

February 29, 2024.

Ciudad de Guatemala, Guatemala

# Implementación de Algoritmos de Binarización y Morfología para Reconocimiento de Arterias Coronarias en Angiogramas Coronarios

## *Implementation of Binarization and Morphology Algorithms for Coronary Artery Recognition in Coronary Angiograms*

**GALICIA. J, ESPINOZA. J, RAMÍREZ. J, MENDEZ. O**

carne 20298. e-mail: gal20298@uvg.edu.gt carne 20022. e-mail: esp20022@uvg.edu.gt carne 20666. e-mail: ram20666@uvg.edu.gt carne 20402. e-mail: men20402@uvg.edu.gt

Universidad del Valle de Guatemala, Facultad de Ingeniería.  
Departamento de Ciencias de la Computación, Computer Vision  
Instructor: Alan Reyes

### ABSTRACT

Este informe presenta la implementación y análisis comparativo de algoritmos de binarización destinados a reconocer la estructura arterial principal en imágenes de angiogramas coronarios. Se investigaron y se implementaron diversos algoritmos de binarización, tanto globales como locales, con el objetivo de lograr este fin. Los algoritmos se aplicaron a un conjunto de datos que comprende 20 pares de imágenes, cada uno consistente en una fotografía original en escala de grises de rayos X que muestra las arterias coronarias y una imagen binaria ground truth etiquetada manualmente que representa la estructura arterial principal.

Se calcularon métricas como precisión, sensibilidad/recuperación, especificidad, precisión y puntaje F1 para cada algoritmo en todo el conjunto de datos, con promedios tabulados para su comparación.

Además, se compararon dos algoritmos de binarización distintos para determinar cuál produce resultados superiores con el conjunto de imágenes. Además, se desarrolló un algoritmo para generar una discretización de la estructura arterial basada en las imágenes ground truth, representada como un grafo de vértices y aristas conectantes.

Los hallazgos resaltan la eficacia de diferentes algoritmos de binarización en la identificación de estructuras arteriales coronarias en imágenes de angiogramas, proporcionando información sobre su rendimiento y aplicabilidad en tareas de análisis de imágenes médicas.

This report presents the implementation and comparative analysis of binarization algorithms aimed at recognizing the main arterial structure in coronary angiogram images. Various binarization algorithms, both global and local, were investigated and implemented to achieve this goal. The algorithms were applied to a dataset comprising 20 pairs of images, each consisting of an original grayscale X-ray photograph showing coronary arteries and a manually labeled ground truth binary image depicting the main coronary artery structure.

Metrics including accuracy, sensitivity/recall, specificity, precision, and F1 score were computed for each algorithm across the dataset, with averages tabulated for comparison.

Furthermore, two distinct binarization algorithms were compared to determine the one yielding superior results with the image dataset. Additionally, an algorithm was developed to generate a discretization of the arterial structure based on the ground truth images, represented as a graph of vertices and connecting edges. The findings highlight the efficacy of different binarization algorithms in identifying coronary artery structures in angiogram images, providing insights into their performance and applicability in medical image analysis tasks.

## I. MARCO CONCEPTUAL

### I-A. CLAHE (CONTRAST LIMITED ADAPTIVE HISTOGRAM EQUALIZATION)

CLAHE modifica el histograma de una imagen para mejorar el contraste localmente mediante la siguiente fórmula para la ecualización de histograma en bloques pequeños o *tiles* de la imagen:

$$H'(v) = \frac{\sum_{i=0}^v H(i)}{N} \quad (1)$$

donde  $H(i)$  es el histograma de intensidades,  $H'(v)$  es el histograma ecualizado, y  $N$  es el número total de píxeles en la región. La interpolación bilineal se utiliza entre regiones adyacentes para suavizar las transiciones.

### I-B.

#### SKIMAGE.MORPHOLOGY.REMOVE\_SMALL\_OBJECTS

La función identifica y elimina componentes conectados de tamaño inferior a un umbral predefinido en una imagen binaria:

$$A' = A - \bigcup_{i=1}^N C_i \quad \text{si } |C_i| < \theta \quad (2)$$

donde  $A$  es la imagen original,  $A'$  es la imagen resultante,  $C_i$  son los componentes conectados,  $|C_i|$  es el número de píxeles en el componente  $i$ ,  $N$  es el número total de componentes, y  $\theta$  es el umbral de tamaño.

