UNIVERSIDAD DE GUANAJUATO

CAMPUS IRAPUATO - SALAMANCA
DIVISIÓN DE INGENIERÍAS

"Modelo generativo para imágenes de angiografía coronaria por rayos X basado en puntos clave de la anatomía de los vasos sanguíneos"

TESIS

Que para obtener el Grado de

Maestría en Ingeniería Eléctrica

PRESENTA:

Ing. Jesus Salvador Ramos Cortez

DIRECTORES:

Dr. Juan Gabriel Aviña Cervantes
Dr. Emmanuel Ovalle Magallanes

Salamanca, Guanajuato, Diciembre, 2024

Salamanca, Gto., a **5** de **mayo** de **2025**.

MTRO. JUAN SIGFRIDO LÓPEZ CUSTODIO
COORDINADOR DE ASUNTOS ESCOLARES
P R E S E N T E.-

Por medio de la presente, se otorga autorización para proceder a los trámites de impresión, empastado de tesis y titulación al alumno(a) **Jesus Salvador Ramos Cortez** del ***Programa de Maestría en Ingeniería Eléctrica (Instrumentación y Sistemas Digitales)*** y cuyo número de ***NUA*** es: **144032** del cual soy director. El título de la tesis es: **Modelo generativo para imágenes de angiografía coronaria por rayos X basado en puntos clave de la anatomía de los vasos sanguíneos**

Hago constar que he revisado dicho trabajo y he tenido comunicación con los sinodales asignados para la revisión de la tesis, por lo que no hay impedimento alguno para fijar la fecha de examen de titulación.

A T E N T A M E N T E

Dr. Juan Gabriel Avina Cervantes

NOMBRE Y FIRMA
DIRECTOR DE TESIS
SECRETARIO

Dr. Emmanuel Ovalle Magallanes

NOMBRE Y FIRMA
DIRECTOR DE TESIS

Dra. Dora Luz Almanza Ojeda

NOMBRE Y FIRMA
PRESIDENTE

Dr. Ángel Díaz Pacheco

NOMBRE Y FIRMA
VOCAL

Dedicatoria

Esta tesis la dedico a mi familia, a mis padres, a mis hermanos, pero en especial a mi madre ***Francisca Cortez Mellado*** que es incansable cuando se trata de apoyar a sus hijos, que si no fuera por ella yo no hubiera hecho la maestría y menos este trabajo. Gracias por todo ***mamá***.

Agradecimientos

Quisiera agradecer en primer lugar a un amigo el Dr. Tat'y Mwata Velu que fue el responsable de que yo este escribiendo este documento, porque me animo a hacer la maestría.

Agradezco tener una familia que siempre me apoya sin que yo se los pida, de tener aún conmigo a mi papá, tener a mi mamá que le estoy muy agradecido por todo lo que ha hecho por mí, su esfuerzo, su sacrificio por sacar a sus hijos adelante, gracias mamá. Mis hermanos Emilio, Rodolfo y Gabriela que la vida no tendría gracia y no sería la misma sin ellos, gracias por ser mis hermanos.

Quiero agradecer al Dr. Juan Gabriel Aviña Cervantes y al Dr. Emmanuel Ovalle Magallanes por aceptar trabajar conmigo, tuvieron que soportar mis errores, mis retrasos con el trabajo, pero me tuvieron paciencia. Gracias por permitirme trabajar con ustedes.

Un especial agradecimiento al Dr. José Ruíz Pinales que cuando tuve alguna duda respecto a redes neuronales siempre me brindo su ayuda, gracias.

Por último, quiero agradecer a unos amigos que conocí en la maestría, que convivir con ellos por dos años me hizo apreciarlos en demasía. Son seis personas maravillosas Melanny Julisa Ramírez Lara, Areli Cabrera Oros, Michel Olaf Chacón Carrero, Luis Diego Rendon Aguilar, Jesús Alberto Parada Ramírez, Francisco Javier Castro Sánchez, no digo que todo fue perfecto hubo desacuerdos, riñas, malos entendidos, pero también hubo momentos divertidos como viajes, salidas al cine, a los bolos, las comidas de fin de cuatrimestre y que decir de

las retas de mario kart, de mortal kombat sin dejar atrás los festejos de cumpleaños donde todo fueron risas y alegría, son muy gratos recuerdos. Conocí personas increíbles que aprecio de verdad y me quedo con grandes anécdotas, hay muchas cosas que quisiera decirles, pero no acabaría. Francisco sin duda un gran amigo, mi mejor amigo en la maestría, aunque nos tratemos de la fregada aprecio lo molesto que puede ser, valoro mucho su amistad. Creo sin duda alguna que lo mejor de la maestría fue haberlos conocido a todos, los quiero a todos por igual y les deseo el mayor de los éxitos en el rumbo que tomen. Me alegrará verlos triunfar. Adiós y gracias.

