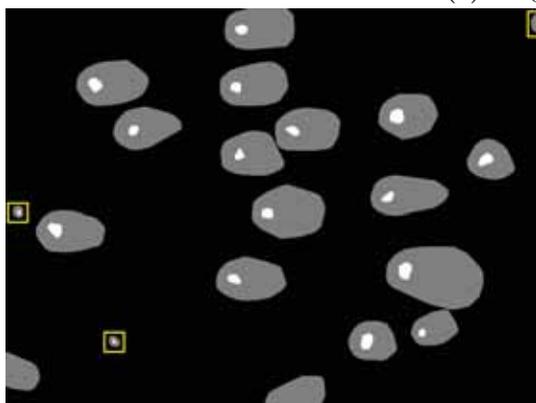
Figura 3.13: Técnica de relleno de orificios en cometas traslapados

3.6.6. Eliminación de cometas con áreas anómalas

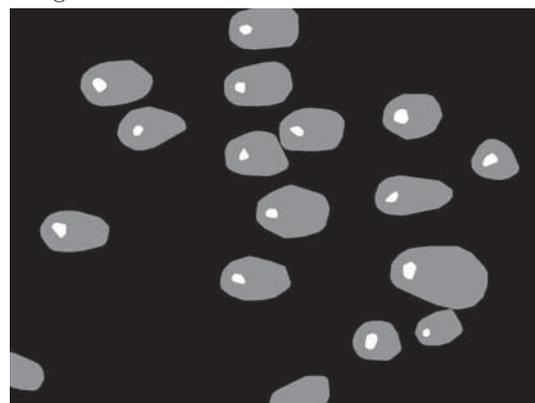
En una imagen del ensayo cometa, normalmente las células de ADN presentan un grado de daño similar, por lo que, si en la imagen segmentada, existe un cometa con un área demasiado diferente al resto de los cometas identificados, es muy probable que esta región haya sido identificada erróneamente como cometa. La solución para este problema consiste en eliminar aquellas regiones con áreas menores al promedio menos k veces la desviación estándar. En primer lugar, se calculan las áreas de todos los cometas contenidos en la imagen. Posteriormente, se calcula el promedio y la desviación estándar de las áreas calculadas de los cometas. Acto seguido, se halla el valor de umbral, determinado por el promedio menos k veces la desviación estándar, donde k tiene un valor inicial de 10. No obstante, si el umbral obtenido es un número negativo, el valor de k va disminuyendo hasta obtener un valor positivo. Finalmente, se remueven las regiones que tengan un área menor al umbral calculado. El resultado de esta técnica se muestra en la Figura 3.14, en ella se observa que se pudieron eliminar algunas partículas que no representaban cometas.



(a) Imagen original



(b) Cometas con áreas anómalas



(c) Resultado de la técnica

Figura 3.14: Técnica de eliminación de cometas con áreas anómalas

3.7. Módulo de partición y etiquetado

Después de obtener la imagen segmentada y refinada, el presente módulo se encarga de etiquetar de manera única cada cometa, en otras palabras, se asigna un número único a cada región identificada como cometa. Lo anterior se realiza con la finalidad de poder identificar individualmente a cada cometa para la extracción de sus características. No obstante, es probable que en la imagen segmentada se encuentren cometas traslapados y esto significaría un inconveniente para la extracción de las características individuales de cada cometa. Por lo tanto, se realiza un proceso exclusivamente para las regiones con este problema, el cual tiene el objetivo de particionar la región de los cometas traslapados en dos o más cometas de acuerdo con el número de cabezas dentro de la región.

De manera inicial, se distingue entre cometas traslapados e intactos. Los cometas traslapados se identifican como aquellas regiones que tienen dos o más cabezas en su interior, mientras que los cometas intactos son las regiones con una única cabeza.

Para etiquetar los cometas intactos se aplica un algoritmo de morfología matemática, en específico, se hace una extracción de componentes conectados. Este algoritmo es capaz de identificar regiones por medio de operaciones de dilatación a la imagen con un elemento estructurante de 3×3 con conectividad 8. Una imagen con los cometas intactos etiquetados se muestra en la Figura 3.15

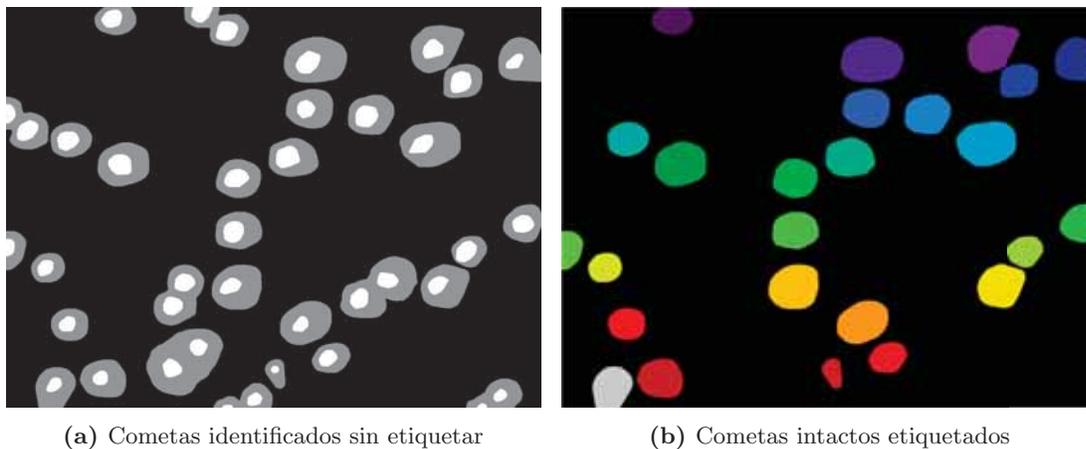


Figura 3.15: Etiquetación de cometas intactos

En cambio, para etiquetar a los cometas traslapados, primero se encuentran las líneas divisorias que particionen al área en cometas individuales. Para esto, se realiza el algoritmo watershed [27], el cual hallará el área correspondiente para cada cabeza, de acuerdo con los valores de intensidad de la imagen de entrada en escala de grises. Los valores de intensidad altos cercanos a una cabeza indican una más alta probabilidad de pertenecer al cometa, que valores de intensidad más bajos. Dicho con otras palabras, se utilizan las cabezas como los marcadores para el algoritmo watershed, y se utilizan los negativos de los valores de intensidad de la imagen de entrada en escala de grises

como el mapa de distancias. El resultado de aplicar el algoritmo Watershed, permite la etiquetación de los cometas traslapados, lo cual se puede observar en la Figura 3.16.

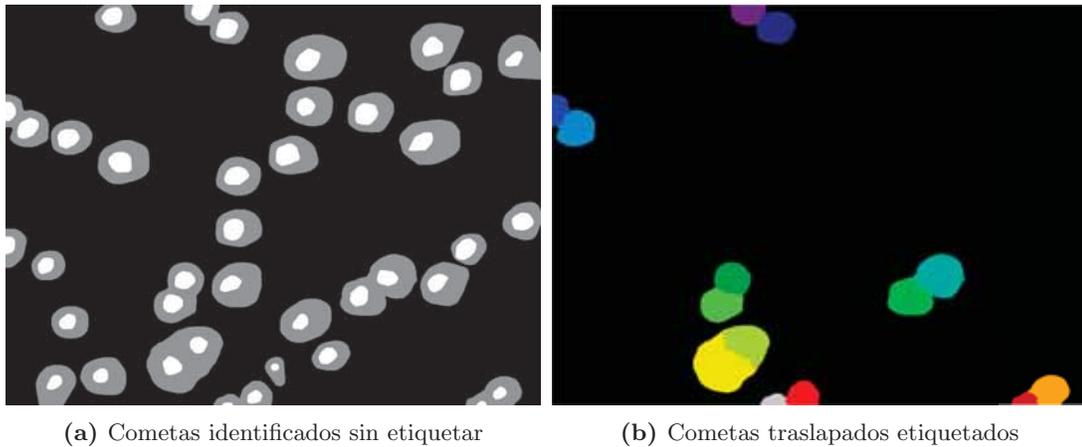


Figura 3.16: Etiquetación de cometas traslapados

Después de etiquetar tanto los cometas intactos, como los cometas traslapados, se fusionan las dos imágenes etiquetadas. Los primeros números de etiqueta se destinan a los cometas intactos, mientras que los últimos números a los cometas traslapados. En la Figura 3.17, se puede observar el etiquetado final que realiza este módulo, donde cada color diferente representa un cometa.

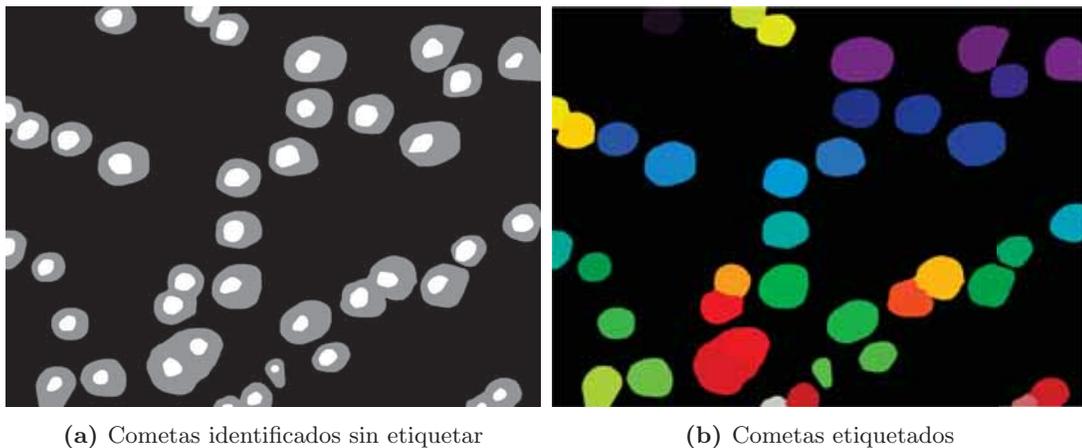


Figura 3.17: Etiquetación de cometas

3.8. Módulo de extracción de características

Este módulo es el encargado de generar uno de los dos resultados del sistema: el archivo CSV con las características extraídas. Para ello, el módulo recibe como entrada la imagen con todos los cometas etiquetados que se obtuvo del módulo anterior y calcula las características presentadas en la Tabla 3.2, tanto para los cometas intactos como para los cometas traslapados, sin distinción alguna.

Característica	Definición
Área del cometa	Número de píxeles del cometa.
Largo del cometa	Distancia desde donde empieza el cometa hasta donde termina.
Contenido de ADN en el cometa	Suma de las intensidades de los píxeles del cometa.
Intensidad promedio del cometa	Contenido de ADN del cometa entre el tamaño del cometa.
Área de la cabeza	Número de píxeles de la cabeza.
Diámetro de la cabeza	Distancia desde donde empieza la cabeza hasta donde termina.
Contenido de ADN en la cabeza	Suma de las intensidades de los píxeles de la cabeza.
Intensidad promedio de la cabeza	Contenido de ADN de la cabeza entre el tamaño de la cabeza.
Porcentaje de ADN en la cabeza	Porcentaje de ADN en la cabeza con respecto al contenido de ADN del cometa.
Área de la cola	Número de píxeles de la cola
Largo de la cola	Distancia desde donde termina la cabeza hasta donde termina el cometa.
Contenido de ADN en la cola	Suma de las intensidades de los píxeles de la cola.
Intensidad promedio de la cola	Contenido de ADN de la cola entre el tamaño de la cola.
Porcentaje de ADN en la cola	Porcentaje de ADN en la cola con respecto al contenido de ADN del cometa.
Momento de cola	Producto entre el largo del cometa y el porcentaje de ADN en la cola.
Momento de Olive	Producto del porcentaje de ADN en la cola con la distancia entre los centroides de la cabeza y cola.

Tabla 3.2: Características extraídas por el sistema

Adicionalmente, se indica el número de etiqueta asociado a cada cometa en el archivo CSV, con el fin de poder visualizar en la imagen de salida, que características le pertenecen a cada cometa. Cabe aclarar, que los primeros números de etiqueta se asocian a los cometas libres, mientras que los últimos números de etiqueta se relacionan con los cometas traslapados. Además, como el sistema es capaz de procesar un conjunto de imágenes, se señala dentro del archivo CSV, el nombre de la imagen que se procesó.

3.9. Módulo de presentación

El módulo de presentación tiene como propósito facilitar la visualización de los cometas identificados al usuario. Para ello, se sobreponen contornos, líneas, puntos y números para representar o indicar distintas características, como los límites del cometa, el largo de la cola, el radio de la cabeza, entre otros. Con el fin de distinguir los trazos sobrepuestos en la imagen de salida, se describe en la Tabla 3.3, el significado de cada trazo junto con su color.

Tipo	Color	Representación
Contorno	Azul	Contorno de cometa intacto
Contorno	Gris	Contorno de cometa traslapado
Contorno	Rojo	Contorno de cabeza
Línea	Verde	Largo de la cola
Línea	Rosa	Radio de la cabeza
Punto	Azul	Centroide del cometa
Punto	Rojo	Centroide de la cabeza
Número	Amarillo	Número de cometa intacto
Número	Gris	Número de cometa traslapado

Tabla 3.3: Código de colores del módulo de presentación

Los bordes de los cometas y sus respectivas cabezas son identificados mediante un algoritmo de morfología matemática que actúa sobre la imagen con cometas etiquetados. Este algoritmo tiene como objetivo encontrar los bordes de las regiones de la imagen, a través de un filtrado con un elemento estructurante de 3×3 con conectividad 4.

En cuanto a las características, como el radio de la cabeza, el largo de la cola y el grado de daño, son definidos con ayuda de las características extraídas del anterior módulo explicado. Estas características se muestran con la finalidad de que el usuario observe con más información, el grado de daño que tiene la célula.

Con respecto a los centroides de los cometas y las cabezas, estos son el resultado del promedio de las coordenadas de todos los píxeles que conforman la región del cometa o la región de la cabeza.

Los números de los cometas son indicados con la finalidad de poder observar en el archivo CSV, que características le pertenecen. En la Figura 3.18 se muestra la salida de este módulo.

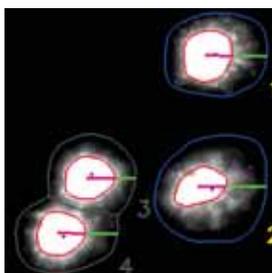


Figura 3.18: Salida del módulo de presentación

En este capítulo se muestran los resultados obtenidos en el proyecto de investigación. Específicamente, se muestra el proceso de aprendizaje de los cinco modelos de red neuronal convolucional del ensamble, así como también, el desempeño de su segmentación de acuerdo con las métricas de precisión, exhaustividad y F1-score. Adicionalmente, se comparan los resultados obtenidos con otros sistemas computacionales enfocados al mismo problema de segmentación y extracción de características de las células del ensayo cometa.

4.1. Análisis del aprendizaje de las redes neuronales convolucionales

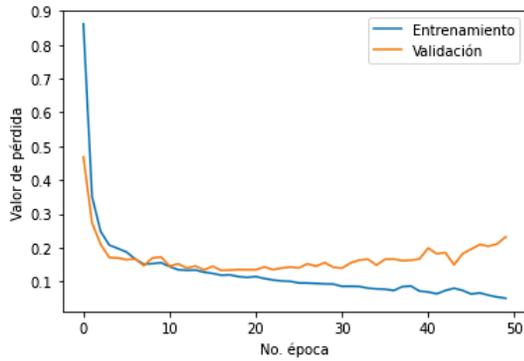
Al entrenar cada uno de los cinco modelos de red neuronal convolucional del ensamble con los parámetros indicados en la sección de metodología, se observó mediante las curvas de aprendizaje (ver Figura 4.1), un comportamiento similar para cada modelo.

Con respecto a las curvas de aprendizaje del entrenamiento, se observó que, a medida que aumentaron las épocas, los modelos fueron ajustándose lentamente a los datos del conjunto de entrenamiento, provocando valores de pérdida cada vez menores. Las curvas de aprendizaje se presentan de una forma suave y tendiendo a un valor de pérdida cero, lo que indica que los modelos aprendieron el problema a un ritmo considerable. Sin embargo, la tendencia de la curva a un valor de pérdida cero indica un posible sobreajuste de los modelos a los datos del conjunto de entrenamiento.

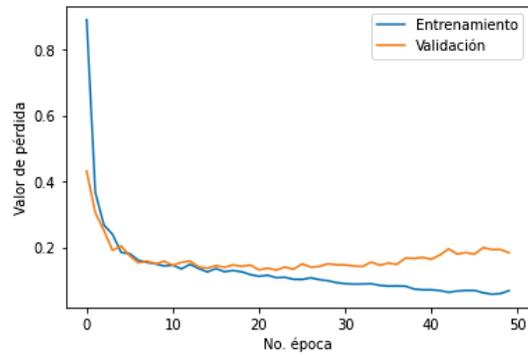
En el caso de las curvas de aprendizaje de la validación, se observa un comportamiento diferente a las curvas del entrenamiento. Si bien al principio se nota una disminución en los valores de pérdida, después de unas épocas, los valores de pérdida fueron creciendo paulatinamente, formando así una curva convexa. Lo previamente explicado, indica que los modelos efectivamente se estaban sobreajustando.

A pesar del sobreajuste de los modelos, debido al método de validación de retención, se obtuvieron los pesos del modelo que dan como resultado el valor mínimo de la curva convexa de validación, por lo que se puede decir, que se tienen los pesos más óptimos para cada modelo de red neuronal convolucional del ensamble, que se pueden obtener con la inicialización aleatoria de Glorot Uniform y durante las 50 épocas.