### I-C.

#### SKIMAGE.MORPHOLOGY.REMOVE\_SMALL\_HOLES

Esta operación rellena huecos dentro de los objetos en una imagen binaria si son menores a un tamaño especificado:

$$A' = A \cup \bigcup_{i=1}^M H_i \quad \text{si } |H_i| < \phi \quad (3)$$

donde  $H_i$  son los huecos,  $|H_i|$  es el número de píxeles en el hueco  $i$ ,  $M$  es el número total de huecos, y  $\phi$  es el umbral de tamaño.

### I-D. BILATERAL FILTER

El filtro bilateral utiliza un promedio ponderado de las intensidades de los píxeles cercanos, considerando tanto la distancia espacial como la diferencia en intensidad:

$$I'(x) = \frac{1}{W_p} \sum_{x_i \in S} I(x_i) \cdot f(\|x - x_i\|) \cdot g(|I(x) - I(x_i)|) \quad (4)$$

donde  $I(x)$  es la intensidad del píxel original,  $I'(x)$  es la intensidad del píxel filtrado,  $S$  es el conjunto de píxeles en la vecindad,  $f$  es la función de peso espacial,  $g$  es la función de peso de intensidad, y  $W_p$  es un factor de normalización.

### I-E. MEDIAN BLUR

El desenfoque mediano reemplaza cada píxel por la mediana de los píxeles en su vecindad:

$$I'(x) = \text{mediana}\{I(x_1), I(x_2), \dots, I(x_n)\} \quad (5)$$

donde  $I(x)$  es la intensidad del píxel original,  $I'(x)$  es la intensidad después del desenfoque, y  $I(x_1), \dots, I(x_n)$  son las intensidades de los píxeles en la ventana alrededor de  $x$ .

### I-F. PLANTEAMIENTO DEL PROBLEMA

El análisis preciso de imágenes de angiogramas coronarios es fundamental para la detección y tratamiento de enfermedades cardiovasculares. Sin embargo, la complejidad y variabilidad de las estructuras arteriales presentes en estas imágenes representan un desafío significativo. Las técnicas de binarización y morfología existentes no logran resaltar de manera eficiente estas estructuras, lo que conduce a una necesidad crítica de desarrollar algoritmos avanzados que mejoren la claridad y utilidad de las imágenes médicas para fines diagnósticos. Este proyecto se enfoca en la implementación y evaluación de algoritmos de binarización y morfología mejorados, con el objetivo de optimizar la visualización de las estructuras arteriales en angiogramas coronarios, facilitando así el diagnóstico y tratamiento de condiciones cardiovasculares.

## II. OBJETIVOS

### II-A. OBJETIVOS

- Desarrollar y evaluar algoritmos de binarización y morfología avanzados para mejorar la claridad y utilidad de las imágenes de angiogramas coronarios, facilitando así el diagnóstico y tratamiento de enfermedades cardiovasculares.

## III. MATERIALES Y METODOLOGÍA

### III-A. MATERIALES

Para la realización de este algoritmo es preciso contar con una computadora con el lenguaje de programación Python instalado. Asimismo, las librerías utilizadas fueron:

1. random: librería utilizada para hacer procesos aleatorios
2. collections: librería que contiene la función Counter, la cuál ayuda a hacer eficiente el proceso de contar elementos en una lista
3. math: librería que contiene las funciones para realizar operaciones matemáticas de manera más eficiente
4. Pygame: librería especializada en facilitar el desarrollo de videojuegos y aplicaciones de multimedia. Utilizada para mostrar dinámicamente el cambio iteración tras iteración del algoritmo mediante frames.
5. Numpy: librería utilizada para la manipulación de matrices
6. Matplotlib-Seaborn: librerías especializadas en visualización. Utilizadas para poder graficar los resultados obtenidos durante la experimentación.

Además se utiliza la página [sudoku.com/es](http://sudoku.com/es) para obtener los sudokus y sus dificultades.