Agradecimientos Institucionales

Expreso mi más sincera gratitud hacia la Universidad de Guanajuato, especialmente a la División de Ingenierías del Campus Irapuato-Salamanca (DICIS) por la formación académica y por el apoyo financiero que he recibido durante mis estudios en esta institución bajo el NUA 144032.

Este trabajo de tesis fue realizado gracias al apoyo invaluable recibido a través de la Secretaría de Ciencia, Humanidades, Tecnología e Innovación, SECIHTI de México, bajo el CVU 1263057.

Índice General

Índice de Figuras

Índice de Tablas

Resumen

Una de las principales causas de muerte en México son las enfermedades cardiovasculares. El diagnóstico de estas enfermedades depende del médico especialista y de su conocimiento para la interpretación de estudios como las angiografías coronarias basadas en rayos X. Los sistemas computacionales buscan de brindar apoyo a los especialistas mediante modelos de *deep learning* entrenados en la detección de enfermedades cardiovasculares por imágenes médicas. Es aquí donde este trabajo de tesis entra en acción. Ya que uno de los problemas de los modelos entrenados en la detección de enfermedades cardiovasculares es la falta de imágenes médicas.

El objetivo de este trabajo es desarrollar un modelo generativo basado en técnicas de *data augmentation* el cual consiste en generar imágenes sintéticas por medio de puntos clave de la anatomía de los vasos sanguíneos. Partiendo de la segmentación de imágenes médicas nos basamos en la arquitectura U-Net, observamos que, al reducir el número de kernel de la U-Net por un factor de escala $\alpha = 4$, se mantiene el desempeño de la red y se reduce el número de parámetros, de 31M a 1.9M. Además, se obtienen mejores resultados en las métricas con mejoras en IoU de 1.46 % (0.76198 a 0.77315), Dice de 0.85 % (0.86277 a 0.87010) y con un incremento en Recall de 1.80 % (0.84741 a 0.86267) y F1-score de 0.80 % (0.86773 a 0.87470); lo que demuestra una mejora en el rendimiento. Para la generalización del esqueleto nos enfocamos en el desarrollo del modelo Multi-task Attention U-Net, con el cual obtuvimos los mejores resultados en segmentación: IoU (0.77335), Dice (0.86954), Recall (0.87634) y F1 Score (0.87393) y para el esqueleto un IoU (0.40551), Dice (0.57252), Recall (0.60339) y F1

Score (0.58014).

Los esqueletos obtenidos fueron procesados morfológicamente para quitar los espacios de píxel a píxel, posteriormente utilizamos el método Shi-Tomasi para la detección de puntos clave, los cuales fueron empleados como centro de kernel que transforman localmente algunas regiones de las imágenes para crear imágenes sintéticas. De este modo, se desarrolla un modelo generativo capaz de generar imágenes nuevas y diferentes a las originales.

Abstract

One of the leading causes of death in Mexico is cardiovascular disease. Diagnosing these diseases depends on the specialist physician and their knowledge of interpreting studies such as X-ray-based coronary angiograms. Computational systems seek to support specialists through deep learning models trained to detect cardiovascular disease through medical images. This is where this thesis comes into play, since one of the problems with models trained to detect cardiovascular disease is the lack of medical images.

The objective of this work is to develop a generative model based on *data augmentation* techniques, which consists of generating synthetic images through key points of the anatomy of blood vessels. Starting from the segmentation of medical images, we rely on the U-Net architecture. We observe that, by reducing the number of U-Net kernels by a scale factor $\alpha = 4$, the performance of the network is maintained and the number of parameters is reduced from 31M to 1.9M. Additionally, improved metrics show improvements in IoU by 1.46 % (0.76198 to 0.77315), Dice by 0.85 % (0.86277 to 0.87010), and Recall by 1.80 % (0.84741 to 0.86267) and F1-score by 0.80 % (0.86773 to 0.87470), demonstrating an improvement in performance. For the generalization of the skeleton we focused on the development of the Multi-task Attention U-Net model, with which we obtained the best results in segmentation: IoU (0.77335), Dice (0.86954), Recall (0.87634) and F1 Score (0.87393) and for the skeleton an IoU (0.40551), Dice (0.57252), Recall (0.60339) and F1 Score (0.58014).