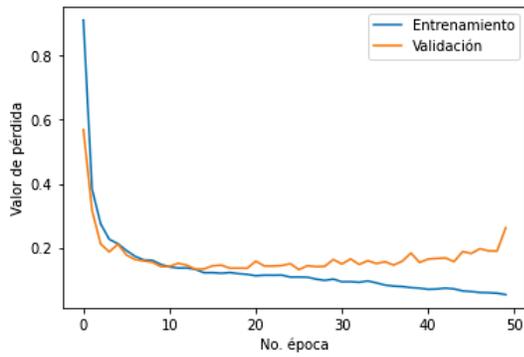
4. RESULTADOS



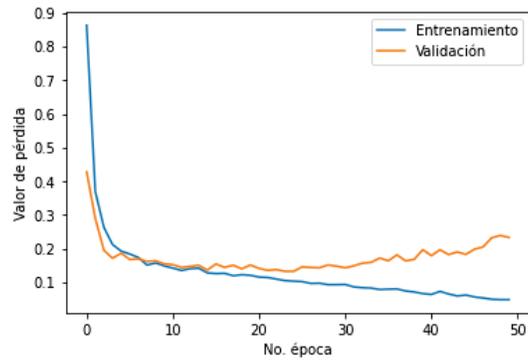
(a) Curva de aprendizaje del modelo 1



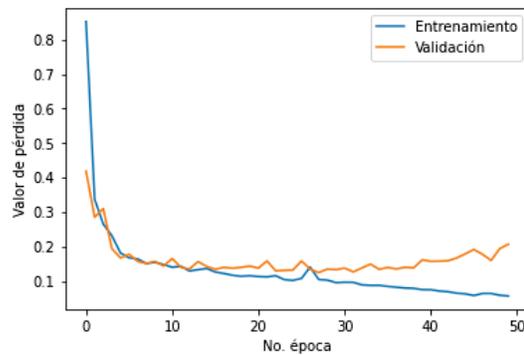
(b) Curva de aprendizaje del modelo 2



(c) Curva de aprendizaje del modelo 3



(d) Curva de aprendizaje del modelo 4



(e) Curva de aprendizaje del modelo 5

Figura 4.1: Curvas de aprendizaje de los modelos del ensamble

4.2. Análisis de la segmentación

En esta sección se muestran los resultados de segmentación obtenidos por el módulo de segmentación y refinamiento del sistema, con el fin de conocer el desempeño de ambos módulos de acuerdo con las métricas de F1-score y evaluar si las técnicas de refinamiento mejoraron el desempeño de la segmentación. De manera general, la métrica de F1-score (ver Tabla 4.1) indica que el módulo de refinamiento pudo mejorar la segmentación para cada una de las clases (fondo, cola y cabeza). En cuanto a la precisión promedio, se observa nuevamente en la Tabla 4.1, que el módulo de refinamiento pudo aumentar el desempeño, no obstante, la exhaustividad promedio obtenida fue ligeramente menor.

Clase	Segmentación			Refinamiento		
	Precisión	Exhaustividad	F1-score	Precisión	Exhaustividad	F1-score
Fondo	0.976	0.968	0.972	0.972	0.977	0.975
Cola	0.808	0.861	0.834	0.847	0.840	0.843
Cabeza	0.888	0.710	0.789	0.926	0.714	0.806
Promedio	0.891	0.846	0.865	0.915	0.844	0.875

Tabla 4.1: Desempeño del módulo de segmentación y del módulo de refinamiento

El desempeño por imagen mostrado en la Tabla 4.2, indica que el módulo de refinamiento pudo mejorar la segmentación de 6 imágenes, en donde la mayor diferencia de F1-score fue de 0.168, lo que significa que si se pudo eliminar regiones no pertenecientes a cometas. En cuanto a las 4 imágenes que no se pudo mejorar su segmentación, la mayor disminución fue de 0.023. Adicionalmente, de estas 4 imágenes se observa que si se mejoró la precisión, pero la exhaustividad fue menor, lo cual sugiere que se eliminó algún cometa, como por ejemplo en la Figura 4.2. Las imágenes segmentadas por ambos módulos se muestran de la Figura 4.2 a la Figura 4.11.

Imagen de prueba Número	Segmentación			Refinamiento		
	Precisión	Exhaustividad	F1-score	Precisión	Exhaustividad	F1-score
1	0.942	0.863	0.899	0.950	0.855	0.897
2	0.949	0.937	0.943	0.945	0.943	0.944
3	0.960	0.911	0.931	0.957	0.916	0.933
4	0.639	0.759	0.688	0.970	0.798	0.856
5	0.905	0.945	0.924	0.952	0.941	0.946
6	0.876	0.824	0.846	0.877	0.784	0.823
7	0.889	0.707	0.762	0.938	0.727	0.791
8	0.720	0.657	0.674	0.863	0.639	0.655
9	0.888	0.773	0.813	0.895	0.763	0.810
10	0.821	0.851	0.814	0.866	0.841	0.832

Tabla 4.2: Comparación por imagen de los resultados del módulo de segmentación y el módulo de refinamiento

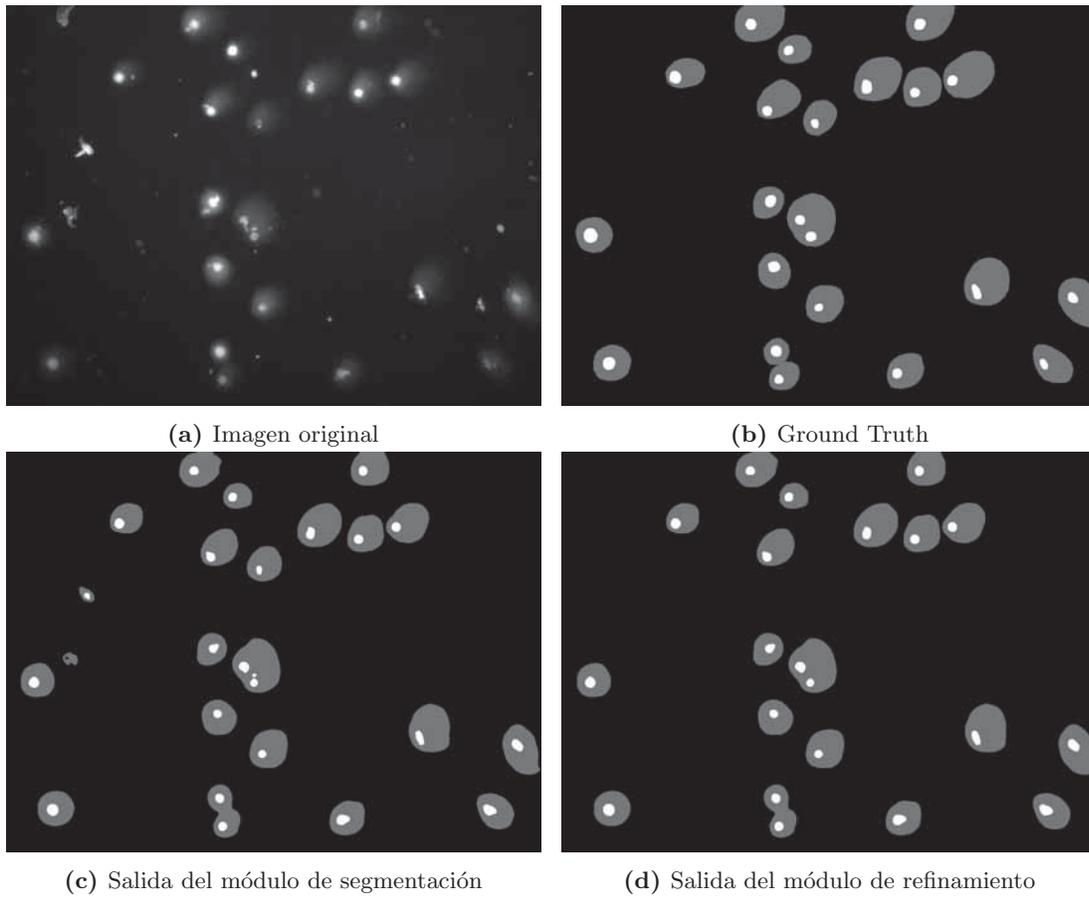


Figura 4.2: Segmentación de la imagen de prueba 1 realizada por los módulos de segmentación y refinamiento

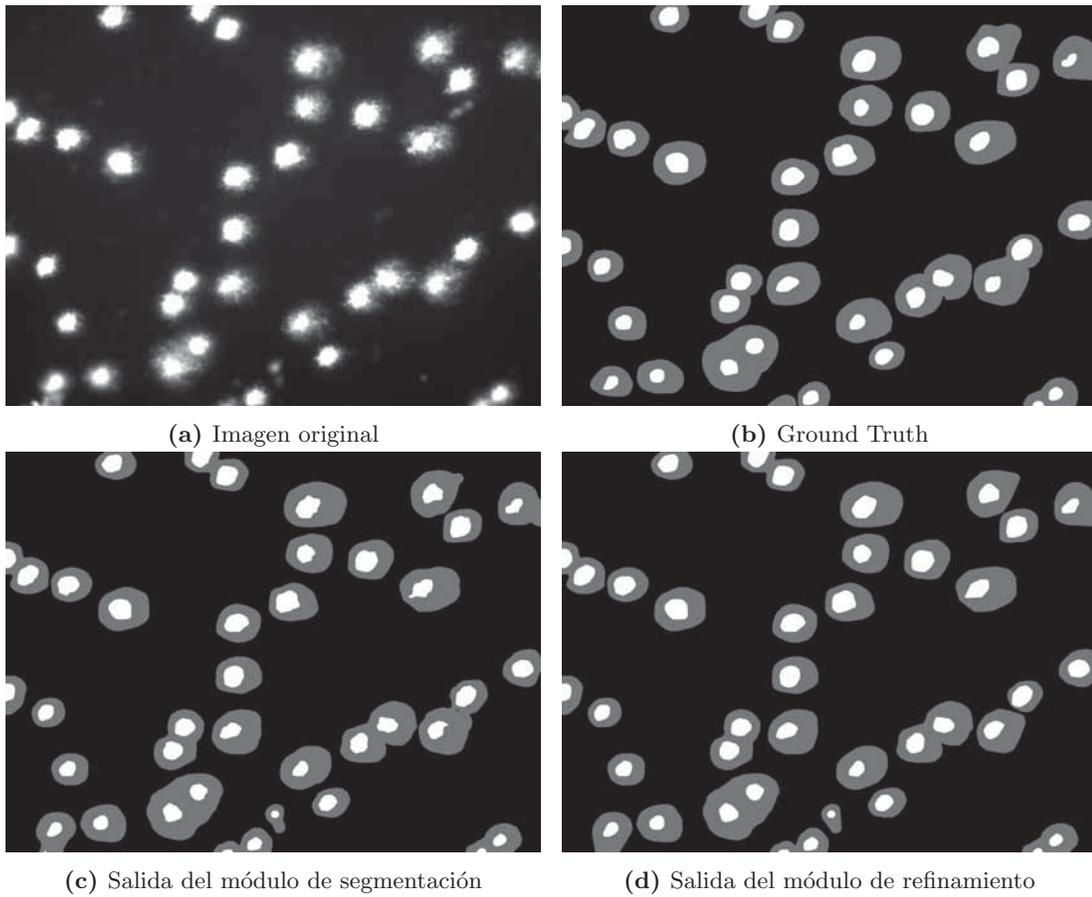


Figura 4.3: Segmentación de la imagen de prueba 2 realizada por los módulos de segmentación y refinamiento

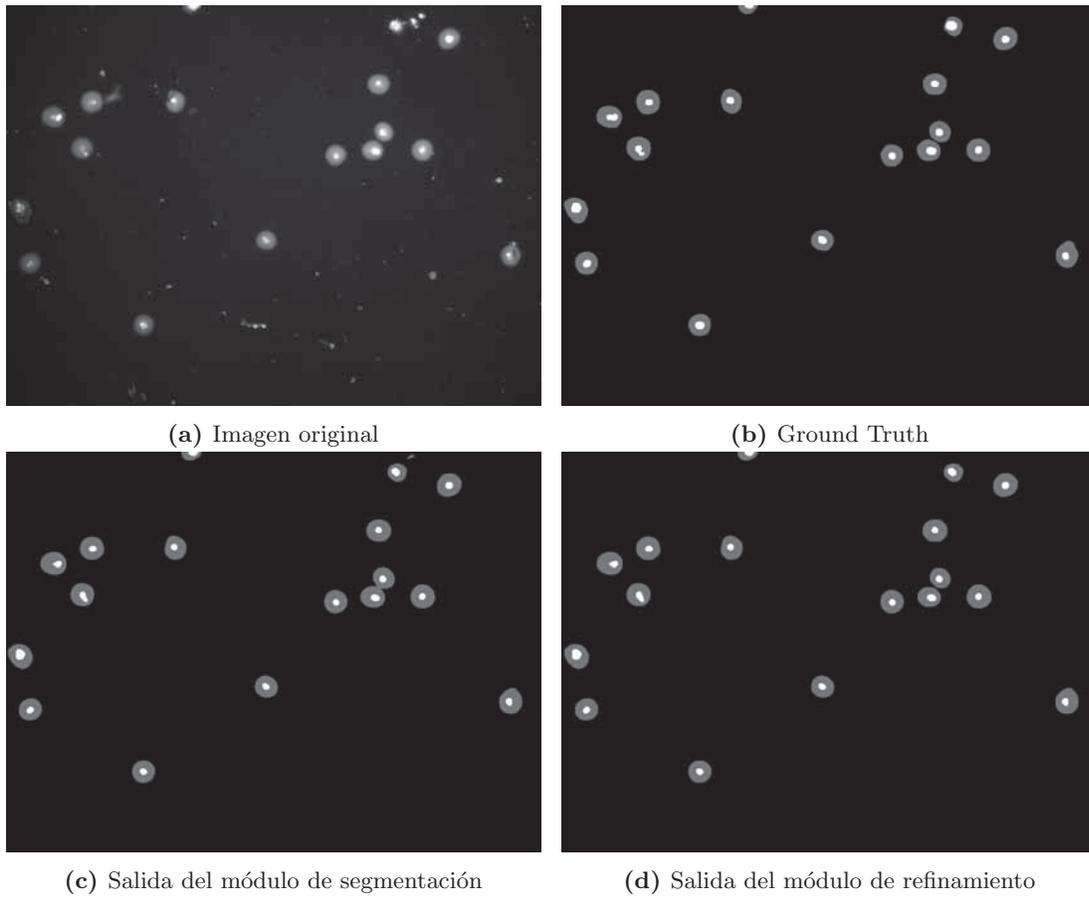


Figura 4.4: Segmentación de la imagen de prueba 3 realizada por los módulos de segmentación y refinamiento

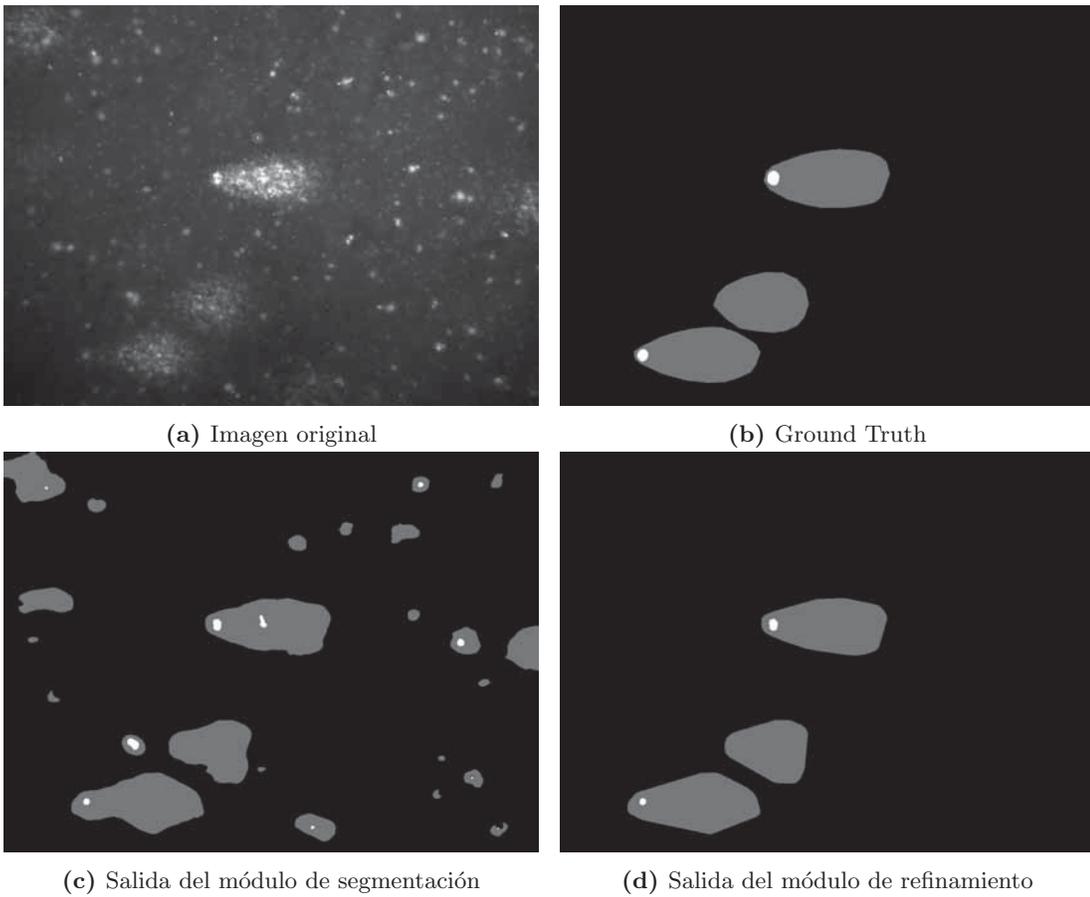


Figura 4.5: Segmentación de la imagen de prueba 4 realizada por los módulos de segmentación y refinamiento

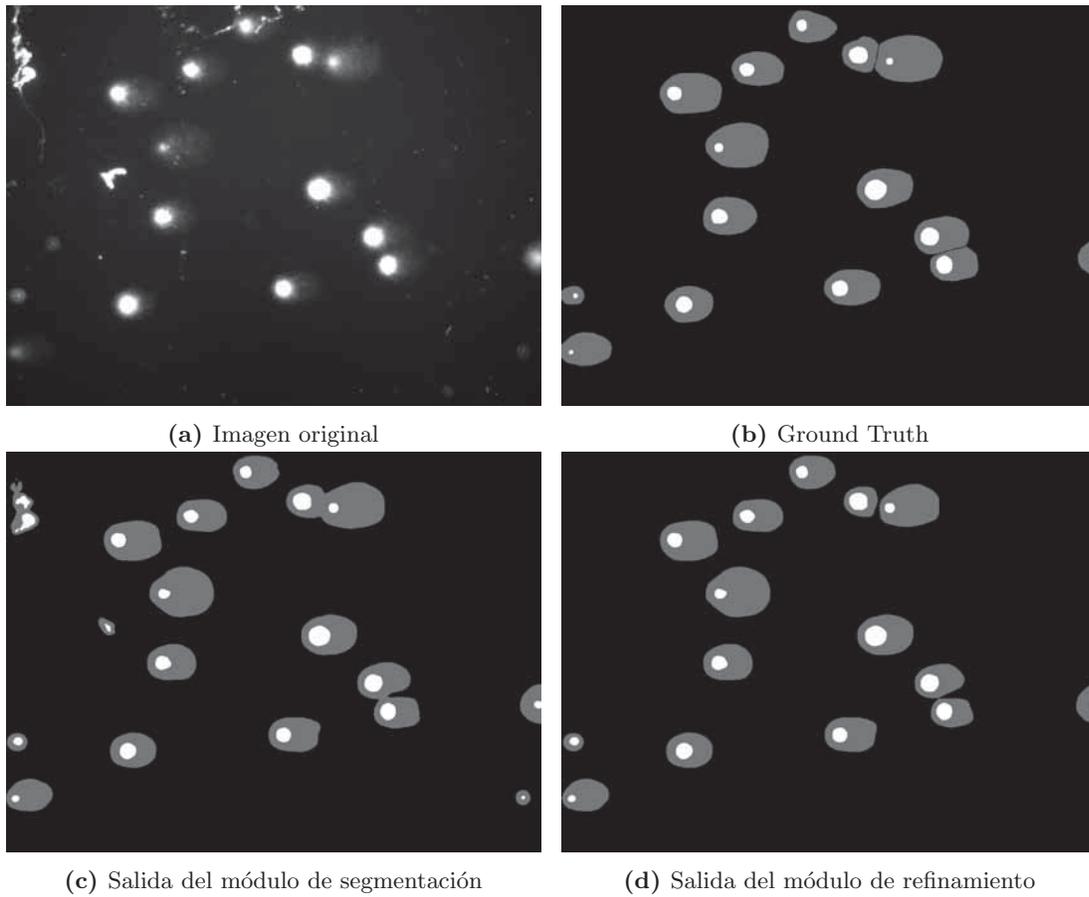


Figura 4.6: Segmentación de la imagen de prueba 5 realizada por los módulos de segmentación y refinamiento