### III-B. METODOLOGIA

La implementación del algoritmo de binarización gaussiana ocurre en los siguientes pasos:

### Algorithm 1 Binarización Gaussiana

```

original_image ← Apply threshold where intensity is less than 35 to 255
image_bilateral ← Apply bilateral filter with d = 9, sigma Color = 23, sigma Space=95
image_clahe ← Apply CLAHE with clip Limit = 3.46 and tile GridSize = (7, 7)
adaptive_thresh ← Apply adaptive Gaussian thresholding
kernel ← Create elliptical kernel of size (3, 3)
kernel2 ← Create elliptical kernel of size (10, 1)
closing ← Apply morphological closing with kernel, iterate 2 times
binary_cleaned ← Remove small objects from closing, min_size = 64
filled_cleaned ← Remove small holes from binary_cleaned, area_threshold = 128
final_image ← Convert filled_cleaned to 255 scale
opening ← Apply morphological opening on final_image with kernel2, iterate 10 times
final_smoothed_image ← Invert opening result to 255 scale
median_filtered ← Apply median blur with size 9
kernel3 ← Create elliptical kernel of size (5, 1)
dilated ← Dilate median_filtered with kernel3, iterate 10 times
rounded_edges_image ← Erode dilated with kernel3, iterate 20 times
return FinalImage =0

```

Ahora bien la implementación del algoritmo de binarización simple utilizando la media ocurre de esta manera:

### Algorithm 2 Calculate\_Binary(original\_image)

```

0: original_image ← np.where(original_image < 30, 255, original_image)
0: preprocessed_image_bilateral ← cv2.bilateralFilter(original_image, 3, 25, 20)
0: adaptive_binary_image ← cv2.adaptiveThreshold(preprocessed_image_bilateral, 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 29, 9)
0: final_filtered_image ← cv2.medianBlur(adaptive_binary_image, 1)
0: adjusted_kernel ← cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5, 1))
0: morph_opening_adjusted ← cv2.morphologyEx(final_filtered_image, cv2.MORPH_OPEN, adjusted_kernel, iterations=1)
0: morph_closing_adjusted ← cv2.morphologyEx(morph_opening_adjusted, cv2.MORPH_CLOSE, adjusted_kernel, iterations=1)
0: final_filtered_image ← cv2.medianBlur(morph_closing_adjusted, 5)
0: return (255 - final_filtered_image) =0

```

De esta manera se pudieron binarizar las imágenes de dos maneras diferentes, aplicando además de thresholds, ciertos elementos de filtros y operadores morfológicos para poder limpiar las imágenes. Luego se tomaron dos clases, la primera clase 0 es el color negro y la clase 1 es el color blanco en nuestras imágenes binarizadas. Comenzando con imágenes como la mostrada en la Figura 1 para poder llegar a compararlas con su imagen respuesta previamente

identificada por expertos:

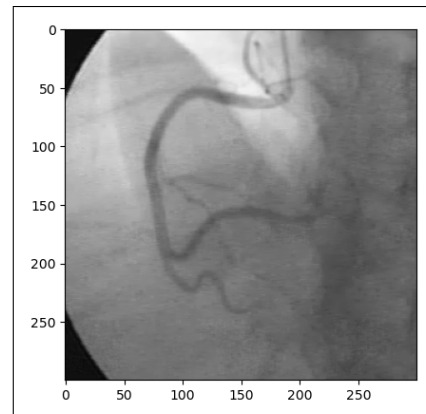


Figura 1: Estructura de Arterias Coronarias

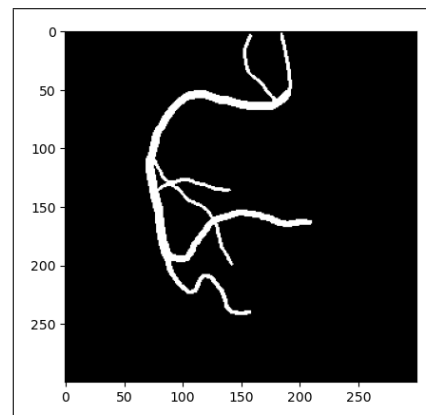


Figura 2: Estructura Binaria de Arterias Coronarias

#### IV. RESULTADOS

En la primera parte se buscó poder comparar ambos métodos de binarización. Obteniendo los resultados generales por imagen para ambos métodos, mostrados en Tabla 1 y Tabla 2. Obteniendo en promedio los siguientes resultados para ambos métodos respectivamente:

Promedio Accuracy	0.951866
Promedio Recall	0.51985
Promedio Specificity	0.985928
Promedio Precision	0.727122
Promedio F1 Score	0.594938

Tabla 1: Promedios de métricas de rendimiento Binarización Simple

Promedio Accuracy	0.964227
Promedio Recall	0.62132
Promedio Specificity	0.986235
Promedio Precision	0.731317
Promedio F1 Score	0.66424

Tabla 2: Promedios de métricas de rendimiento Binarización Gauss

Donde podemos observar que en general todas las métricas son superiores en el algoritmo de binarización mediante un threshold Gauss. Asimismo, analizando las métricas por categoria, es decir por separado el grupo de pixeles negros y el grupo de pixeles blancos, obtenemos los siguientes resultados:

##### Algoritmo Binarización Adaptativa Simple

Promedio Accuracy	0.951866
Promedio Recall	0.985928
Promedio Specificity	0.51985
Promedio Precision	0.963196
Promedio F1 Score	0.97435

Tabla 3: Métricas de rendimiento para el Grupo 0 con Algoritmo 1

Promedio Accuracy	0.951866
Promedio Recall	0.51985
Promedio Specificity	0.985928
Promedio Precision	0.727122
Promedio F1 Score	0.594938

Tabla 4: Métricas de rendimiento para el Grupo 1 con Algoritmo 1

##### Algoritmo Binarización Adaptativa Gauss

Por lo que podemos ver que en la comparación por grupo, principalmente enfocandonos en el grupo 1 que es el que nos interesa al ser el grupo de pixeles blancos que son nuestro objetivo, podemos ver que de igual manera la mejoría es notoria en el algoritmo de binarización de Gauss a comparación del Adaptativo por medio de la Media.

Asimismo, podemos observar una comparación entre ambos métodos, donde el color estilo verde es la binarización utilizando un threshold Gauss y el fondo estilo naranja es

Promedio Accuracy	0.964227
Promedio Recall	0.986235
Promedio Specificity	0.62132
Promedio Precision	0.975975
Promedio F1 Score	0.981039

Tabla 5: Métricas de rendimiento para el Grupo 0 con Algoritmo 2

Promedio Accuracy	0.964227
Promedio Recall	0.62132
Promedio Specificity	0.986235
Promedio Precision	0.731317
Promedio F1 Score	0.66424

Tabla 6: Métricas de rendimiento para el Grupo 1 con Algoritmo 2

la binarización mediante un threshold de la media. Donde podemos observar que este ultimo tipo de binarización que es mucho más sencilla no logró eliminar mucho del ruido cerca de la arteria coronaria. Por lo que la imagen mediante binarización Gaussiana es mucho más limpia.

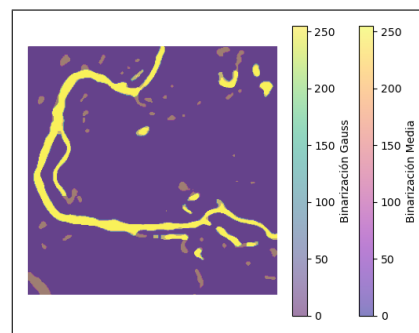


Figura 3: Comparación Grafica entre Binarizaciones

Luego podemos observar una comparación gráfica entre el método de binarización utilizando un threshold Gauss contra su respectiva imagen de GroundTruth. Donde podemos observar de color fosforescente la binarización Gauss y de fondo la imagen de respuesta. Por lo que podemos ver que la binarización salió en este caso bastante bien, pero no logró identificar pequeños pedazos de la arteria muy delgados. Y de igual manera no pudo eliminar absolutamente todo el ruido.

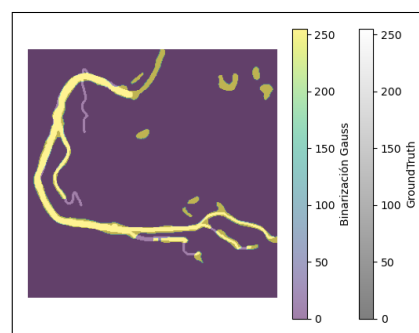


Figura 4: Comparación Grafica entre Binarización Gauss y GroundTruth

## V. DISCUSIÓN

Se observa claramente una tendencia favorable hacia el algoritmo de binarización mediante un threshold Gauss en comparación con el método adaptativo basado en la media. Este resultado se refleja tanto en las métricas generales por imagen como en el análisis por categorías de píxeles blancos y negros.