The resulting skeletons were morphologically processed to remove pixel-by-pixel gaps.

We then used the Shi-Tomasi method to detect key points, which were used as kernel centers to locally transform some regions of the images to create synthetic images. In this way, a generative model was developed capable of generating new images that are distinct from the originals.

CAPÍTULO 1

Introducción

En este capítulo abordaremos las razones que nos llevan al surgimiento de la idea que se desarrolla a lo largo de esta investigación de tesis partiendo de la motivación y justificación es que vemos una ventana de oportunidad. A continuación, se presenta el planteamiento de los objetivos, así como la estructura que se seguirá en esta tesis.

1.1. Motivación

En México se estima que miles de personas mueren cada año debido a enfermedades cardiovasculares. Entre las enfermedades cardiovasculares más comunes se encuentran aquellas que se presentan en las arterias coronarias y en las arterias cerebrales, como apoplejía, embolia o derrame cerebral. Las enfermedades en las arterias coronarias se caracterizan por el estrechamiento de las arterias, comúnmente conocido como estenosis [Secretaría de Salud, 2022]. Este padecimiento surge a raíz de la acumulación de grasa en las arterias, lo que dificulta el envío de sangre, oxígeno y nutrientes al músculo cardíaco [Mayo Clinic, 2022].

Sólo en el primer semestre del 2023, se tiene registrado en el INEGI (Instituto Nacional

de Estadística y Geografía) que la primera causa de muerte son las enfermedades del corazón provocando 97,179 casos. Se puede observar en la Figura 1.1 las defunciones debido a enfermedades del corazón en los primeros seis meses de cada año durante el periodo 2020-2023 [INEGI, 2022].

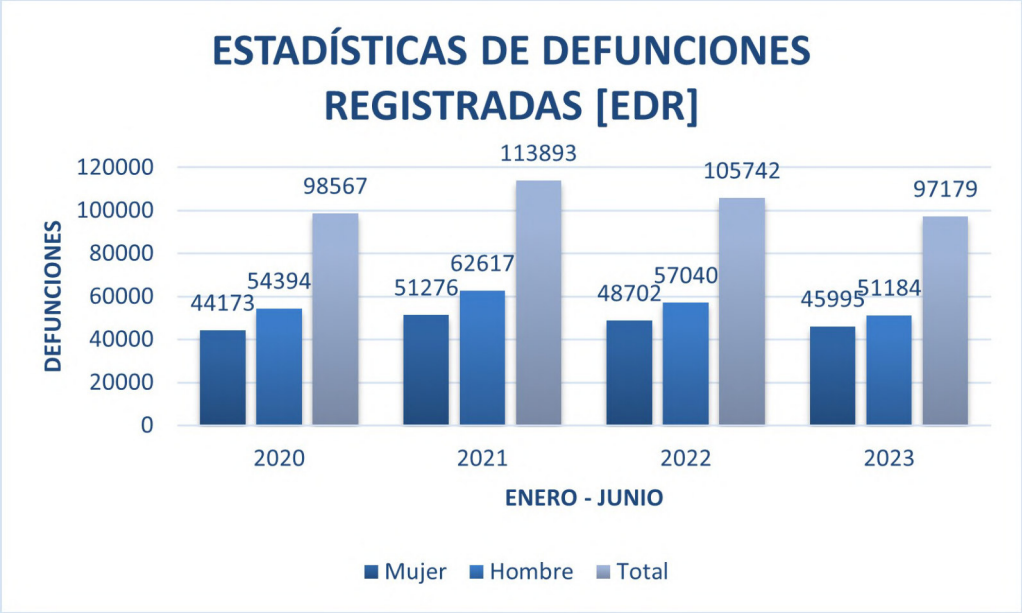


Figura 1.1. Número de muertes por enfermedades del corazón registradas durante el periodo 2020-2023.

Al tener en cuenta que en la sociedad mexicana la principal causa de muerte son las enfermedades del corazón como la estenosis, podemos enfocar esfuerzos para contribuir en la labor del médico de diagnosticar dicha enfermedad de manera más eficiente.