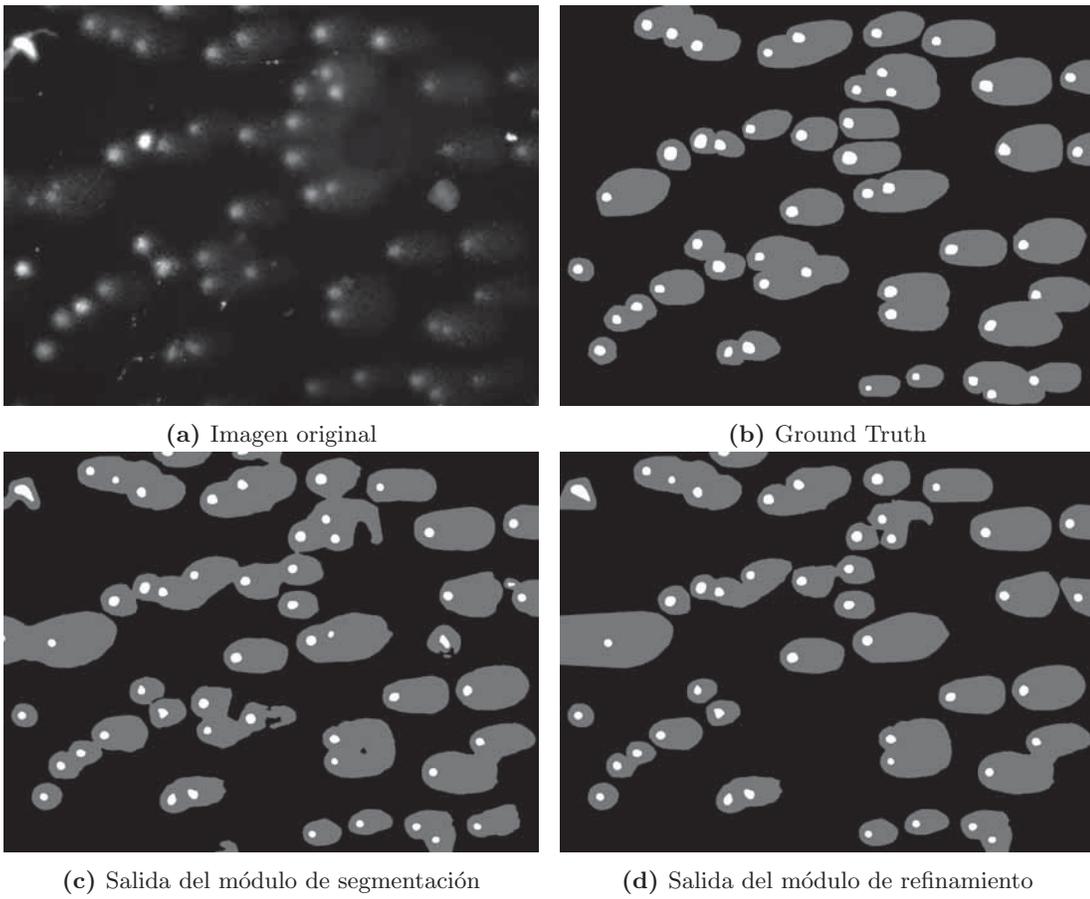


Figura 4.7: Segmentación de la imagen de prueba 6 realizada por los módulos de segmentación y refinamiento

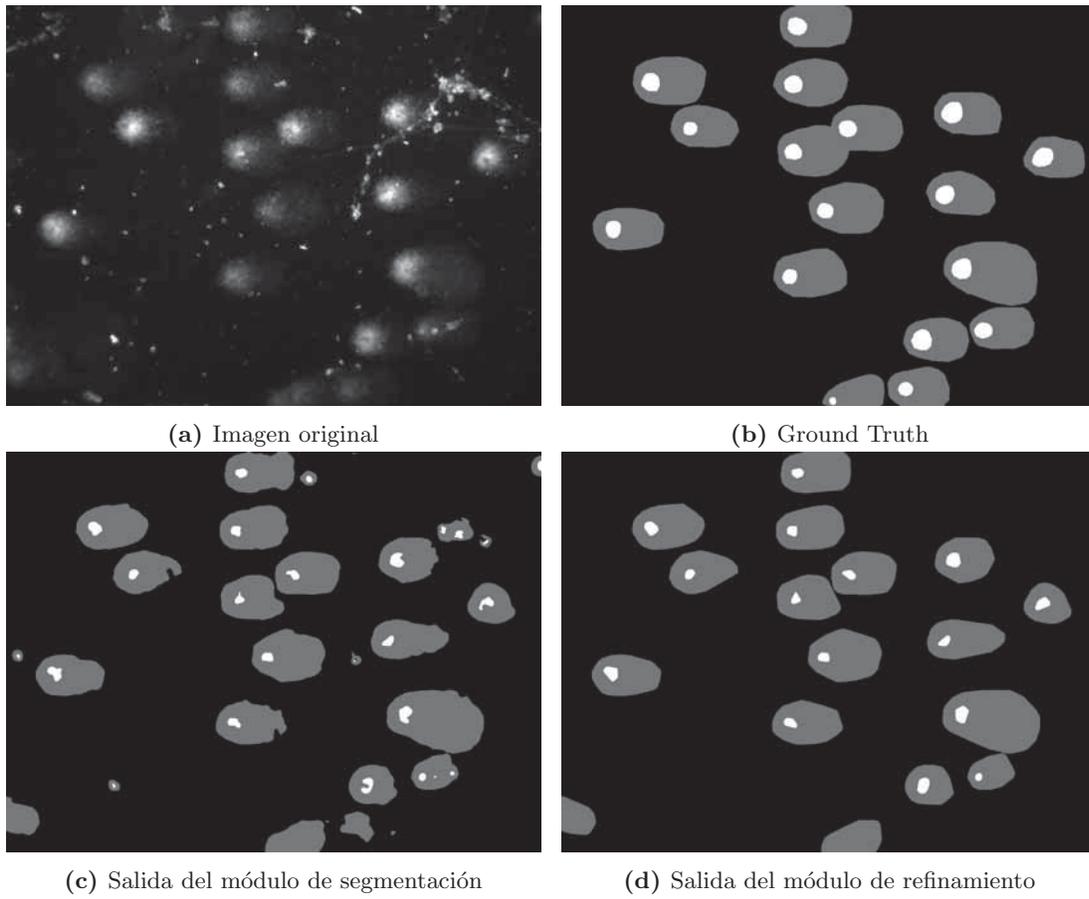


Figura 4.8: Segmentación de la imagen de prueba 7 realizada por los módulos de segmentación y refinamiento

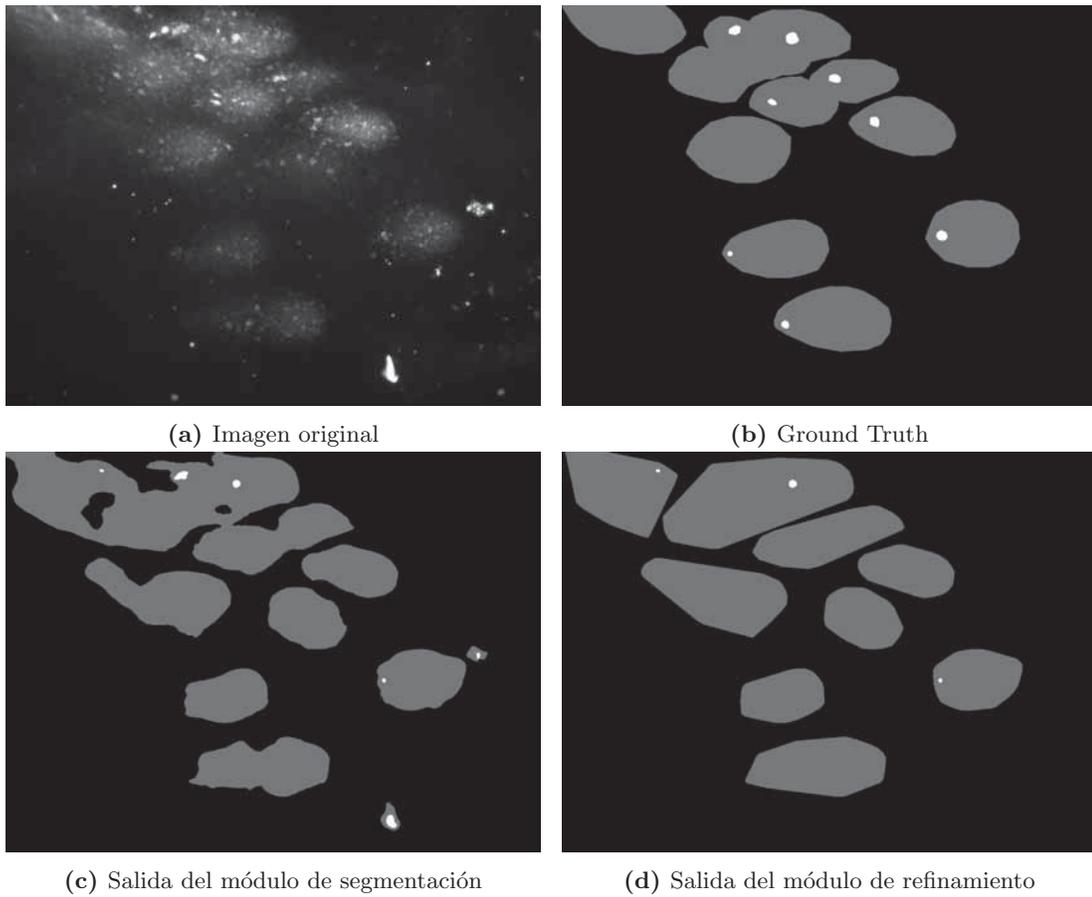


Figura 4.9: Segmentación de la imagen de prueba 8 realizada por los módulos de segmentación y refinamiento

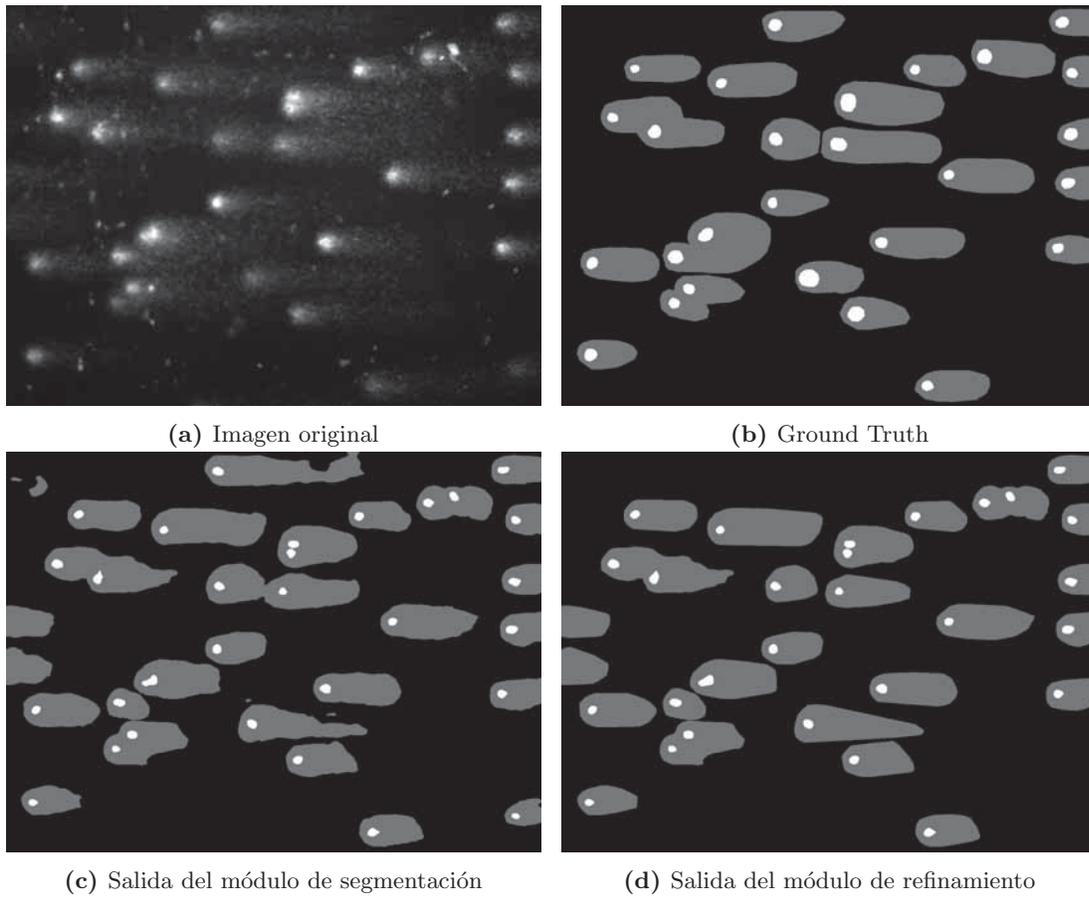


Figura 4.10: Segmentación de la imagen de prueba 9 realizada por los módulos de segmentación y refinamiento

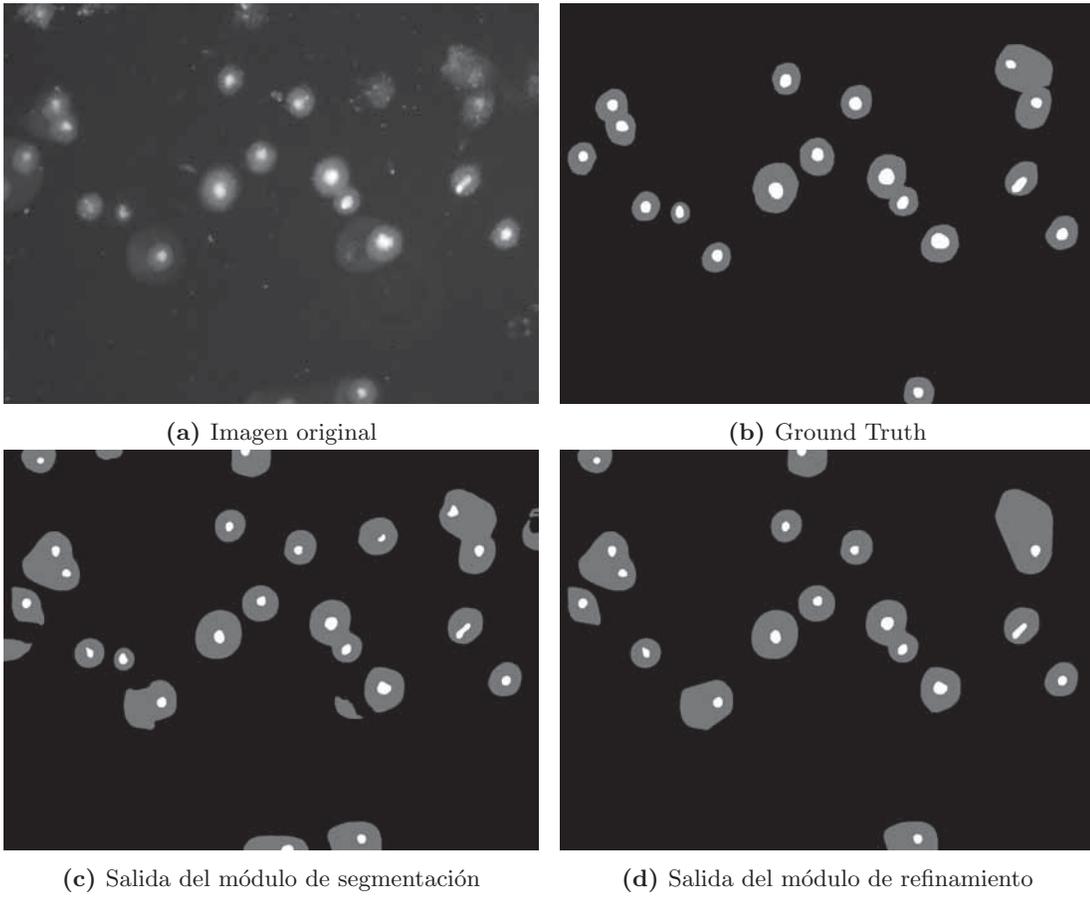


Figura 4.11: Segmentación de la imagen de prueba 10 realizada por los módulos de segmentación y refinamiento

4.3. Comparación de segmentación con otros sistemas computacionales

Para esta sección, se procesaron las 10 imágenes de prueba en dos sistemas computacionales, OpenComet y HiComet, con el fin de obtener las segmentaciones correspondientes y compararlas con el Groun Truth y la segmentación refinada realizada en este proyecto. De acuerdo con las Tabla 4.3, se observa que el sistema propuesto obtuvo mejor desempeño de F1-score para todas las clases, lo que significa que la segmentación realizada por el ensamble es más parecida al Ground Truth que las segmentaciones realizadas por los otros dos sistemas.

Para la clase de fondo, se observa que tanto OpenComet y HiComet tienen una ligera mejor exhaustividad que el sistema propuesto, pero una menor precisión. Lo anterior puede significar que estos dos sistemas segmentan los cometas con una menor área que la real, provocando que píxeles que realmente pertenecen a un cometa se clasifiquen como fondo. En cambio, el sistema propuesto segmenta los cometas con un área más parecida a la real, por lo que no se clasifican muchos píxeles de cometa como fondo, y como consecuencia, aumenta su precisión.

Con relación a la clase cola, se observa que el sistema propuesto obtuvo una exhaustividad mucho mayor a la de los otros dos sistemas, lo que significa que el sistema propuesto identificó más cometas o bien segmentó las cabezas con un tamaño más parecido al real. No obstante, los sistemas de OpenComet y HiComet obtuvieron una leve mayor precisión que el sistema propuesto, que tomando en cuenta la baja exhaustividad de los dos primeros sistemas, implica que son muy estrictos para clasificar un píxel como cola, al contrario del sistema propuesto, que es más equilibrado.

Con respecto a la clase cabeza, se observa que el sistema propuesto obtuvo una mejor precisión y exhaustividad que los otros dos sistemas, sobre todo en precisión. Esto indica que el sistema propuesto identifica más cabezas y con un tamaño más parecido al real que los sistemas de OpenComet y HiComet.