Inicialmente, al comparar los resultados generales por imagen, se evidencia que todas las métricas son consistentemente superiores en el algoritmo de binarización mediante un threshold Gauss. Esto sugiere una mayor precisión y efectividad en la segmentación de la estructura arterial principal en las imágenes de angiogramas coronarios utilizando este enfoque.

Al profundizar en el análisis por categorías de píxeles blancos y negros, se confirma la superioridad del algoritmo de binarización Gaussiano, particularmente en el grupo de píxeles blancos, que es de particular interés ya que representa la estructura arterial que buscamos identificar y segmentar. Los resultados muestran una mejora notable en la precisión y sensibilidad en la detección de los píxeles blancos utilizando el método de binarización Gaussiano en comparación con el método adaptativo basado en la media.

Estos hallazgos sugieren que el algoritmo de binarización mediante un threshold Gauss es más robusto y efectivo para reconocer y segmentar la estructura arterial principal en las imágenes de angiogramas coronarios. Esto puede atribuirse a la capacidad del método Gaussiano para adaptarse mejor a las variaciones locales de intensidad de los píxeles, lo que resulta en una segmentación más precisa y detallada de la estructura arterial. En contraste, el enfoque adaptativo basado en la media puede ser más susceptible a fluctuaciones en la intensidad de los píxeles, lo que puede afectar negativamente la precisión de la segmentación.

## VI. CONCLUSIONES

En conclusión, los resultados de este estudio respaldan la elección del algoritmo de binarización mediante un threshold Gauss como la opción preferida para la segmentación de estructuras arteriales en imágenes de angiogramas coronarios, debido a su superioridad en términos de precisión y sensibilidad.

Se concluye que el método de binarización Gaussiano supera al método adaptativo basado en la media en términos de precisión y sensibilidad para la segmentación de estructuras arteriales en angiogramas coronarios.

Aunque el método de binarización Gaussiano es más eficiente computacionalmente, es importante considerar la generalización y robustez de ambos métodos en diferentes contextos clínicos para determinar su idoneidad en aplicaciones médicas.

## VII. RECOMENDACIONES

Basado en los resultados, se recomienda prioritariamente el uso del método de binarización Gaussiano sobre el método adaptativo basado en la media para la segmentación

de estructuras arteriales en angiogramas coronarios, dada su superioridad en precisión y sensibilidad. Se sugiere validar estos métodos en conjuntos de datos más amplios y variados, optimizando los parámetros específicos de cada método para adaptarse mejor a las características de las imágenes. Además, se insta a explorar técnicas adicionales de procesamiento de imágenes y documentar detalladamente los métodos y resultados para facilitar la comparación y la investigación continua en el campo de la imagen médica.

## VIII. MARCO CONCEPTUAL EJERCICIO 2

### VIII-A. ESQUELETIZACIÓN

Para una imagen binaria continua  $X \subset \mathbb{R}^2$ , el esqueleto  $S(X)$  se define como la unión de las intersecciones de erosiones y aberturas morfológicas de la imagen  $X$ , aplicadas con elementos estructurantes de diferentes tamaños. La fórmula de Lantuéjoul para imágenes continuas se expresa como:

$$S(X) = \bigcup_{\rho > 0} \bigcap_{\mu > 0} [(X \ominus \rho B) - (X \ominus \rho B) \circ \mu \bar{B}]$$

donde  $\ominus$  y  $\circ$  representan la erosión y la apertura morfológica, respectivamente,  $\rho B$  es una bola abierta de radio  $\rho$ , y  $\bar{B}$  es el cierre de  $B$ .

Para imágenes binarias discretas  $X \subset \mathbb{Z}^2$ , la fórmula de Lantuéjoul se discretiza de la siguiente manera. El esqueleto  $S(X)$  se calcula como la diferencia entre dos erosiones morfológicas, también conocidas como el operador "by openings":

$$S_n(X) = (X \ominus nB) - (X \ominus nB) \circ B$$

donde  $nB$  representa una familia de formas obtenidas por la dilatación repetida de un elemento estructurante  $B$   $n$  veces, y  $\circ B$  representa la apertura morfológica con  $B$ .