La detección y diagnóstico de la estenosis están en gran medida sujetas a la experiencia y conocimiento del médico para determinar si existe o no un caso de estenosis al interpretar las imágenes de angiografías coronarias, por lo que en muchas ocasiones no cuenta con apoyo de otros medios para corroborar sus diagnósticos. Es por ello, que en la última década han sido propuestos diferentes algoritmos para el diagnóstico asistido por computadora. Entre ellos se destaca principalmente el *Deep Learning* o Aprendizaje profundo, que se caracteriza por aprender patrones de manera automática que permiten identificar áreas de interés como arterias y lesiones [Zhou y cols., 2021]. Sin embargo, estos algoritmos se ven sesgados por la cantidad limitada de imágenes con las cuales fueron entrenados.

1.2. Justificación

Una angiografía coronaria es un estudio basado en rayos X que permite observar los vasos sanguíneos del corazón o arterias coronarias obteniendo imágenes como se muestra en la Figura 1.2, con el propósito de saber si existe alguna obstrucción o estrechamiento en los vasos sanguíneos. Este estudio que realiza el médico pertenece al grupo denominado cateterismo cardíaco. Para llevar a cabo el cateterismo cardíaco, se emplean una o más sondas delgas y huecas llamadas catéteres, las cuales pasan a través de una arteria y se desplazan por los principales vasos sanguíneos y el corazón inyectando un tinte especial (medio de contraste). Esta modalidad de imágenes sigue siendo el estándar para diagnosticar enfermedades en las arterias coronarias [Mayo Clinic, 2024].

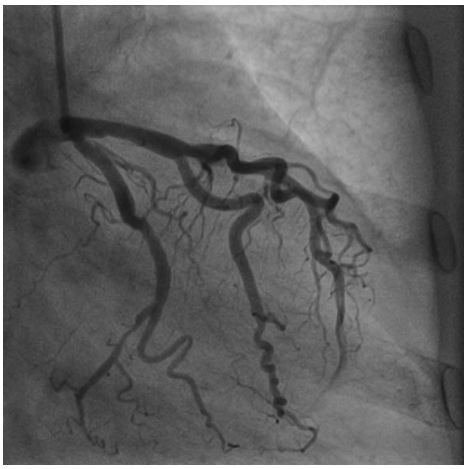


Figura 1.2. Angiografía coronaria de rayos X proveniente de la base de datos de [Zhao y cols., 2021].

Por cuestiones legales y éticas, tener acceso a una vasta cantidad de angiografías coronarias de rayos X de pacientes es costoso y complicado. Por ello, se buscan alternativas para crear bases de datos de imágenes médicas que sirvan al entrenamiento de algoritmos de *Deep Learning* enfocadas en esta modalidad de imágenes.

Comúnmente, el método de *Data Augmentation* o aumento de datos es el más usado, el cual consta de aplicar cierto tipo de transformaciones a las imágenes como transformaciones geométricas [Shorten y Khoshgoftaar, 2019], como la rotación, que consiste en cambiar la orientación de las imágenes; el escalado, que cambia el tamaño de las imágenes; la traslación, que desplaza las imágenes en sus ejes; forma espejo, que invierte la imagen de manera horizontal o vertical; y la perspectiva, que modifica la perspectiva de las imágenes. Además, se tienen transformaciones fotométricas [Dodge y Karam, 2017], las cuales muestran algunas técnicas

que se enfocan en el brillo, el contraste, la luminosidad de las imágenes; el tono, la saturación haciendo que cambien la intensidad del color, la temperatura de color para emular distintas iluminaciones. También existen transformaciones espaciales [Mumuni y Mumuni, 2022], tales como: el recorte, que corta secciones de la imagen; la deformación elástica, para variar la forma de las imágenes; Zoom, que aplica un efecto de alejar o acercar la imagen; enfoque, la iluminación difusa o nítida.

Sin embargo, el reto más relevante de estas técnicas de *Data Augmentation* es poder identificar que transformaciones serían adecuadas para aumentar el desempeño durante el entrenamiento y la inferencia de los algoritmos de *Deep Learning*. En el caso de imágenes médicas, y en particular en las imágenes de angiografía de rayos X, dependiendo de las modificaciones hechas a las imágenes se puede generar distorsiones que generen artefactos no deseados en la imagen, tales como modificar la presencia o proporción de estenosis.