Por último, se muestran las segmentaciones de los tres sistemas para cada una de las 10 imágenes de prueba en las figuras de la 4.12 a la 4.21. En ellas se observa visualmente que la segmentación del sistema propuesto identifica más cometas, y segmenta las cabezas de mejor manera que el OpenComet y HiComet.

Clase	Sistema Propuesto			OpenComet			HiComet		
	P	R	F1	P	R	F1	P	R	F1
Fondo	0.972	0.977	0.975	0.889	0.992	0.937	0.873	0.998	0.931
Cola	0.847	0.840	0.843	0.867	0.206	0.333	0.856	0.152	0.258
Cabeza	0.926	0.714	0.806	0.207	0.448	0.283	0.480	0.596	0.531
Promedio	0.915	0.844	0.875	0.654	0.549	0.518	0.736	0.582	0.574

Tabla 4.3: Comparación del desempeño del sistema propuesto con el OpenComet y HiComet

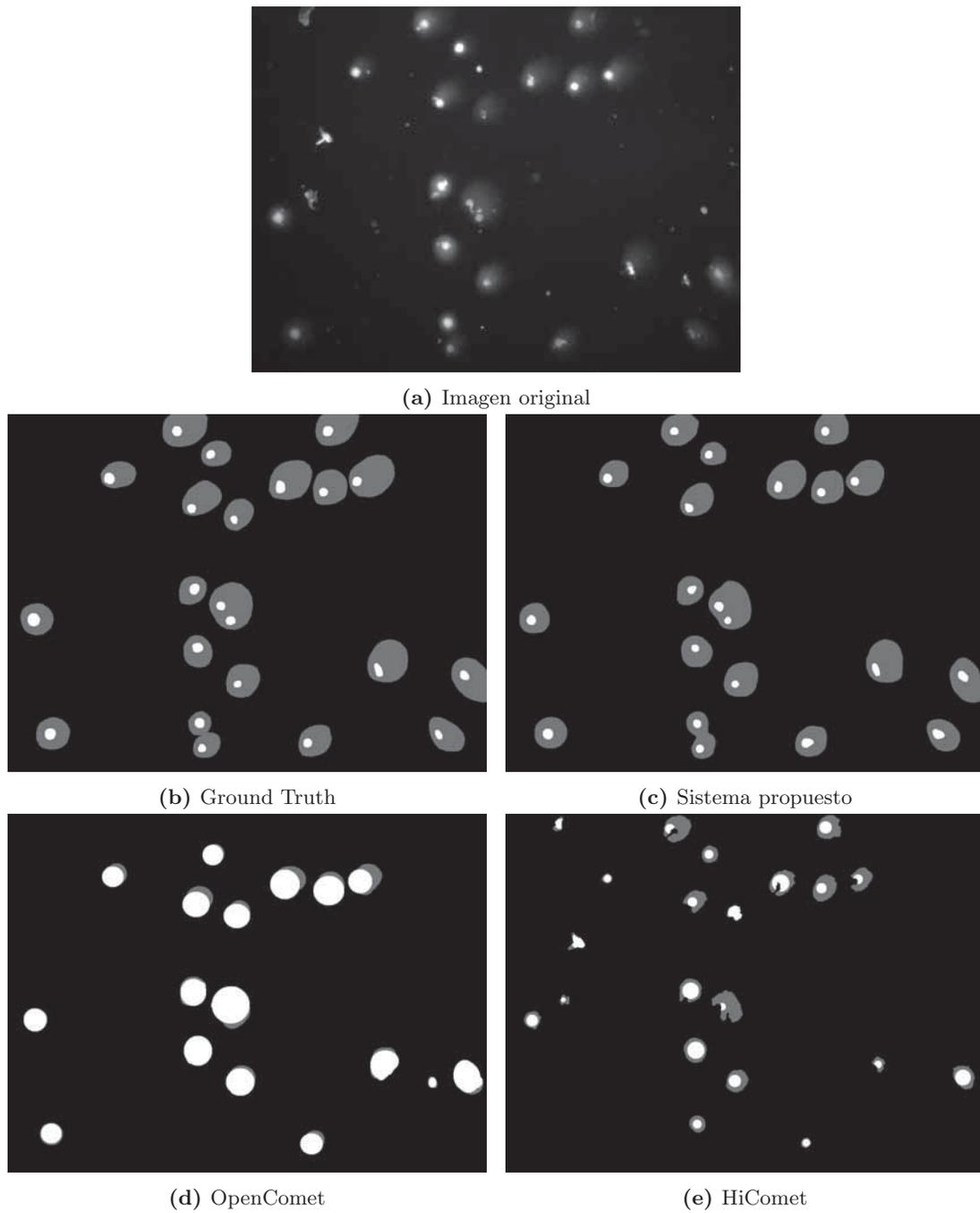
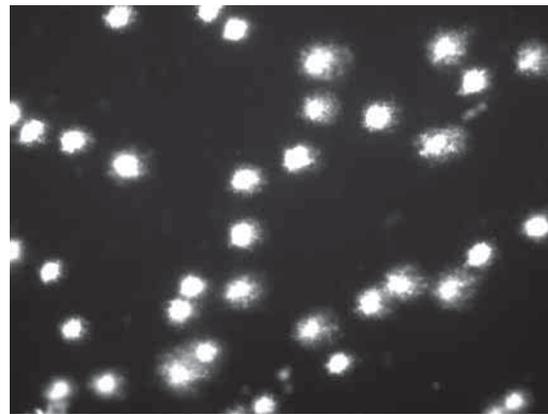
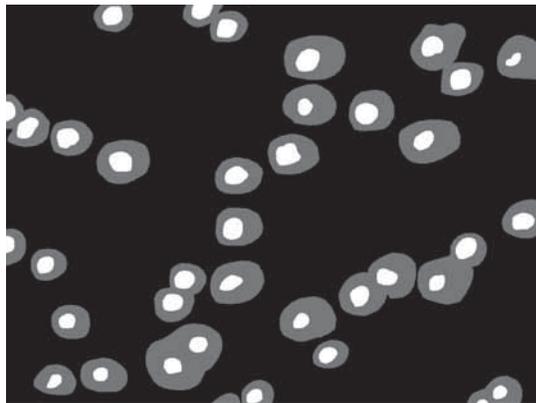


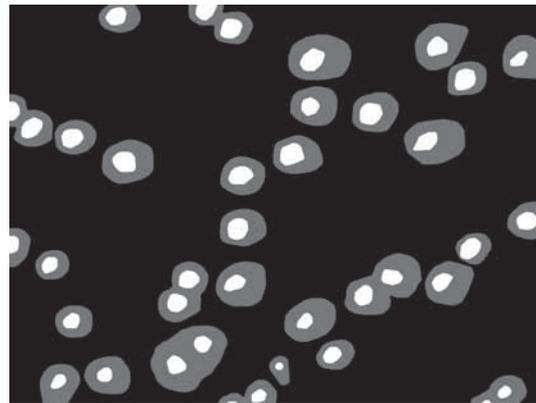
Figura 4.12: Segmentación de la imagen de prueba 1 realizada por el sistema propuesto, OpenComet y HiComet



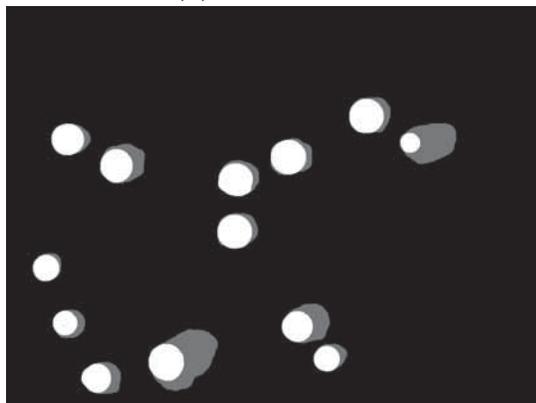
(a) Imagen original



(b) Ground Truth



(c) Sistema propuesto

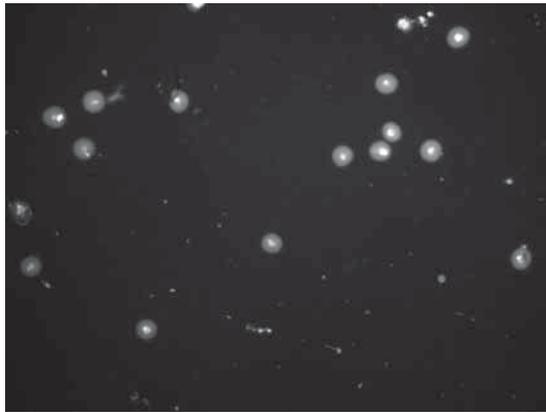


(d) OpenComet

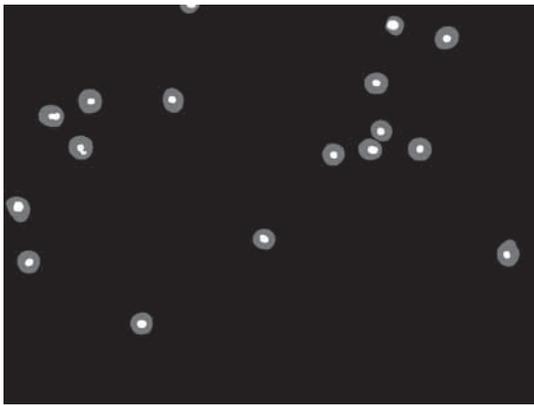


(e) HiComet

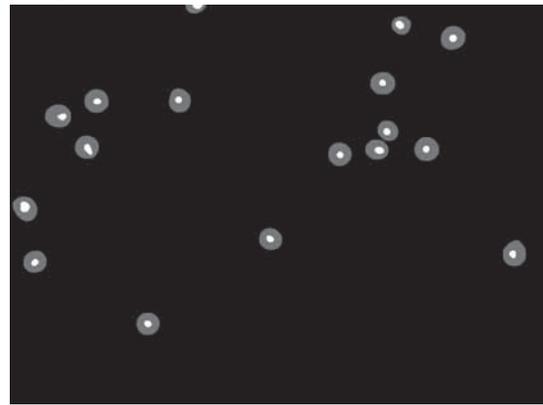
Figura 4.13: Segmentación de la imagen de prueba 2 realizada por el sistema propuesto, OpenComet y HiComet



(a) Imagen original



(b) Ground Truth



(c) Sistema propuesto

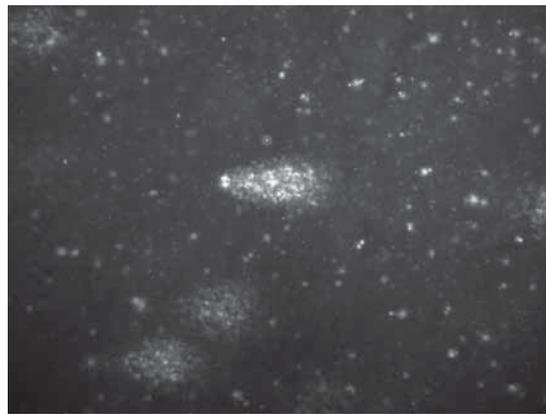


(d) OpenComet

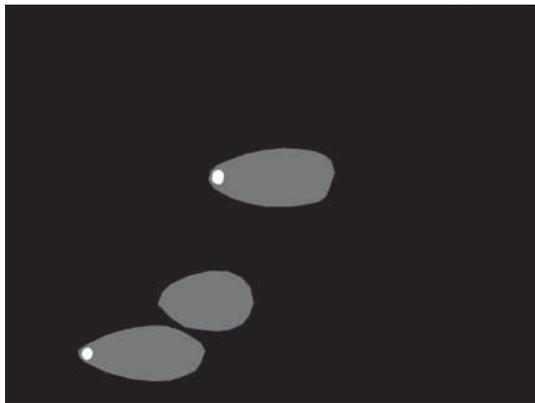


(e) HiComet

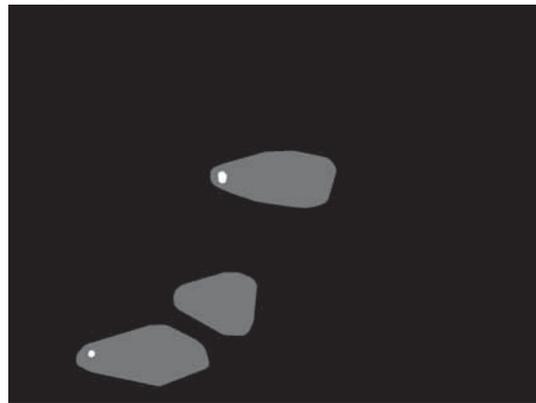
Figura 4.14: Segmentación de la imagen de prueba 3 realizada por el sistema propuesto, OpenComet y HiComet



(a) Imagen original



(b) Ground Truth



(c) Sistema propuesto

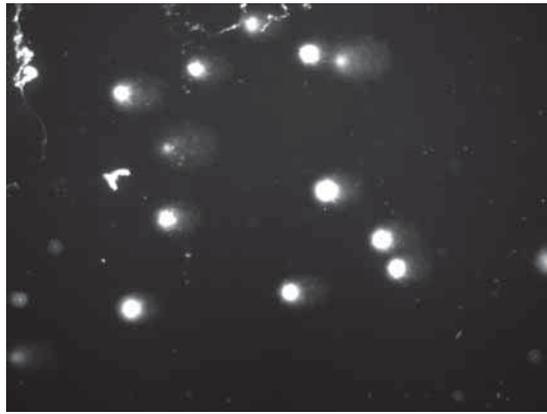


(d) OpenComet

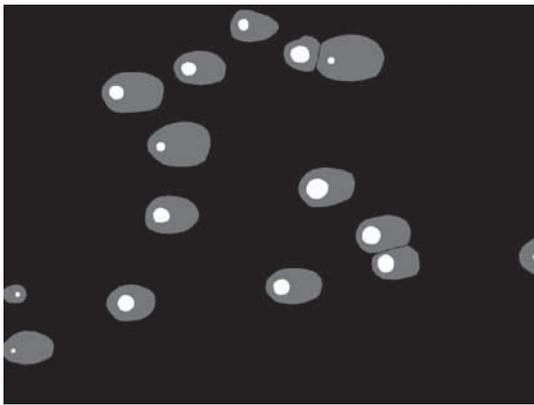


(e) HiComet

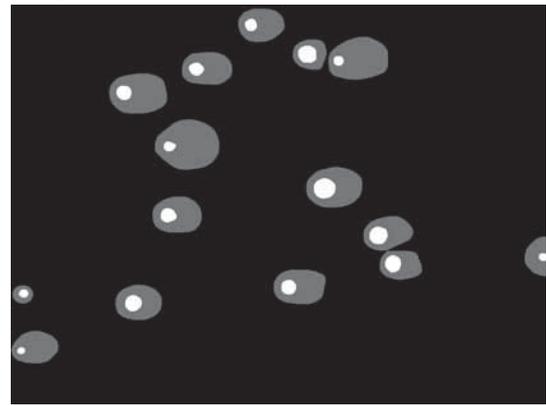
Figura 4.15: Segmentación de la imagen de prueba 4 realizada por el sistema propuesto, OpenComet y HiComet



(a) Imagen original



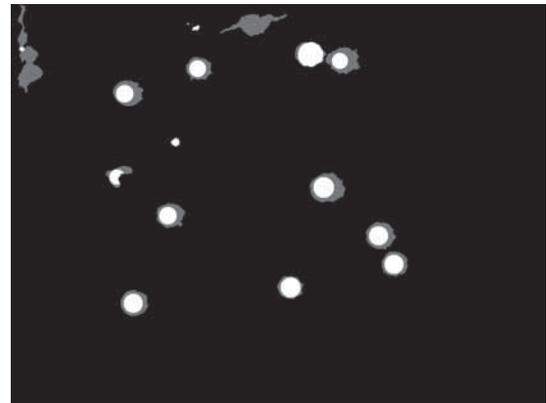
(b) Ground Truth



(c) Sistema propuesto

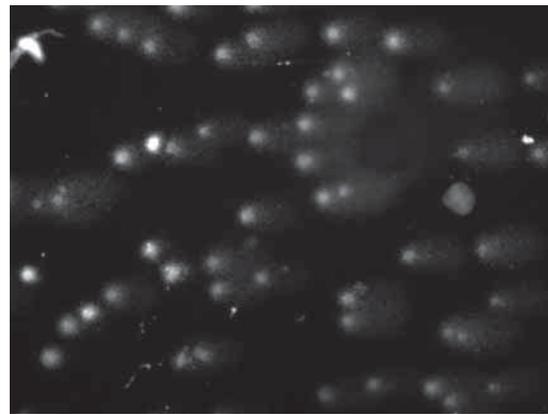


(d) OpenComet

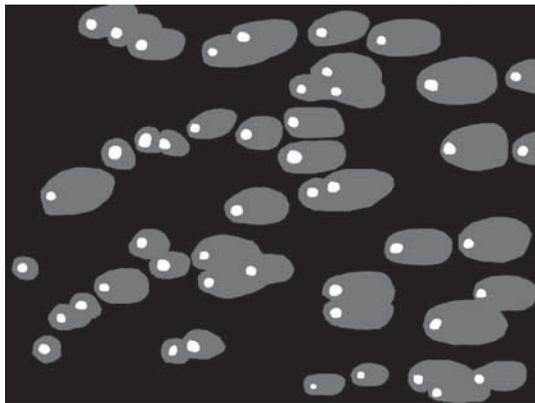


(e) HiComet

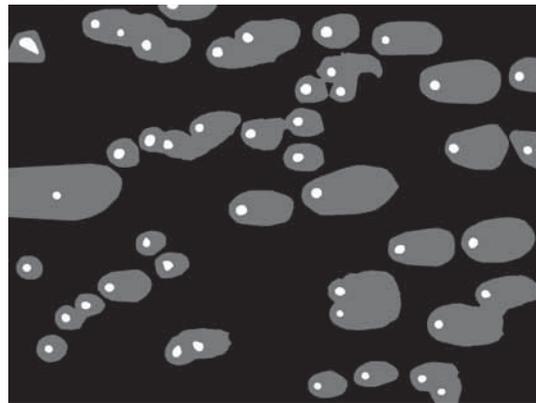
Figura 4.16: Segmentación de la imagen de prueba 5 realizada por el sistema propuesto, OpenComet y HiComet