El esqueleto  $S(X)$  consiste en los centros de los discos máximos en el conjunto  $X$ , donde un disco máximo es una forma circular que no puede expandirse sin abandonar  $X$ . Estos centros se obtienen como la diferencia entre dos erosiones morfológicas, lo que resulta en una representación simplificada de la forma y la topología de  $X$ .

### VIII-B. SKNW

La biblioteca `sknw` es una herramienta para la extracción de esqueletos de imágenes binarias. Utiliza algoritmos de morfología matemática y operaciones de grafos para encontrar la topología de las estructuras presentes en una imagen.

1. **Extracción del esqueleto:** La función `build_sknw` extrae el esqueleto de una imagen binaria utilizando el algoritmo desarrollado. Toma como entrada una imagen binaria y produce un grafo que representa la topología de las estructuras presentes en la imagen.
2. **Proceso de marcado y llenado:** El proceso comienza marcando los píxeles de la imagen de acuerdo con ciertas reglas predefinidas. Luego, se procede a llenar



las regiones marcadas para identificar los nodos del grafo.

3. **Construcción del grafo:** Una vez identificados los nodos, se establecen las conexiones entre ellos para construir el grafo. Esto se logra trazando los bordes y determinando las conexiones entre los nodos vecinos.
4. **Representación del esqueleto:** El esqueleto se representa como un conjunto de nodos y aristas, donde los nodos representan los puntos singulares y las aristas representan las conexiones entre ellos.
5. **Visualización del esqueleto:** Finalmente, el esqueleto se puede visualizar superpuesto en la imagen original para facilitar la interpretación visual de la topología de las estructuras presentes en la imagen.

## IX. METODOLOGÍA EJERCICIO 2

### IX-A. FUNCIÓN GENERATE\_GRAPH

La función `generate_graph` tiene como objetivo construir un grafo a partir de la eskeletonización de una imagen utilizando la biblioteca `sknw`. Esta función toma como entrada la ruta de la imagen y devuelve un grafo representado como un objeto de la biblioteca `NetworkX`.

---

#### Algorithm 3 Generación de Grafo desde Esqueleto de Imagen

---

```

0: Lee la imagen utilizando OpenCV.
0: Aplica la eskeletonización a la imagen utilizando la
  función skeletonize.
0: Construye un grafo a partir de la eskeletonización utilizando
  la función sknw.build_sknw.
0: Inicializa un grafo vacío utilizando la biblioteca NetworkX.
0: Inicializa un contador para asignar identificadores a los
  nodos del grafo.
0: for all aristas ( $s, e$ ) en el grafo de eskeletonización do
0:   Obtiene los puntos de la arista reduciendo su densidad.
0:   Concatena los puntos de inicio y fin de la arista con
     los puntos de la arista.
0:   Elimina puntos duplicados y mantiene el orden original.
0:   Busca si el nodo ya existe en el grafo de NetworkX.
0:   if el nodo existe then
0:     Actualiza el nodo previo.
0:   else
0:     Añade el nodo al grafo con un nuevo identificador.
0:   end if
0:   Añade la arista al grafo de NetworkX.
0: end for
0: Devuelve el grafo resultante. =0

```

---

### IX-B. FUNCIÓN DISPLAY\_GRAPH\_INFO

La función `display_graph_info` analiza y visualiza la estructura del grafo generado a partir de la eskeletonización de una imagen. Esta función proporciona información

sobre los diferentes tipos de nodos presentes en el grafo, así como la ubicación de las aristas.