Recientemente, los *Large Language Model* (LLM) o Modelos de Lenguaje Grandes, han surgido como sistemas de inteligencia artificial que procesan y generan texto e imagen a partir de un *prompt* [Naveed y cols., 2023]. Por esta razón, se exploró la posibilidad de generar una imagen de angiografía coronaria de rayos X utilizando distintos LLM, empleando el mismo prompt "X ray coronary angiography image", con la intención de comparar las imágenes resultantes entre ella y la imagen real, como se observa en la Figura Figura 1.3. Lo que se aprecia es que aún no son capaces de generar imágenes sintéticas semejantes a las reales, ya que a simple vista las imágenes generadas por las inteligencias artificiales tienden a parecer ilustraciones poco precisas de la representación de un corazón real y sus vasos sanguíneos.

(a) ***(b)*** ***(c)***

Figura 1.3. Imágnes generadas por IAs: Imagen (a) DaVinci IA, Imagen (b) Picstar e Imagen (c) Canva IA.

1.3. Objetivos

1.3.1. Objetivo General

El desarrollo de un modelo generativo que permita generar imágenes sintéticas de angiografías coronarias de rayos X con alta similitud a las imágenes reales. El modelo generativo se basa en puntos clave que se obtienen a partir de imágenes reales de angiografías, con el propósito de crear base de datos que contribuya al entrenamiento de redes neuronales para mejorar tareas como segmentación de arterias coronarias y la detección de estenosis.

1.3.2. Objetivos Específicos

1. Revisión de la literatura respecto a la segmentación y esqueletización a partir de la base de datos de imágenes de angiografías coronarias de rayos X reales y sus respectivas máscaras de segmentación y esqueletización, todo esto mediante la implementación de un modelo generativo utilizando la red Multi-task U-Net.

2. Revisión de la literatura respecto a los algoritmos de detección de esquinas y bordes, como el método de Shi Tomasi. Ya teniendo el esqueleto de las imágenes se procede a encontrar los puntos iniciales, finales y bifurcaciones como puntos clave.

3. Aplicar transformaciones en los puntos claves obtenidos y así modificar de manera local las imágenes reales para generar imágenes sintéticas que se asemejen a las originales.

1.4. Estructura del documento

- El Capítulo 2 presenta los trabajos del estado del arte relacionados a los problemas de segmentación, la clasificación de los componentes de un vaso sanguíneo, la generación de imágenes sintéticas mediante técnicas de *Deep Learning*

- El Capítulo 3 describe el modelo generativo propuesto para imágenes de angiografía coronaria por rayos X basado en la segmentación y esqueletización de la arteria, así como la extracción de puntos clave de la anatomía de los vasos sanguíneos.

- El Capítulo 4 presenta los resultados numéricos y visuales del modelo generativo propuesto., demostrando su robustez del modelo para las diferentes tareas asociadas: segmentación, esqueletización y generación de imágenes sintéticas.
- El Capítulo 5 incluye las observaciones y conclusiones obtenidas a lo largo del proyecto de maestría, así como las posibles futuras aportaciones derivadas de esta investigación.

CAPÍTULO 2

Trabajos relacionados

En este capítulo se presentan los trabajos relacionados con el tema de esta tesis, el cual se centra en la generación de imágenes sintéticas mediante la implementación de un modelo generativo.

En la actualidad, las ramas de la Inteligencia Artificial, de *Machine Learning* y *Deep Learning* se están volviendo indispensables para el sector salud, principalmente en el apoyo a los médicos mediante sistemas de asistencia médica que mejoren la precisión y los tiempos de diagnóstico. Sin embargo, dichos sistemas requieren de una gran cantidad de datos etiquetados para su entrenamiento, lo que puede generar problemas de generalización si no disponen de suficientes datos.

Recientemente, [Ibrahim y cols., 2024] realizaron un estudio de modelos generativos enfocados en sintetizar diferentes tipos de datos médicos, en los cuales se encuentran los de imágenes médicas, series temporales y datos tabulares (EHR).

Dicho estudio se centra en analizar diferentes modelos generativos desde 2021 utilizando bases de datos como Scopus, PubMed, ArXiv. Los autores presentan una revisión de tres aspectos que consideran relevantes: aplicación de síntesis y propósito de la síntesis, técnicas de

generación y los métodos de evaluación implementados en los modelos.

Una gran herramienta de *Deep Learning* son los modelos generativos, los cuales pueden ser incondicionales o condicionales como se ilustran en la Figura 2.1. Los modelos incondicionales pueden tomar variables aleatorias de entrada como ruido, mientras que los modelos condicionales incorporan una parte de control mediante datos externos a priori (imagen, texto, señales, entre otros), que guíen a la generación de los nuevos datos. En ambos casos, los modelos generativos utilizan el ruido para generar los nuevos datos. Este ruido sigue una distribución probabilística como una distribución gaussiana, conocida como ruido gaussiano con el propósito de proporcionar variabilidad a las muestras.