(a) Imagen original



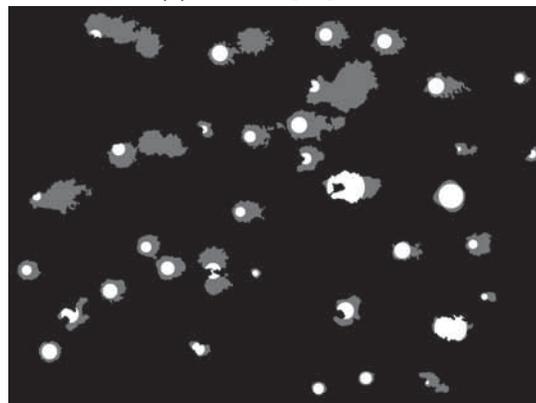
(b) Ground Truth



(c) Sistema propuesto

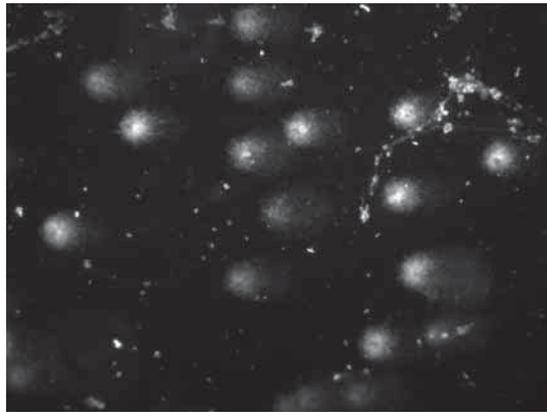


(d) OpenComet

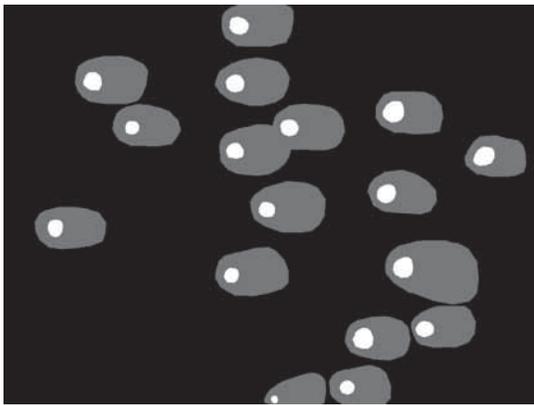


(e) HiComet

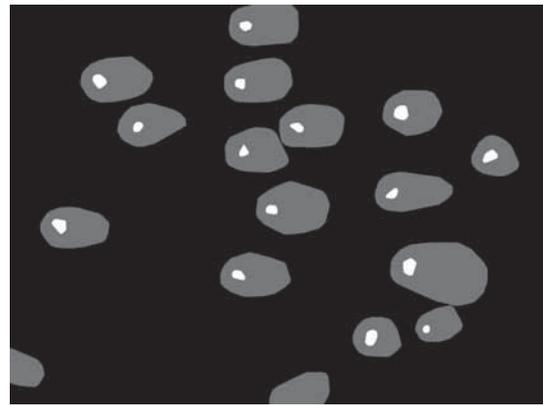
Figura 4.17: Segmentación de la imagen de prueba 6 realizada por el sistema propuesto, OpenComet y HiComet



(a) Imagen original



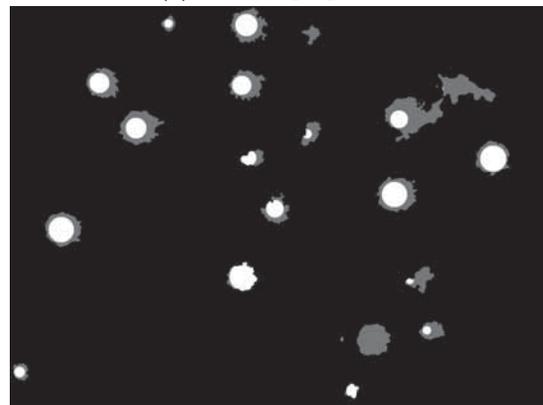
(b) Ground Truth



(c) Sistema propuesto

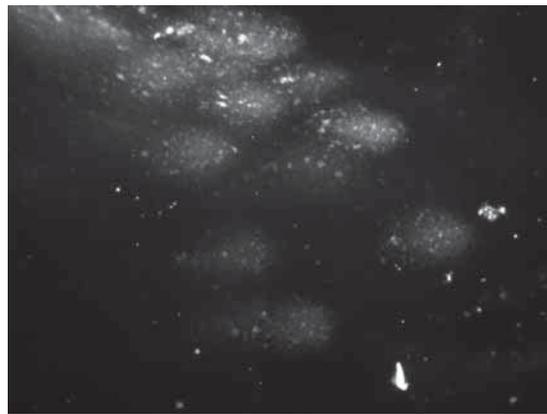


(d) OpenComet

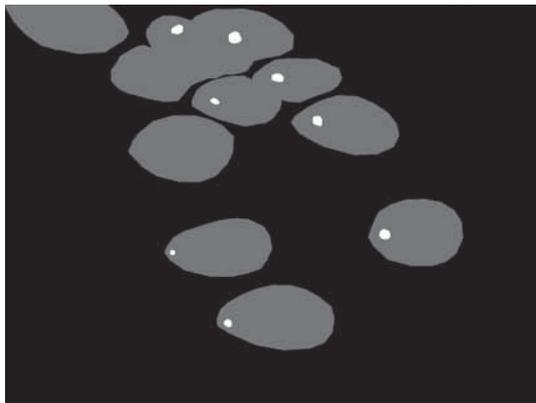


(e) HiComet

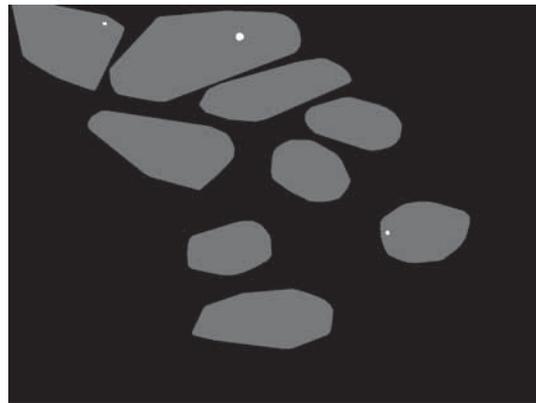
Figura 4.18: Segmentación de la imagen de prueba 7 realizada por el sistema propuesto, OpenComet y HiComet



(a) Imagen original



(b) Ground Truth



(c) Sistema propuesto



(d) OpenComet



(e) HiComet

Figura 4.19: Segmentación de la imagen de prueba 8 realizada por el sistema propuesto, OpenComet y HiComet

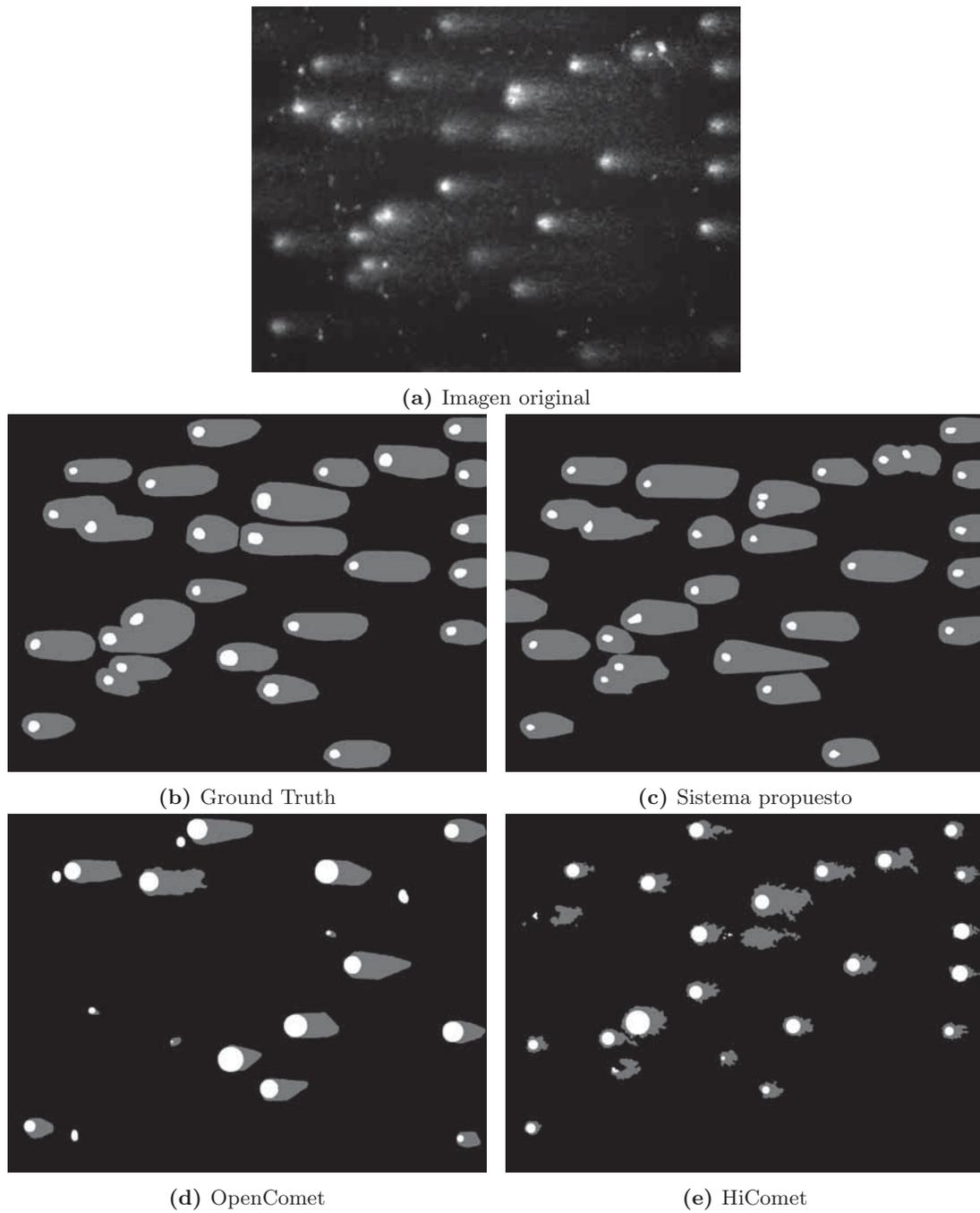
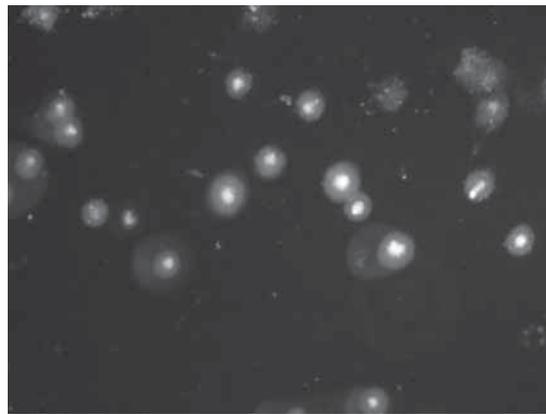
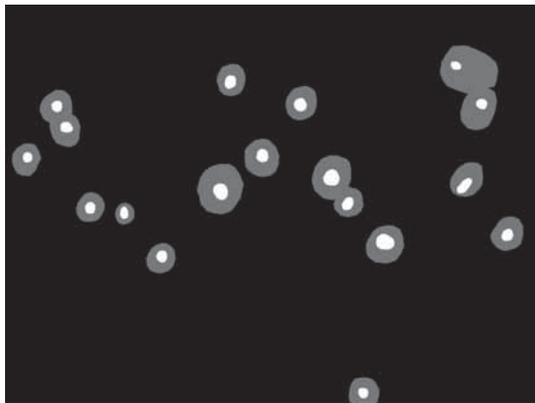


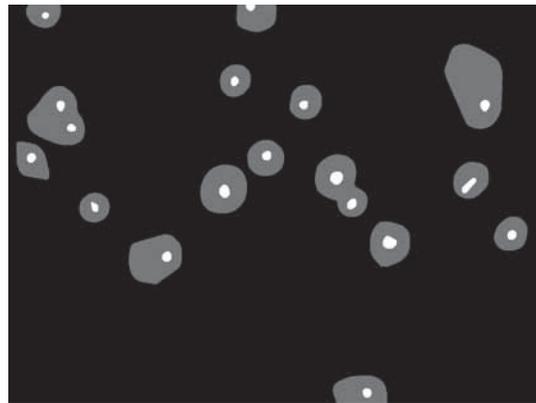
Figura 4.20: Segmentación de la imagen de prueba 9 realizada por el sistema propuesto, OpenComet y HiComet



(a) Imagen original



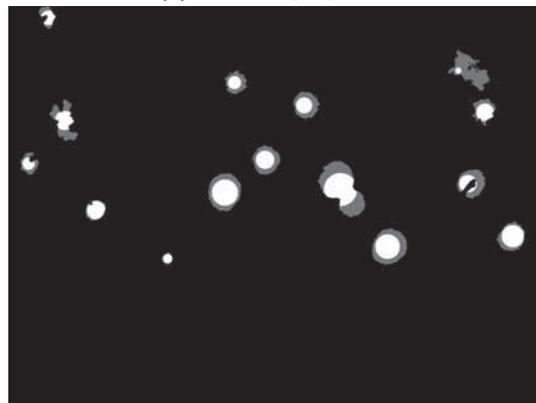
(b) Ground Truth



(c) Sistema propuesto



(d) OpenComet



(e) HiComet

Figura 4.21: Segmentación de la imagen de prueba 10 realizada por el sistema propuesto, OpenComet y HiComet

4.4. Análisis y comparación de los resultados finales

El resultado final se refiere a la imagen de salida procesada por el sistema, la cual consiste en la imagen segmentada y refinada que pasó por los módulos de partición y presentación. En las figuras de la 4.22 a la 4.31 se muestran las imágenes de salida del sistema propuesto, así como las imágenes de salida de los sistemas OpenComet y HiComet.

En la Figura 4.22 se puede visualizar que el sistema propuesto identificó todos los cometas presentes, a excepción de uno que debería estar a la derecha del cometa 8. De igual manera, se observa que se pudo particionar de una manera aceptable los cometas 18,19,20 y 21. Además, no se segmentaron algunos residuos como las regiones 3,11 y 14 del HiComet, lo que indica una buena precisión del sistema propuesto.

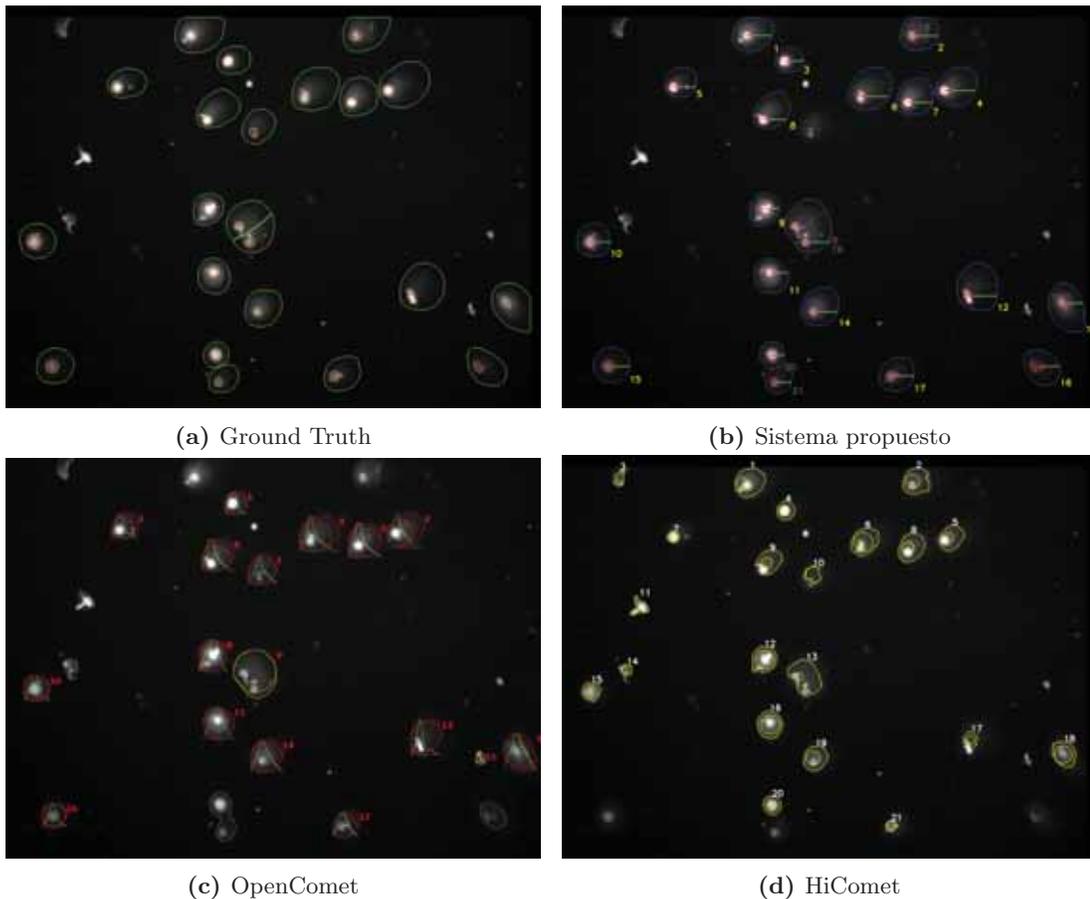


Figura 4.22: Resultados de la imagen de prueba 1 obtenidas del sistema propuesto, OpenComet y HiComet

4. RESULTADOS

De acuerdo con la Figura 4.23, se considera que se obtuvo una buena segmentación y partición en el sistema propuesto, pero se identificó incorrectamente la región 23 como cometa. Al comparar con el OpenComet, se observa que se definieron las cabezas demasiado grandes, lo cual no pasa en el sistema propuesto y en el HiComet, aunque este último falla en la identificación de cabezas de los cometas traslapados.