---

#### Algorithm 4 Visualizar Información del Grafo

---

```

0: function DISPLAY_GRAPH_INFO( $G$ ,  $og\_path$ ,  $res\_path$ ,  $info\_path$ )
0:  $img \leftarrow$  Leer la imagen original desde  $original\_img\_path$ 
0:  $extremos \leftarrow$  Nodos con grado 1 en  $G$ 
0:  $intermedios \leftarrow$  Nodos con grado 2 en  $G$ 
0:  $bifurcaciones \leftarrow$  Nodos con grado 3 en  $G$ 
0:  $trifurcaciones \leftarrow$  Nodos con grado mayor o igual a
  4 en  $G$ 
0:  $nodos \leftarrow$  Concatenar extremos, intermedios, bifurcaciones
  y trifurcaciones
0:  $graph\_dict \leftarrow$  Diccionario con la información del
  grafo
0:  $graph\_dict["nodos"] \leftarrow$  Información de todos los
  nodos en  $G$ 
0:  $graph\_dict["extremos"] \leftarrow$  Identificadores de
  nodos extremos
0:  $graph\_dict["bifurcaciones"] \leftarrow$  Identificadores de
  nodos bifurcaciones
0:  $graph\_dict["trifurcaciones"] \leftarrow$  Identificadores de
  nodos trifurcaciones
0:  $graph\_dict["nodos\_intermedios"] \leftarrow$  Identificadores
  de nodos intermedios
0:  $graph\_dict["aristas"] \leftarrow$  Información de todas las
  aristas en  $G$ 
0: Guardar  $graph\_dict$  en un archivo JSON en  $info\_path$ 
0: Generar una visualización del grafo y superponerla en
  la imagen original
0: Guardar la imagen resultante en  $result\_img\_path$ 
0: end function=0

```

---

## X. RESULTADOS

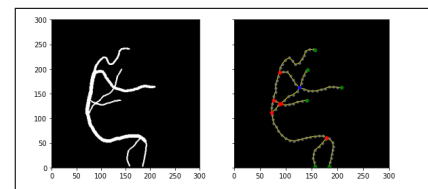


Figura 5: Ejercicio 2: Imagen 1

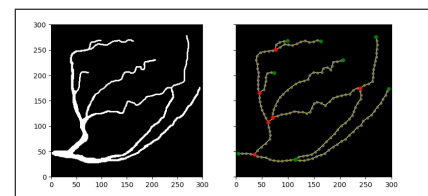


Figura 6: Ejercicio 2: Imagen 12

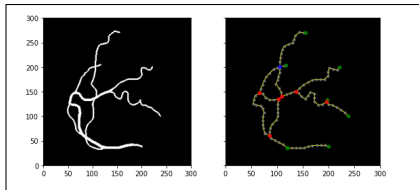


Figura 7: Ejercicio 2: Imagen 11

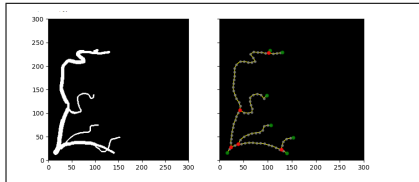


Figura 8: Ejercicio 2: Imagen 8

## Referencias

- [1] Image-Py/sknw. (s.f.). Recuperado de <https://github.com/Image-Py/sknw/blob/master/README.md>
- [2] Wikipedia. (s.f.). *Morphological skeleton*. Recuperado de [https://en.wikipedia.org/wiki/Morphological\\_skeleton](https://en.wikipedia.org/wiki/Morphological_skeleton)

## XIV. ANEXOS

### XI. DISCUSIÓN EJERCICIO 2

La implementación del algoritmo logra con éxito la creación del grafo a partir del esqueleto de la imagen, como evidencian las figuras proporcionadas en el análisis. Sin embargo, se observa una limitación significativa en la capacidad de la biblioteca sknw para detectar las estructuras más finas de las venas, lo que resulta en su desaparición en ciertas regiones, como se ilustra en la Imagen 12. Esta deficiencia puede atribuirse a la sensibilidad del algoritmo de esqueletonización ante ciertos tipos de patrones de imagen, especialmente cuando se trata de estructuras vasculares altamente ramificadas y delgadas. Por otro lado, se destaca que el algoritmo logra identificar sin errores los nodos intermedios, bifurcaciones y trifurcaciones, así como los extremos, proporcionando una representación precisa de la topología del sistema vascular en la imagen analizada.

### XII. CONCLUSIÓN EJERCICIO 2

En resumen, el algoritmo implementado demuestra una eficacia notable en la construcción del grafo a partir del esqueleto de la imagen. A pesar de las limitaciones encontradas en la capacidad de la biblioteca sknw para detectar las estructuras más finas de las venas, el algoritmo logra identificar correctamente los nodos intermedios, bifurcaciones y trifurcaciones, así como los extremos. Esto indica una sólida capacidad para representar la topología de la red vascular en la imagen analizada.