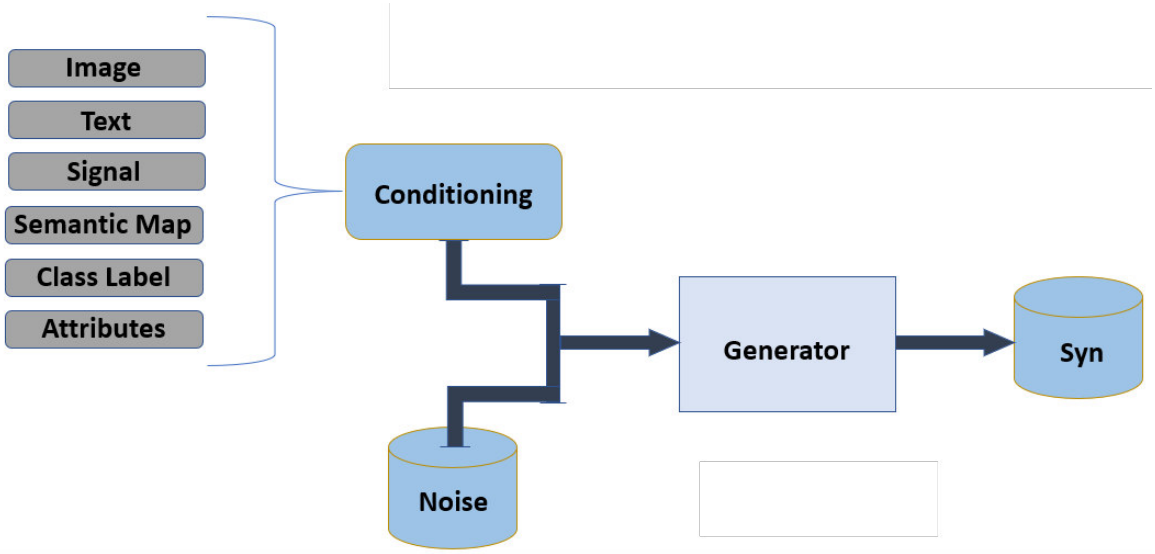
Otro elemento clave de los modelos incondicionales y condicionales es el uso de diferentes arquitecturas de tipo *Generative Adversarial Network* (GAN) las cuales se compone de una red generadora G (*Generator* G) y una red discriminadora D (*Discriminator* D). Diferentes variantes de la arquitectura GAN son analizadas por [AlAmir y AlGhamdi, 2022] para el área médica, indicando que llegan a tener una mejora en la calidad de las imágenes por medio de un mejor entrenamiento. Los avances en las investigaciones señalan que existe una relación entre el tamaño de la base de datos y el tamaño de la red, ya que el rendimiento del entrenamiento depende directamente de la cantidad de datos disponibles. Las GAN han sido utilizadas en imágenes médicas para tareas como la detección, clasificación, aumento y reconstrucción de imágenes.

Sin embargo, existen otros modelos generativos condicionales que no utilizan el ruido como fuente para generar nuevos datos, en su lugar emplean etiquetas, máscaras (*Ground Truth*), que, como segunda entrada en lugar de ruido, condicionan el entrenamiento del modelo. Estos modelos se basan principalmente en la red U-Net, que emplea imágenes para su entrenamiento y utiliza máscaras como su *Ground Truth*. Su arquitectura está diseñada para extraer las características más relevantes a la vez que conserva la información contextual, por lo cual es implementada para la segmentación de imágenes [Mohammed y Clarke, 2024].

Por otra parte se encuentran los modelos de difusión, son modelos efectivos para el aprendizaje de datos complejos, los modelos de difusión se centran primordialmente en la generación de imágenes, eso da la posibilidad de generar imágenes médicas sintéticas 2D o 3D. [Kazerouni y cols., 2022] hablan sobre modelos de difusión con aplicaciones como el registro, la eliminación de ruido, la clasificación, la reconstrucción de imágenes entre otras técnicas que pueden ser implementadas en el campo de las imágenes médicas.