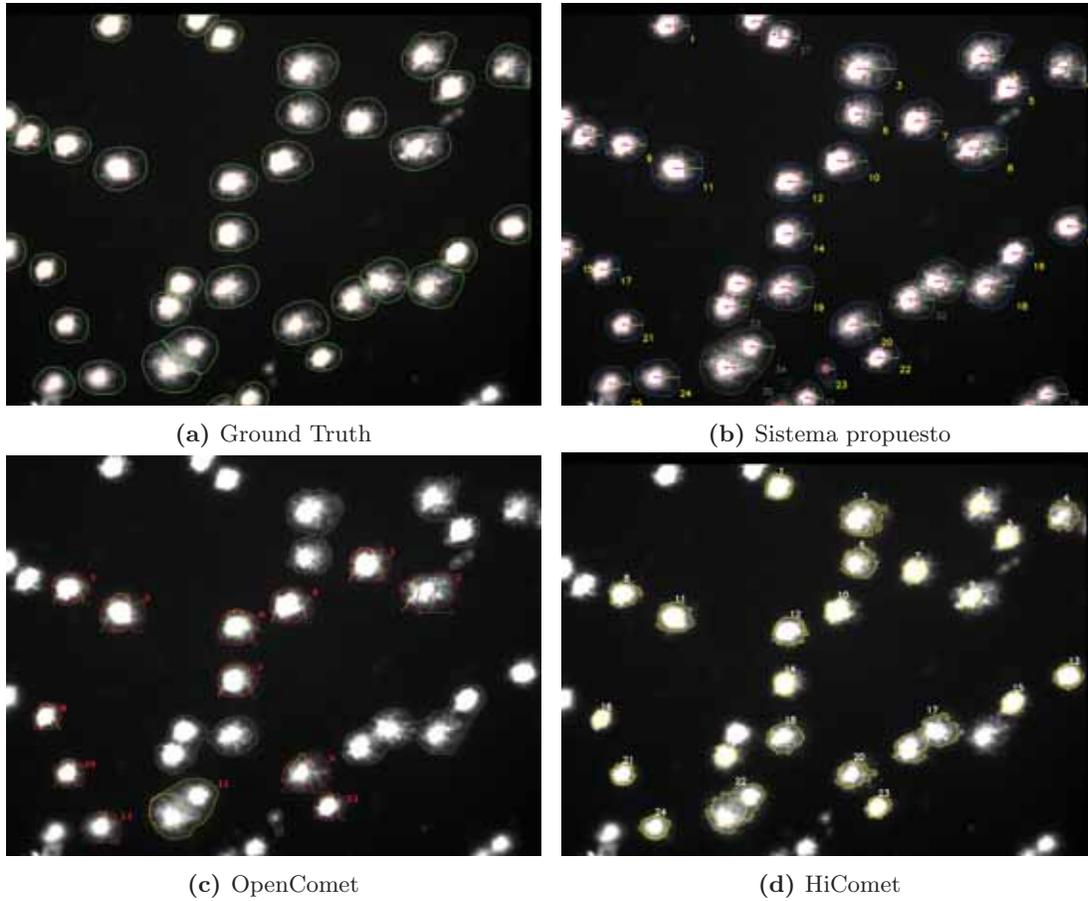


Figura 4.23: Resultados de la imagen de prueba 2 obtenidas del sistema propuesto, OpenComet y HiComet

Con respecto a la Figura 4.24, se puede ver que los tres sistemas fueron capaces de identificar los cometas contenidos en la imagen, sin embargo, la segmentación de cabezas fue mejor en el sistema propuesto.

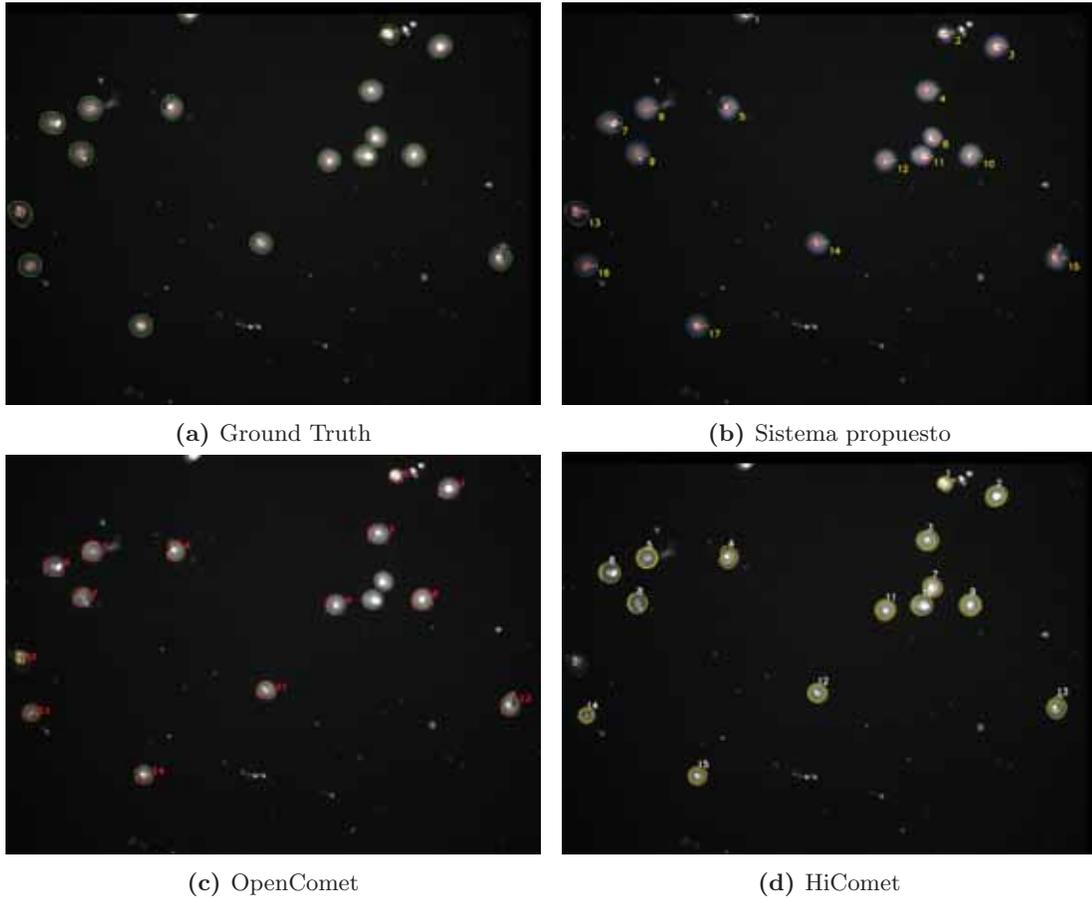


Figura 4.24: Resultados de la imagen de prueba 3 obtenidas del sistema propuesto, OpenComet y HiComet

4. RESULTADOS

En relación con la Figura 4.25, se observa una gran cantidad de falsos positivos en el resultado del OpenComet, mientras que en el sistema propuesto no, puesto que fueron removidos en el refinamiento. De parte de HiComet sólo se identificó un cometa y su cabeza es demasiado grande con respecto a la real.

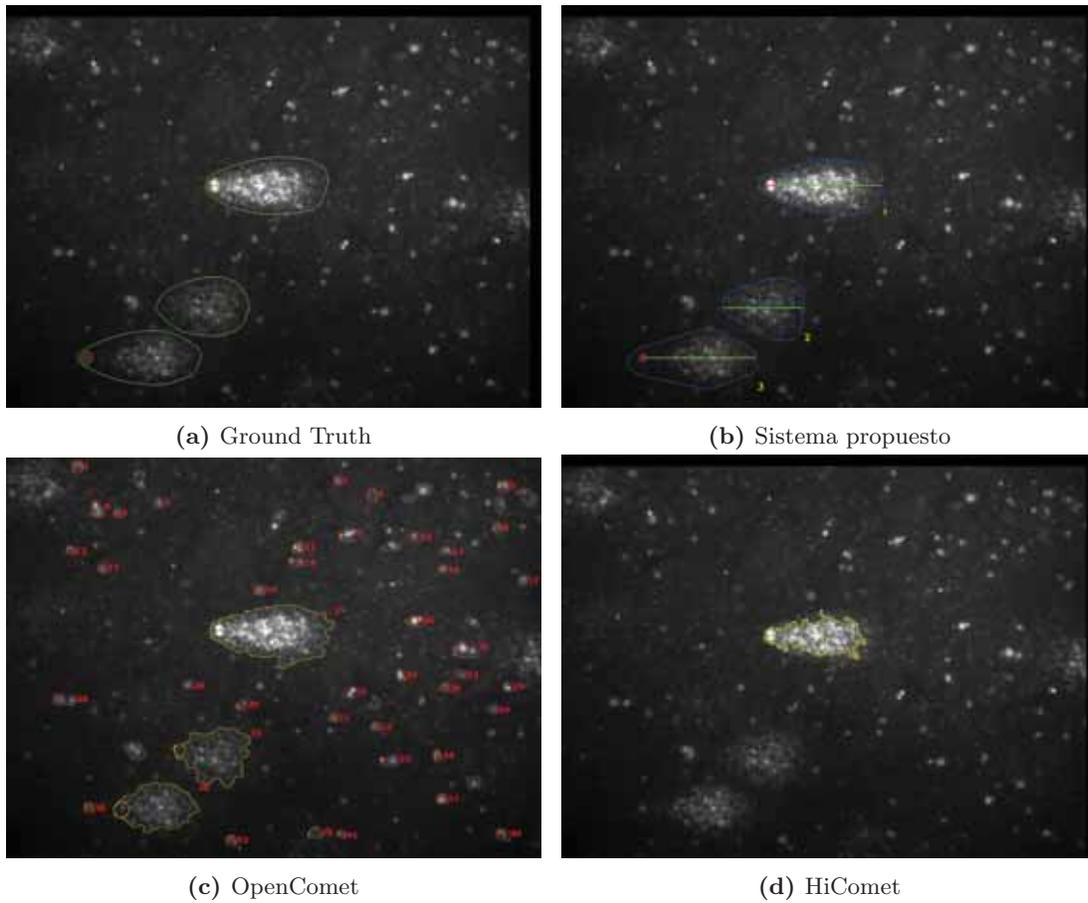


Figura 4.25: Resultados de la imagen de prueba 4 obtenidas del sistema propuesto, OpenComet y HiComet

Para la Figura 4.26, se realizaron buenas segmentaciones en el sistema propuesto y OpenComet, aunque este último identifica erróneamente las regiones 5 y 8 como cometas. En cambio, HiComet, identifica los cometas más pequeños de lo que en realidad son.

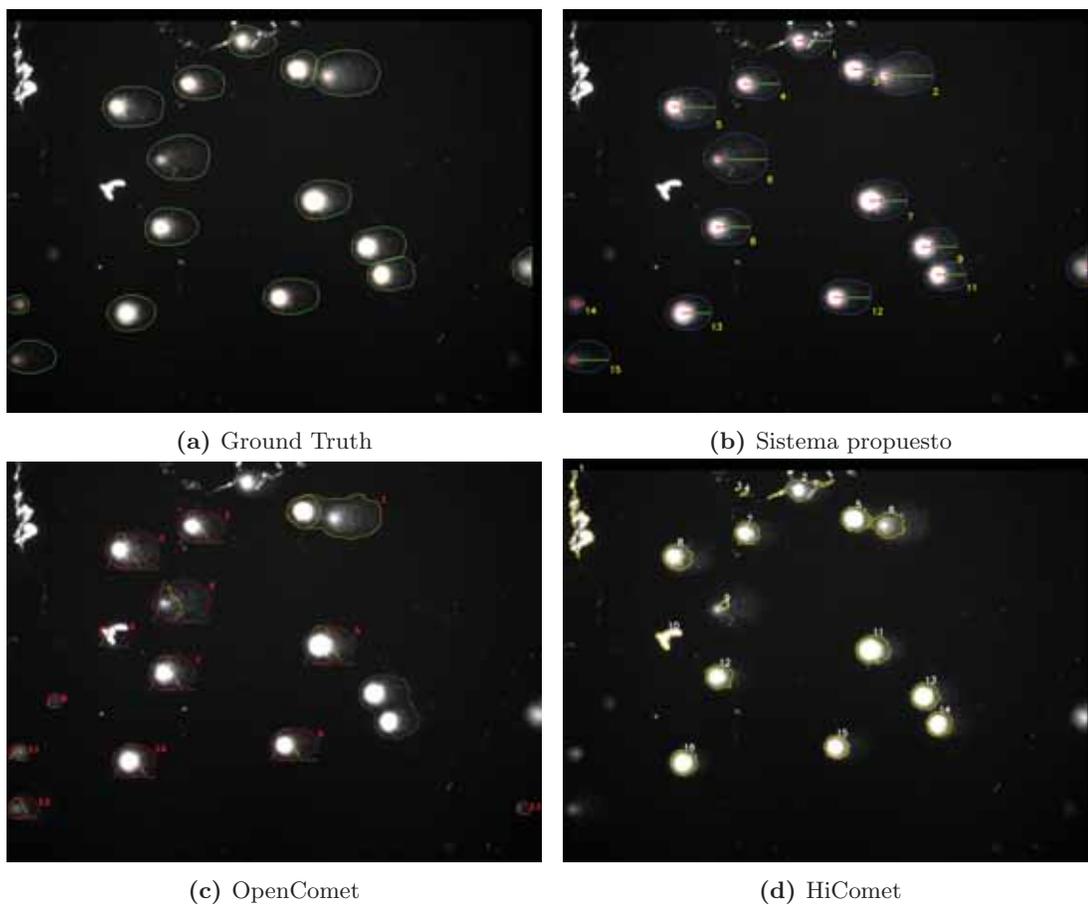


Figura 4.26: Resultados de la imagen de prueba 5 obtenidas del sistema propuesto, OpenComet y HiComet

4. RESULTADOS

En cuanto a la Figura 4.27, se observa que la partición de los cometas traslapados en el sistema propuesta no es buena, sobre todo para los cometas 31 y 33. Además, los cometas debajo del cometa 13, no fueron identificados. Pese a esto, se considera que se obtuvo una mejor identificación de cometas en el sistema propuesto que en los otros dos sistemas

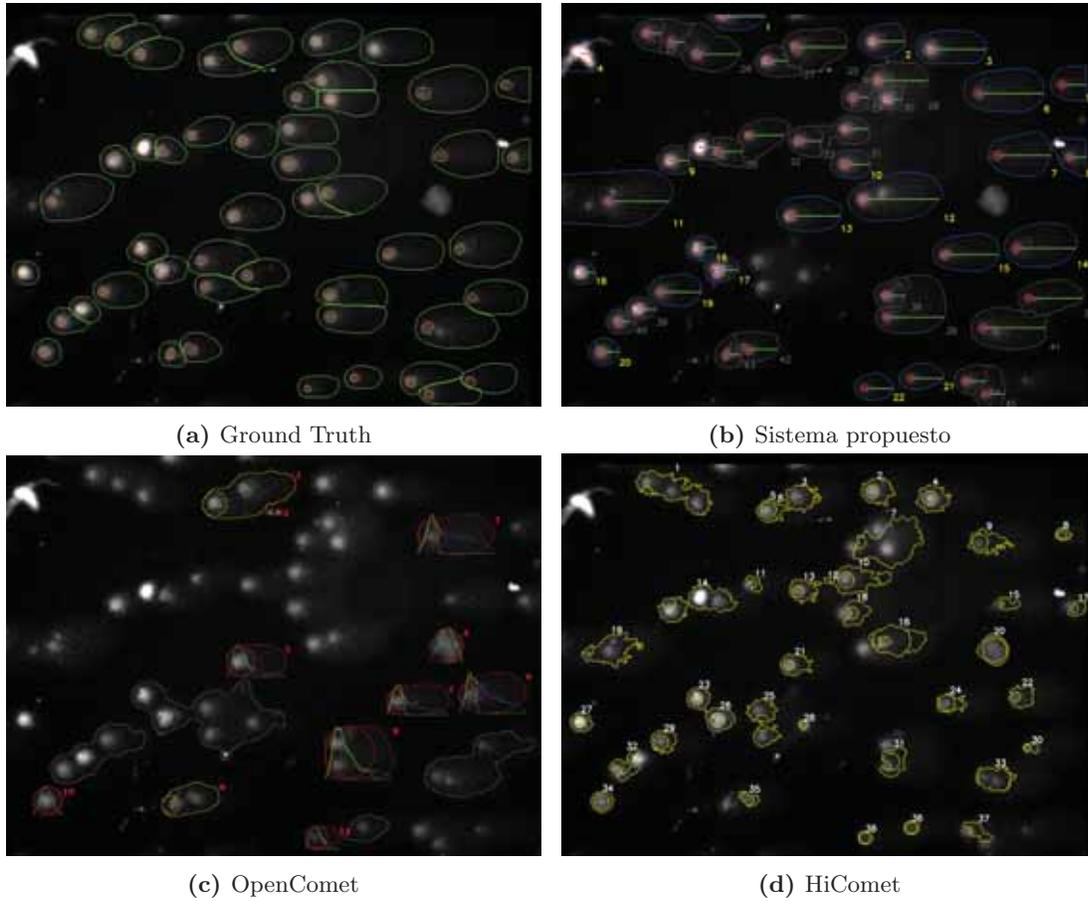


Figura 4.27: Resultados de la imagen de prueba 6 obtenidas del sistema propuesto, OpenComet y HiComet

En la Figura 4.28, se presenta una imagen del ensayo cometa con muchos residuos que no son cometas. En el sistema propuesto se evita satisfactoriamente identificar estos residuos como cometas, mientras que en los otros dos sistemas no. No obstante, el sistema propuesto identificó incorrectamente la región 16 como cometa y no identificó el cometa a la derecha del cometa 17.

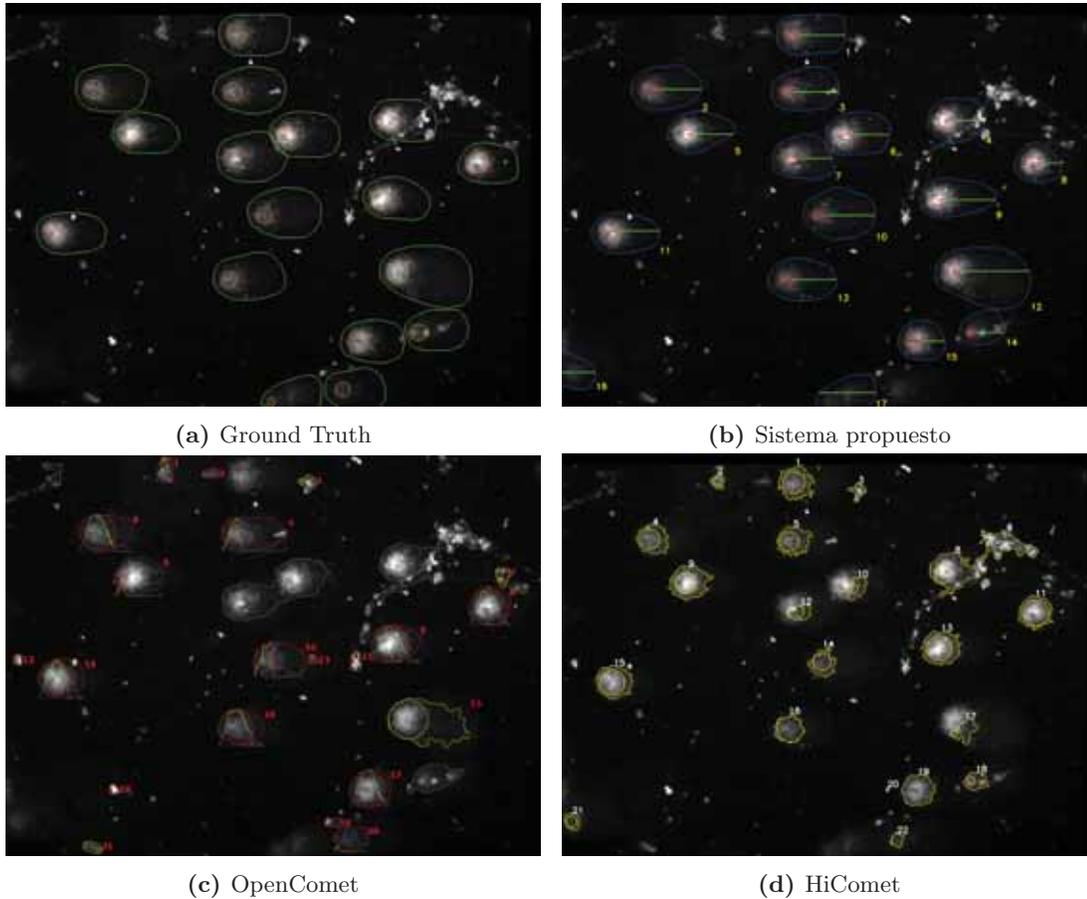


Figura 4.28: Resultados de la imagen de prueba 7 obtenidas del sistema propuesto, OpenComet y HiComet

4. RESULTADOS

La Figura 4.29 representa la imagen del conjunto de prueba más difícil. En el caso de HiComet, no se identifica ningún cometa contenido en la imagen. En el OpenComet se identifican 7 regiones como cometas, aunque 4 de ellos son falsos positivos. Con respecto al sistema propuesto, se identifican varios cometas, aunque la exhaustividad de las cabezas es muy baja, ya que no identifica la mayoría de ellas. Además, como consecuencia de lo anterior, el sistema propuesto no particiona los cometas traslapados, por lo que se obtienen cometas con áreas muy grandes.