### XIII. RECOMENDACIÓN EJERCICIO 2

Para mejorar aún más la precisión y robustez del algoritmo, se recomienda realizar ajustes en los parámetros utilizados en los procesos de esqueletonización y construcción del grafo, así como explorar técnicas complementarias para el preprocesamiento de imágenes y la mejora de la detección de estructuras vasculares delicadas. Además, la evaluación y validación exhaustivas del algoritmo en una variedad de conjuntos de datos pueden proporcionar información útil para su refinamiento y optimización futuros.

Tabla 7: Tabla de resultados por imagen método adaptativo 1

Imagen	Accuracy	Recall	Specificity	Precision	F1 Score	
0	1.pmg	0.951989	0.456965	0.989083	0.758265	0.570264
1	2.pmg	0.955767	0.520175	0.990334	0.810264	0.633594
2	3.pmg	0.968933	0.590790	0.992795	0.838024	0.693017
3	4.pmg	0.962389	0.582682	0.978118	0.524516	0.552071
4	5.pmg	0.937144	0.326445	0.988012	0.694009	0.444029
5	6.pmg	0.936167	0.333469	0.990122	0.751371	0.461928
6	7.pmg	0.961678	0.593411	0.984495	0.703386	0.643735
7	8.pmg	0.949033	0.462577	0.978766	0.571088	0.511137
8	9.pmg	0.945856	0.480532	0.987838	0.780930	0.594963
9	10.pmg	0.923756	0.318916	0.983137	0.649949	0.427881
10	11.pmg	0.932311	0.326183	0.983375	0.623054	0.428196
11	12.pmg	0.946456	0.619936	0.985675	0.838660	0.712898
12	13.pmg	0.953822	0.456084	0.992941	0.835475	0.590057
13	14.pmg	0.957078	0.703996	0.976228	0.691437	0.697660
14	15.pmg	0.957278	0.540373	0.992896	0.866652	0.665681
15	16.pmg	0.947411	0.513899	0.993412	0.892218	0.652164
16	17.pmg	0.961178	0.591763	0.980253	0.607433	0.599496
17	18.pmg	0.955744	0.574277	0.975648	0.551657	0.562740
18	19.pmg	0.973700	0.747123	0.983277	0.653800	0.697353
19	20.pmg	0.959633	0.657404	0.992160	0.900251	0.759897

Tabla 8: Tabla total método adaptativo 2

Imagen	Accuracy	Recall	Specificity	Precision	F1 Score	
0	1.pmg	0.976022	0.684978	0.990936	0.794763	0.735798
1	2.pmg	0.964322	0.585603	0.991649	0.834981	0.688404
2	3.pmg	0.970622	0.611398	0.991914	0.817578	0.699614
3	4.pmg	0.962933	0.601973	0.975995	0.475735	0.531461
4	5.pmg	0.968300	0.549901	0.987903	0.680492	0.608266
5	6.pmg	0.963989	0.504425	0.988930	0.712066	0.590524
6	7.pmg	0.958867	0.565000	0.985000	0.714221	0.630907
7	8.pmg	0.965022	0.647488	0.978111	0.549417	0.594434
8	9.pmg	0.963489	0.615166	0.986737	0.755837	0.678285
9	10.pmg	0.974244	0.692580	0.988142	0.742401	0.716626
10	11.pmg	0.945956	0.405529	0.987098	0.705272	0.514958
11	12.pmg	0.954800	0.672247	0.985808	0.838660	0.746289
12	13.pmg	0.974722	0.644646	0.992139	0.812291	0.718823
13	14.pmg	0.954778	0.672775	0.978105	0.717654	0.694490
14	15.pmg	0.963133	0.583473	0.993095	0.869595	0.698364
15	16.pmg	0.956400	0.571157	0.990766	0.846571	0.682113
16	17.pmg	0.970478	0.762923	0.977966	0.555401	0.642828
17	18.pmg	0.968211	0.721080	0.979817	0.626560	0.670506
18	19.pmg	0.972433	0.688700	0.987290	0.739391	0.713146
19	20.pmg	0.955811	0.645348	0.987297	0.837457	0.728958