[Kora Venu y Ravula, 2020] proponen mediante la implementación de modelos

(a) Unconditional



(b) Conditional

Figura 2.1. El diagrama (a) muestra los modelos incondicionales y el diagrama (b) muestra los modelos condicionales. Diagramas inspirados de [Ibrahim y cols., 2024]

generativos basados en *Deep Convolutional Generative Adversarial Network* (DCGAN), los cuales son una modificación de la GAN original, donde tanto la red generadora como la red discriminadora usan capas convolucionales y capas convolucionales transpuestas como se muestra en la Figura 2.2, tienen como objetivo la creación de imágenes sintéticas de radiografías de tórax para hacer data augmentation, para posteriormente evaluar la calidad de las imágenes generadas mediante la Ecuación (2.1) Fréchet Inception Distance (FID), la cual se presenta a continuación,

$$d_{\text{FID}}(x, g) = \|\mu_x - \mu_g\|^2 + \text{Tr}\left[\Sigma_x + \Sigma_g - 2(\Sigma_x\Sigma_g)^{\frac{1}{2}}\right], \tag{2.1}$$

donde μ_x y μ_g representan la media de las características tanto de las imágenes reales como sintéticas, donde Σ_x y Σ_g son sus respectivas matrices de covarianza y Tr representa la suma de todos los elementos de la matriz diagonal.

Para la realización de datos sintéticos de radiografías de tórax [Ng y Hargreaves, 2023] tratan de centrarse en el uso de las GAN y analizan como los datos obtenidos son

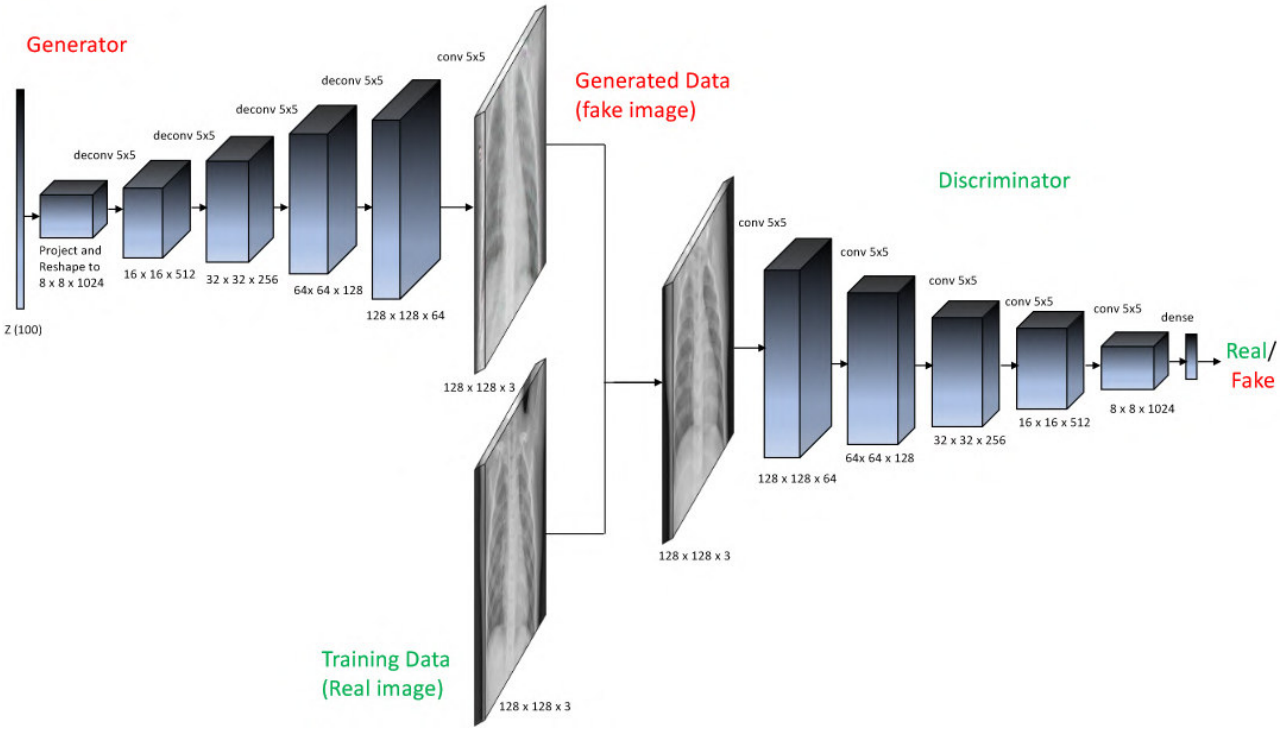


Figura 2.2. Diagrama de la DCGAN implementada para la generación de imágenes sintéticas de radiografías de tórax recuperado de [Kora Venu y Ravula, 2020].

afectados por el tamaño del entrenamiento y posteriormente comparan el rendimiento de diferentes arquitecturas GAN como DCGAN y *Wasserstein Generative Adversarial Networks with Gradient Penalty* (WGAN-GP) evaluando la calidad de las imágenes mediante Fréchet Inception Distance (FID).