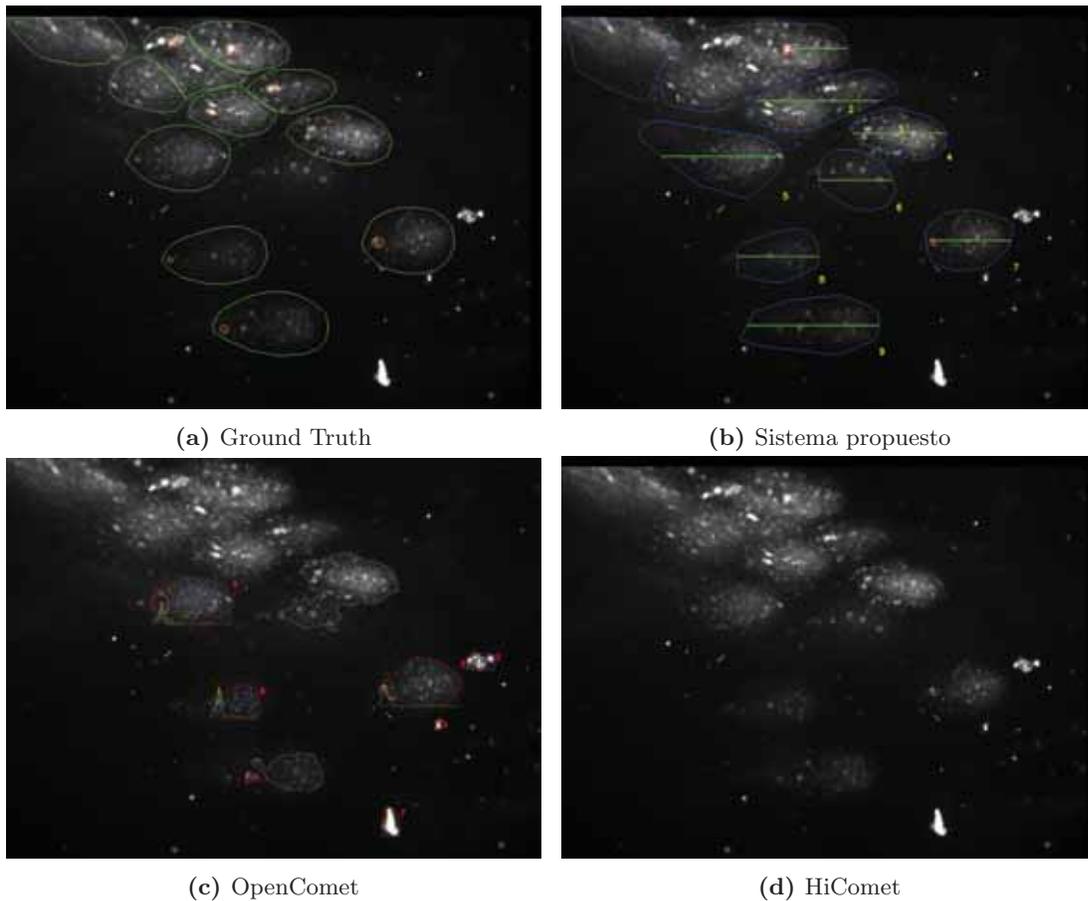


Figura 4.29: Resultados de la imagen de prueba 8 obtenidas del sistema propuesto, OpenComet y HiComet

En la penúltima imagen de prueba, la Figura 4.30, se observa que el sistema propuesto pudo identificar la mayoría de los cometas, aunque nuevamente falla en la partición de los cometas traslapados, como es el caso de los cometas 27,28,29 y 30. El OpenComet identifica menos cometas que el sistema propuesto, puesto que en la imagen hay muchos cometas traslapados, y este los rechaza en su procesamiento. En el caso del HiComet, se identifican más cometas que el OpenComet, pero con un tamaño menor al real.

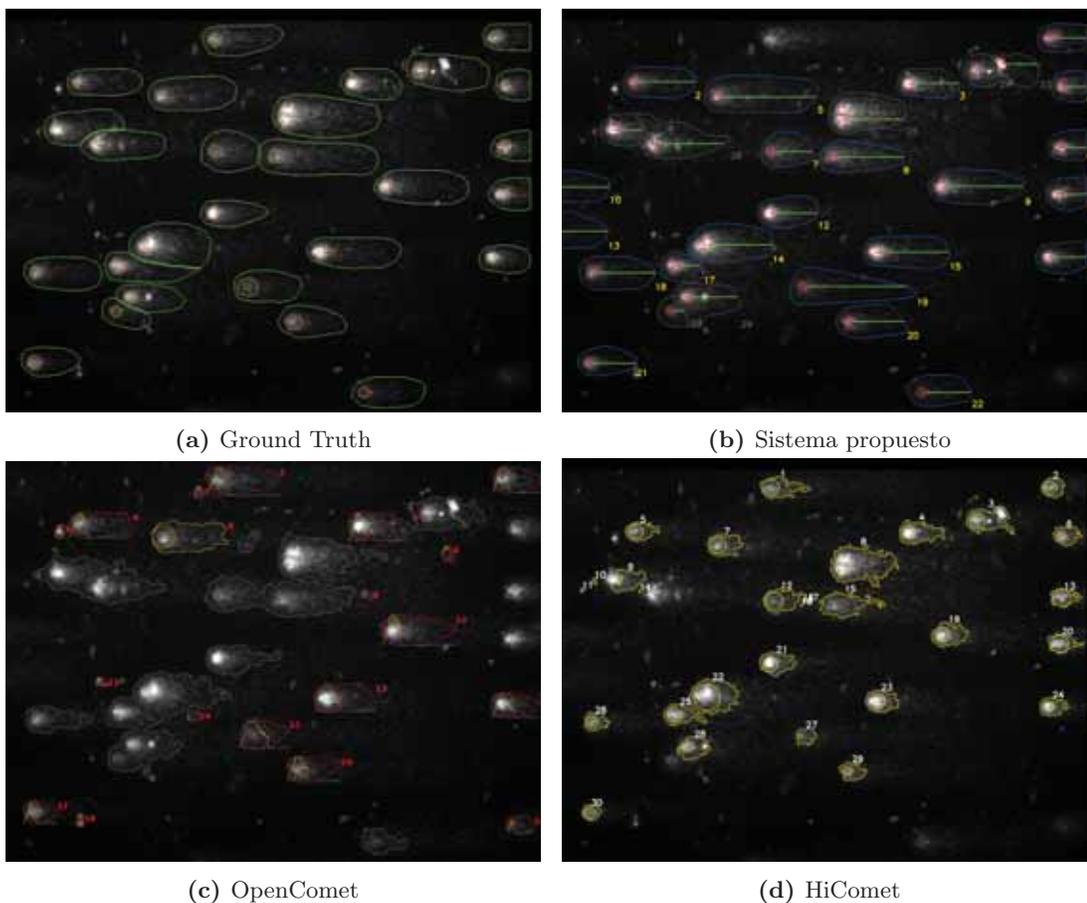


Figura 4.30: Resultados de la imagen de prueba 9 obtenidas del sistema propuesto, OpenComet y HiComet

4. RESULTADOS

Finalmente, en la Figura 4.31, se visualizar que los tres sistemas identifican la mayoría de los cometas de la imagen, sin embargo, la identificación de las cabezas en el sistema propuesto está más cerca del ground truth. Adicionalmente, se observa en el sistema propuesto, que se tiene una partición aceptable para los cometas 17 y 18. Sin embargo, la región número 3 no puede ser particionada en dos cometas, puesto que no se identificó otra cabeza con una probabilidad alta.

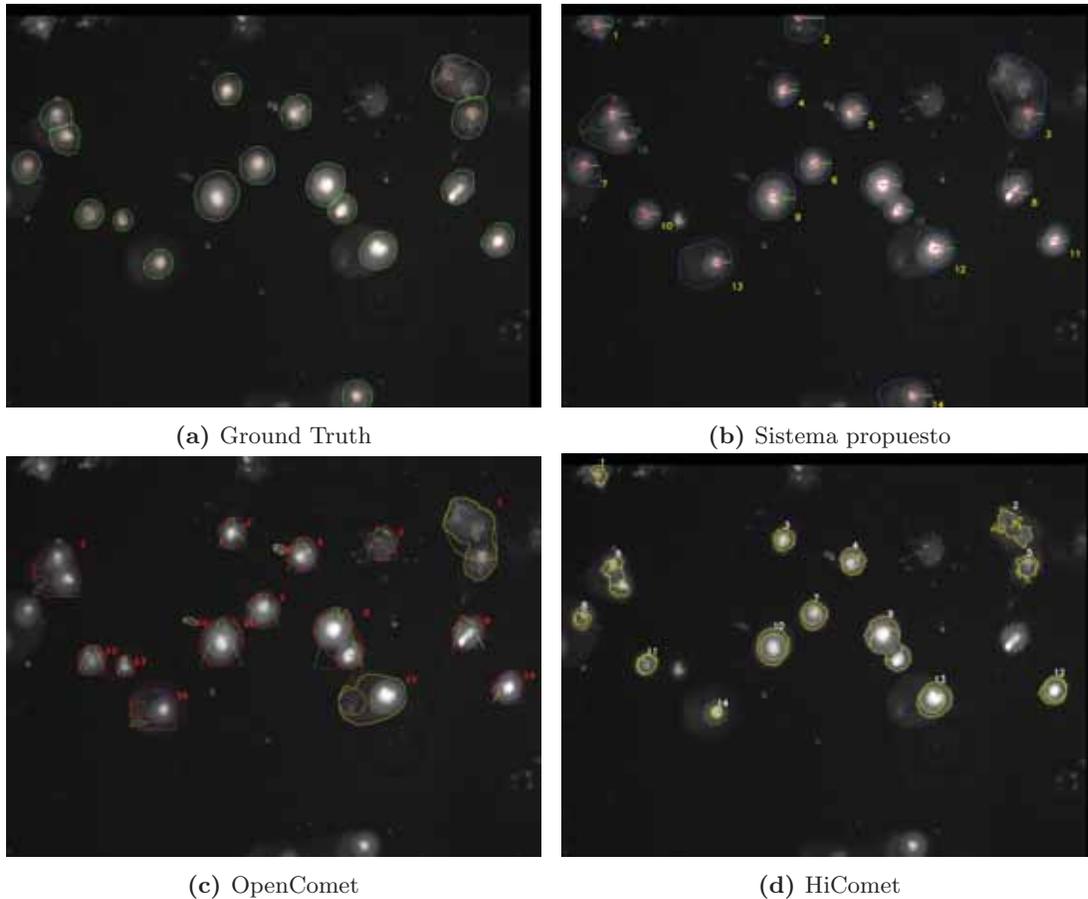


Figura 4.31: Resultados de la imagen de prueba 10 obtenidas del sistema propuesto, OpenComet y HiComet

4.5. Análisis y comparación de los parámetros calculados

Para determinar el desempeño de extracción de características del sistema propuesto, se realizó un cálculo de parámetros sobre ground truth de las 10 imágenes de prueba. Posteriormente, se representó mediante diagramas de caja los parámetros calculados del sistema propuesto y del ground truth. De manera similar, se agregaron los diagramas de caja pertenecientes a los parámetros calculados por los sistemas OpenComet y HiComet, con el fin de realizar una comparación con el sistema propuesto.

Los diagramas de caja, los cuales representan las distribución de los 16 parámetros calculados por los sistemas, se muestran de la Figura 4.32 a la Figura 4.47. Para comparar el desempeño, se debe observar si localización del máximo, del mínimo, de la mediana (línea naranja) y de la media (punto negro), así como el tamaño y ubicación de la caja, coinciden o están cerca de las posiciones respectivas en el diagrama de caja del ground truth.

Generalmente, se observa en las figuras de la 4.32 a la 4.47, que la distribución de los parámetros calculados por el sistema propuesto es parecida a la distribución de los parámetros reales, y que además, muestra un mejor desempeño que los sistemas OpenComet y HiComet. No obstante, se observan valores atípicos muy grandes en algunos parámetros como área, contenido de ADN y largo del cometa. Lo anterior, posiblemente se deba a una región de cometas traslapados que se identificó como un único cometa, provocando que su tamaño y largo sea más grande de lo real.

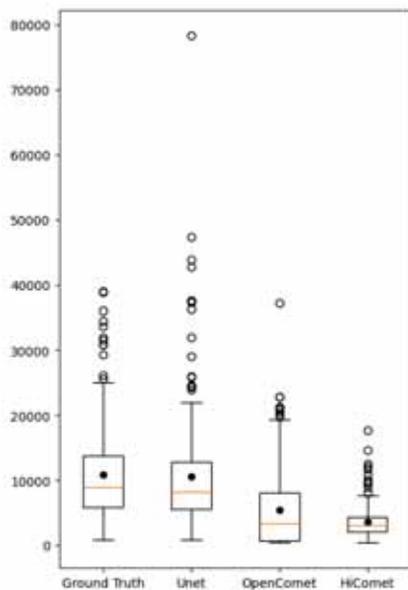


Figura 4.32: Diagrama de caja del área del cometa

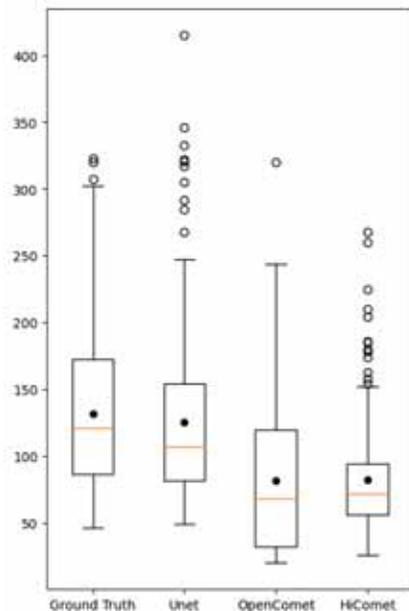


Figura 4.33: Diagrama de caja del largo del cometa

4. RESULTADOS

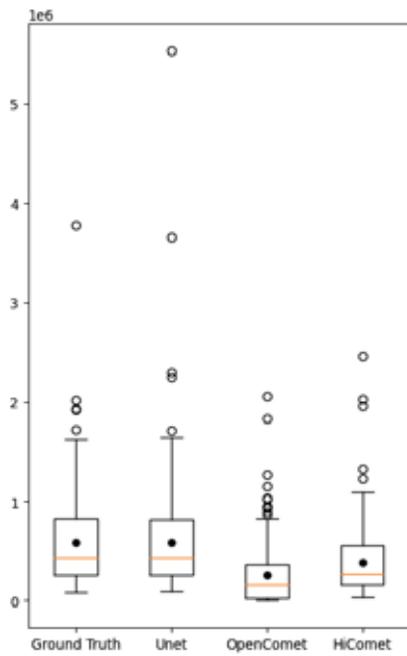


Figura 4.34: Diagrama de caja del contenido de ADN en el cometa

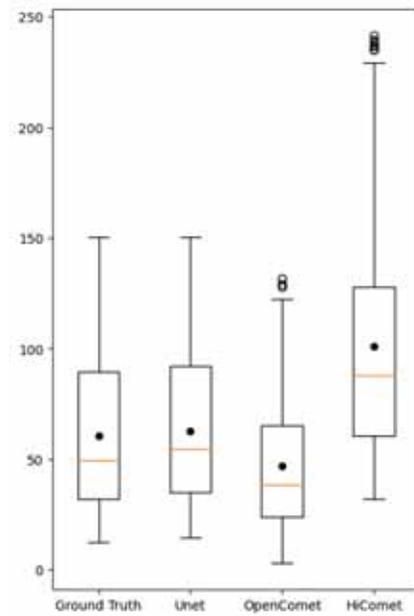


Figura 4.35: Diagrama de caja de la intensidad promedio en el cometa

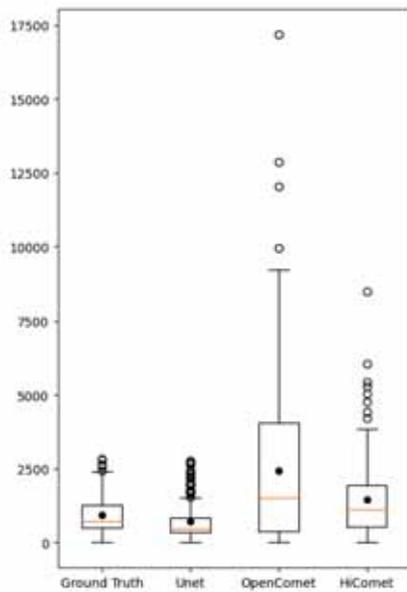


Figura 4.36: Diagrama de caja del área de la cabeza

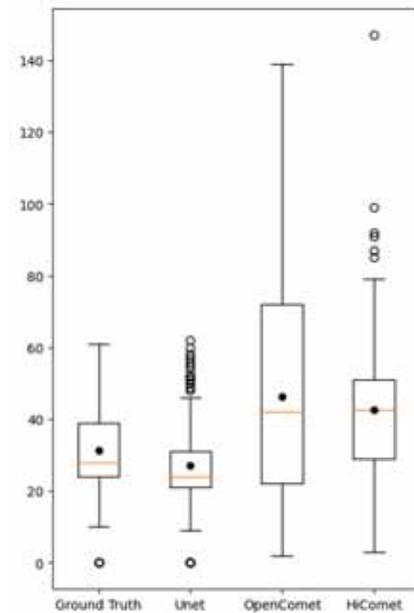


Figura 4.37: Diagrama de caja del diámetro de la cabeza

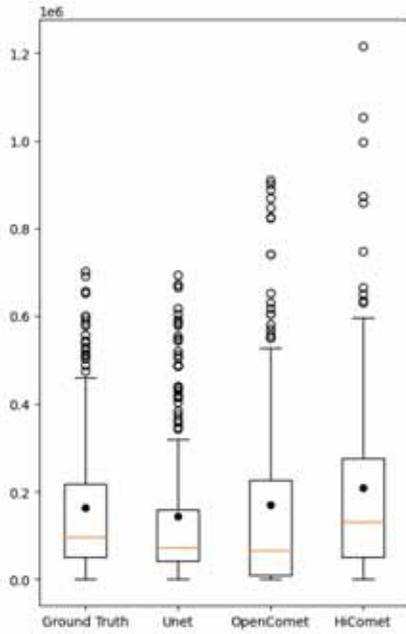


Figura 4.38: Diagrama de caja del contenido de ADN en la cabeza

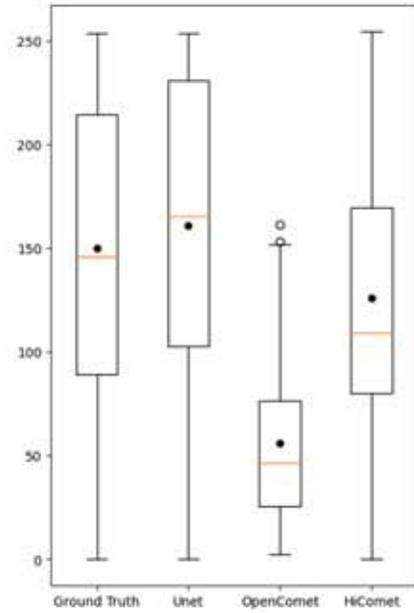


Figura 4.39: Diagrama de caja de la intensidad promedio en la cabeza

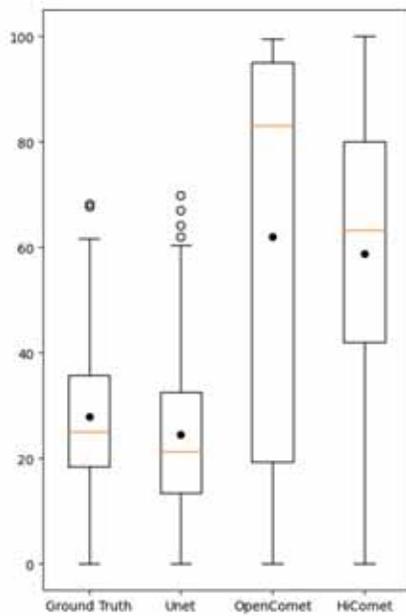


Figura 4.40: Diagrama de caja del porcentaje de ADN en la cabeza

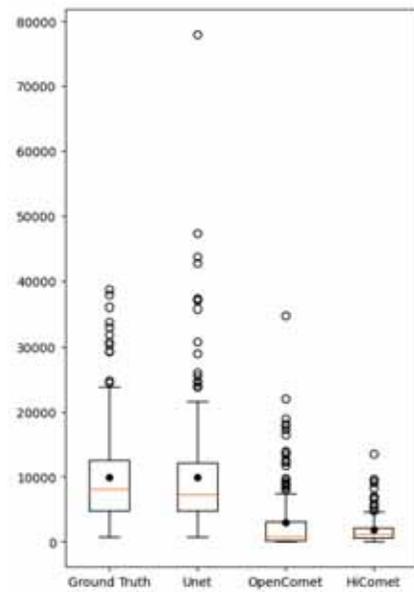


Figura 4.41: Diagrama de caja del área de la cola

4. RESULTADOS

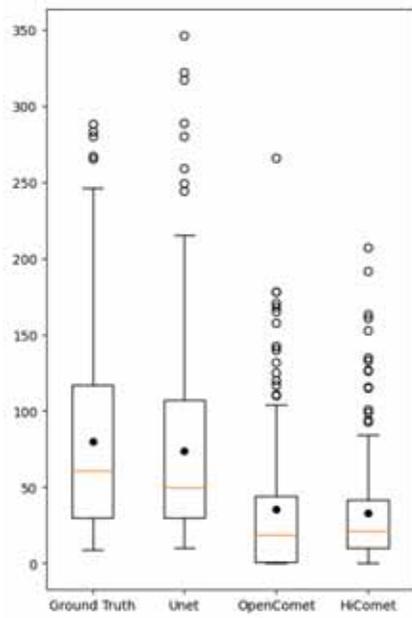


Figura 4.42: Diagrama de caja del largo de la cola

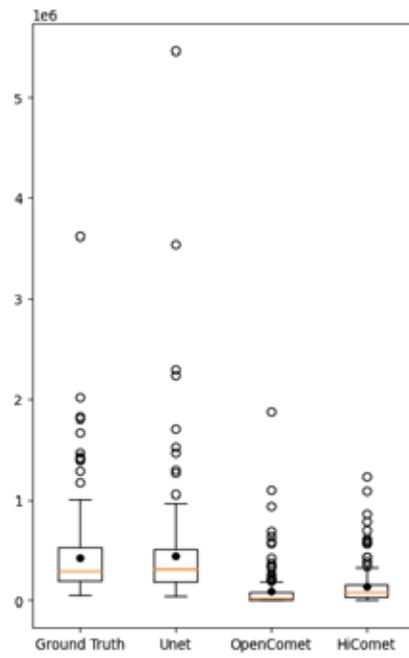


Figura 4.43: Diagrama de caja del contenido de ADN en la cola

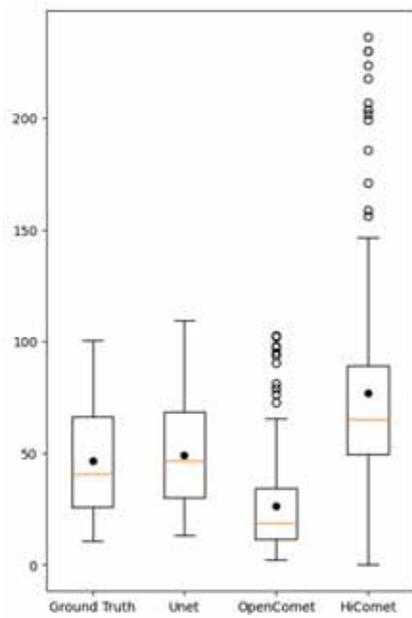


Figura 4.44: Diagrama de caja de la intensidad promedio en la cola

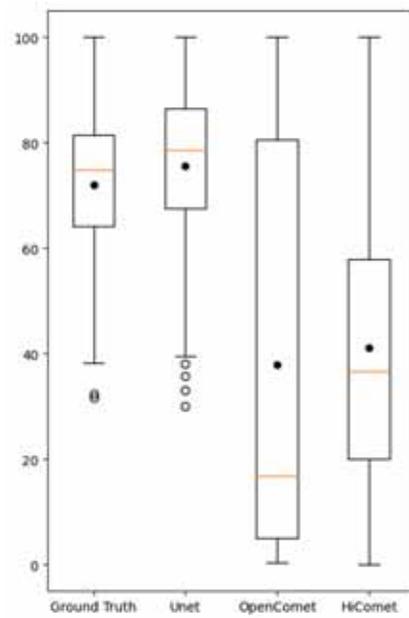


Figura 4.45: Diagrama de caja del porcentaje de ADN en la cola

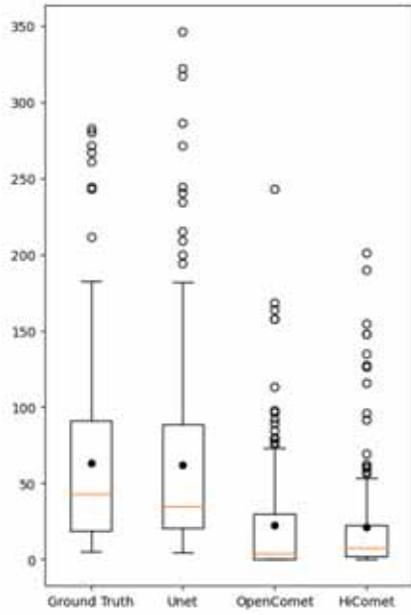


Figura 4.46: Diagrama de caja del momento de cola

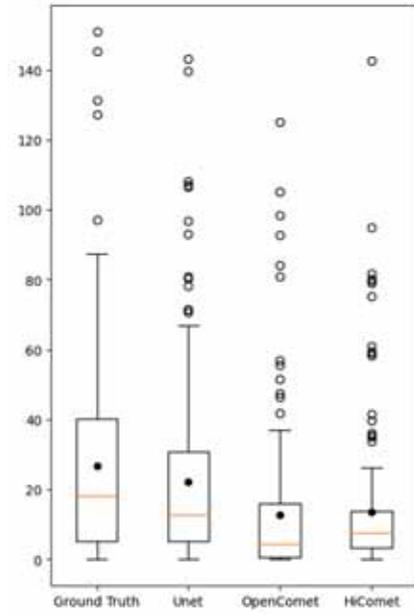


Figura 4.47: Diagrama de caja del momento de Olive

Conclusiones

En el presente trabajo se desarrolló un sistema computacional completamente automático, basado en un ensamble de redes neuronales convolucionales U-Net, con la tarea de segmentar y extraer las características de las células procesadas por el ensayo cometa.

Entre las principales contribuciones de este proyecto, se encuentra un modelo de aprendizaje profundo entrenado, que es capaz de segmentar y extraer características con un mejor desempeño que otros sistemas del estado de arte basados en técnicas de umbralización.

Las técnicas usadas en el módulo de refinamiento implicaron una mejora en el resultado de la segmentación, puesto que se aumentó el desempeño del F1-score y se eliminaron varios falsos positivos.

Por último, la manera de identificar cometas traslapados, falló en algunas regiones en donde sólo se identificó una cabeza en el interior. Además, el uso del algoritmo watershed para realizar la división de cometas traslapados resultó ser inconsistente en algunos casos, ya que no separaba las células de la mejor manera.

5.1. Trabajo futuro

Los resultados obtenidos en este trabajo representan un indicio para continuar desarrollando sistemas con modelos de aprendizaje profundo enfocados a la identificación de células procesadas por el ensayo cometa.

En un principio, se considera que sería razonable identificar los cometas traslapados de una manera diferente, como por ejemplo, el grado de redondez y no en función del número de cabezas en el interior de la región de traslape. De igual manera, sería conveniente implementar un algoritmo diferente al watershed, que sea capaz de dividir con un mejor desempeño los cometas traslapados.

Otro trabajo a futuro sería aumentar la base de datos usada en este trabajo, agregando imágenes de cometas con diferentes formas y grados de daño, con la finalidad de obtener un modelo más general que reconozca todo tipo de cometa.

Por último, se podría experimentar con otras arquitecturas de aprendizaje profundo, ya sean arquitecturas nuevas, existentes o más profundas que la U-Net, en caso de que se tenga acceso a recursos computacionales más potentes.

Bibliografía

- [1] N. P. Singh, M. T. McCoy, R. R. Tice, and E. L. Schneider, “A simple technique for quantitation of low levels of DNA damage in individual cells,” *Experimental Cell Research*, vol. 175, no. 1, pp. 184–191, 1988. [17](#), [25](#)
- [2] O. Ostling and K. Johanson, “Microelectrophoretic study of radiation-induced DNA damages in individual mammalian cells,” *Biochemical and Biophysical Research Communications*, vol. 123, no. 1, pp. 291–298, 1984. [17](#), [25](#)
- [3] A. Azqueta and A. R. Collins, “The essential comet assay: a comprehensive guide to measuring DNA damage and repair,” *Arch Toxicol*, vol. 87, no. 6, pp. 949–968, 2013. [17](#)
- [4] A. R. Collins, V. L. Dobson, M. Dusinská, G. Kennedy, and R. Stětina, “The comet assay: what can it really tell us?,” *Mutation Research*, vol. 375, no. 2, pp. 183–193, 1997. [17](#)
- [5] D. W. Fairbairn, P. L. Olive, and K. L. O’Neill, “The comet assay: a comprehensive review,” *Mutation Research/Reviews in Genetic Toxicology*, vol. 339, no. 1, pp. 37–59, 1995. [17](#)
- [6] T. Kumaravel, B. Vilhar, S. Faux, and N. J. Awadhesh, “Comet assay measurements: A perspective,” *Cell biology and toxicology*, vol. 25, no. 1, pp. 53–64, 2009. [18](#)
- [7] C. Helma and M. Uhl, “A public domain image-analysis program for the single-cell gel- electrophoresis (comet) assay,” *Mutation Research - Genetic Toxicology and Environmental Mutagenesis*, vol. 466, no. 1, pp. 9–15, 2000. [18](#), [19](#)
- [8] K. Końca, A. Lankoff, A. Banasik, H. Lisowska, T. Kuszewski, S. Gózdź, Z. Kozza, and A. Wojcik, “A cross-platform public domain PC image-analysis program for the comet assay,” *Mutation Research - Genetic Toxicology and Environmental Mutagenesis*, vol. 534, no. 1-2, pp. 15–20, 2003. [18](#), [19](#)
- [9] E. Kiziltan and E. Yurtcu, “Semi-automatic scoring tool for comet assay,” *Journal of Serbian Society for Computational Mechanics*, vol. 9, pp. 27–33, 2015. [19](#)

- [10] T. Lee, S. Lee, W. Y. Sim, Y. M. Jung, S. Han, J.-H. Won, H. Min, and S. Yoon, “HiComet: A high-throughput comet analysis tool for large-scale DNA damage assessment,” *BMC Bioinformatics*, vol. 19, no. 1, pp. 49–61, 2018. [19](#), [21](#), [23](#)
- [11] W. Böcker, W. Rolf, T. Bauch, W. Müller, and C. Streffer, “Automated comet assay analysis,” *Cytometry*, vol. 35, no. 2, pp. 134–144, 1999. [20](#)
- [12] P. Zamperoni, *Methoden der digitalen Bildsignalverarbeitung*. Vieweg, 1991. [20](#)
- [13] B. M. Gyori, G. Venkatachalam, P. Thiagarajan, D. Hsu, and M.-V. Clement, “OpenComet: An automated tool for comet assay image analysis,” *Redox Biology*, vol. 2, no. 1, pp. 457–465, 2014. [20](#), [23](#)
- [14] L.-K. Huang and M.-J. J. Wang, “Image thresholding by minimizing the measures of fuzziness,” *Pattern Recognition*, vol. 28, no. 1, pp. 41–51, 1995. [20](#)
- [15] Álvaro J. Bocanegra Pérez, D. S. Delgado, S. D. P. Castro, A. O. O. Méndez, F. J. L. Hoyos, and J. M. L. López, “Flexible automatic algorithm for comet assay analysis,” in *14th International Symposium on Medical Information Processing and Analysis*, vol. 10975, pp. 106–109, 2018. [21](#)
- [16] M. Sansone, O. Zeni, and G. Esposito, “Automated segmentation of comet assay images using gaussian filtering and fuzzy clustering,” *Medical and Biological Engineering and Computing*, vol. 50, pp. 523–532, 2012. [22](#)
- [17] J. Bezdek, J. Keller, R. Krisnapuram, and N. Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Springer, 2005. [22](#)
- [18] S. Ganapathy, A. Muraleedharan, P. Sathidevi, P. Chand, and R. P. Rajkumar, “CometQ: An automated tool for the detection and quantification of DNA damage using comet assay image analysis,” *Computer Methods and Programs in Biomedicine*, vol. 133, 2016. [22](#)
- [19] S. Ganapathy, A. Muraleedharan, P. Chand, R. P. Rajkumar, and P. Sathidevi, “An improved automatic detection of true comets for dna damage analysis,” *Procedia Computer Science*, vol. 46, pp. 135–142, 2015. [22](#)
- [20] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. [22](#)
- [21] S. Ganapathy, A. Muraleedharan, P. Sathidevi, P. Chand, and R. P. Rajkumar, “Quantification of dna damage by the analysis of silver stained comet assay images,” *IRBM*, vol. 36, no. 5, pp. 306–314, 2015. [22](#)
- [22] S. Ganapathy, K. Manoj, Aparna, and Sathidevi, “A level set approach combined with support vector machine for comet detection from comet assay images,” *Medical Imaging and Health Informatics*, vol. 8, pp. 984–991, 2016. [22](#)

-
- [23] S.-C. Huang, F.-C. Cheng, and Y.-S. Chiu, “Efficient contrast enhancement using adaptive gamma correction with weighting distribution,” *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 1032–1041, 2013. [22](#)
- [24] Y. Hong, H.-J. Han, H. Lee, D. Lee, J. Ko, Z.-y. Hong, J.-Y. Lee, J.-H. Seok, H. S. Lim, and I. Son, Woo-Chan ad Sonh, “Deep learning method for comet segmentation and comet assay image analysis,” *Scientific Reports*, vol. 10, 2016. [23](#)
- [25] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson Education, 2018. [27](#), [34](#), [35](#), [37](#), [46](#), [48](#), [50](#), [51](#)
- [26] H. Digabel and C. Lantuejoul, “Iterative algorithms,” in *Proceedings of the 2nd European Symposium Quantitative Analysis of Microstructures in Material Science, Biology and Medicine*, pp. 85–89, 1978. [29](#)
- [27] J. B. Roerdink and A. Meijster, “The watershed transform: definitions, algorithms and parallelization strategies,” *Fundamenta Informaticae*, vol. 41, no. 1-2, pp. 187–228, 2000. [29](#), [68](#)
- [28] M. Hossin and M. Sulaiman, “A review on evaluation metrics for data classification evaluations,” *International Journal of Data Mining and Knowledge Management Process*, vol. 5, no. 2, pp. 01–11, 2015. [31](#), [32](#)
- [29] M. V. Joshi, “On evaluating performance of classifiers for rare classes,” in *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pp. 641–644, 2002. [32](#)
- [30] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, L. Wang, G. Wang, J. Cai, and T. Chen, “Recent advances in convolutional neural networks,” *Pattern Recognition*, vol. 77, pp. 354–377, 2018. [33](#), [38](#)
- [31] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, 2017. [33](#)
- [32] I. Sobel, “An isotropic 3x3 image gradient operator,” 2014. [34](#)
- [33] J. Prewitt, “Object enhancement and extraction,” *Picture Processing and Psychopictorics*, pp. 75–149, 1970. [34](#)
- [34] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 249–256, 2010. [34](#), [58](#)
- [35] H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1520–1528, 2015. [36](#)

- [36] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*, p. 807–814, 2010. [38](#)
- [37] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006. [38](#)
- [38] J. Brownlee, *Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions*. 2019. [39](#), [41](#), [42](#), [45](#)
- [39] N. Cui, “Applying gradient descent in convolutional neural networks,” *Journal of Physics: Conference Series*, vol. 1004, 2018. [40](#)
- [40] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representation (ICLR)*, 2015. [40](#)
- [41] S. Yadav and S. Shukla, “Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification,” in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, pp. 78–83, 2016. [42](#)
- [42] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015. [44](#), [56](#), [59](#)
- [43] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, “The quickhull algorithm for convex hulls,” *ACM Transactions on Mathematical Software*, vol. 22, no. 4, p. 469–483, 1996. [46](#)
- [44] J. S. Greenfield, “A proof for a quickhull algorithm,” in *Electrical Engineering and Computer Science - Technical Report*, 1990. [46](#)