Referente a la generación de imágenes de vasos sanguíneos, han existido diversos trabajos que implementan técnicas en sintetizar imágenes. Tal es el caso de [Wolterink y cols., 2018] quienes proponen utilizar una red (GAN) para modelar vasos sanguíneos sintéticos. La red generadora G transforma un vector de ruido z que es muestreado de una distribución p_z en una parametrización $1D$ de cuatro canales, $G(z)$ de una arteria coronaria. La red discriminadora D hace una comparación entre las geometrías de arterias coronarias reales y sintéticas con la intención de predecir una puntuación alta para arterias reales y una puntuación baja para arterias sintéticas a medida que se entrenan las redes, la red discriminadora no pueda distinguir una imagen de una arteria real de una arteria sintética. El flujo de datos de esta metodología se ilustra en la Figura 2.3.

Para el entrenamiento condicional, el generador trata de minimizar la función objetivo mientras que el discriminador intenta maximizarla como se muestra en la Ecuación (2.2)

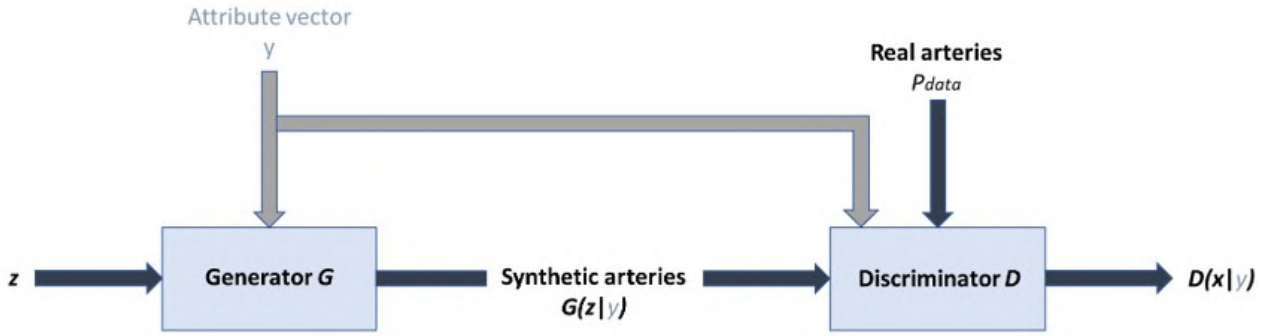


Figura 2.3. Diagrama del método inspirado de [Wolterink y cols., 2018].

$$\min_{G} \max_{D \in \mathcal{D}} V^{(D)}(D,G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \left[D(\mathbf{x}|\mathbf{y}) \right] - \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}} \left[D(G(\mathbf{z}|\mathbf{y})|\mathbf{y}) \right] \ldots \\ -\lambda \mathbb{E}_{\hat{\mathbf{x}} \sim p_{\hat{\mathbf{x}}}} \left[\left(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1 \right)^2 \right], \tag{2.2}$$

donde G es el generador, D es el discriminador, x son los datos reales de la distribución $Pdata$, z es el vector de ruido muestreado de la distribución P_z, y es el vector de atributos donde se condicionan G y D, $\hat{x}$ es la interpolación de los datos reales x con los datos generados $G(z|y)$, λ es un factor para regularizar en la función de pérdida y $\mathbb{E}$ representa la esperanza de la función objetivo.

Un método enfocado en la reconstrucción de arterias coronarias en 3D basado en *Deep Learning* es presentado por [Hwang y cols., 2021], el cual consiste en la identificación de los dos puntos finales de un vaso sanguíneo partiendo de la angiografía coronaria de rayos X, junto con un método de utilización de *Template Model* (plantillas) de angiografías coronarias para que coincidan los vasos sanguíneos segmentados bidimensionales usando dos ángulos diferentes de la angiografía coronaria. Implementan una red U-Net conformada por un *encoder* (Resnet) y un *decoder* como se puede apreciar en la Figura 2.4.

Entrenaron la U-Net para la segmentación de los vasos sanguíneos con el propósito de construir las arterias en 3D a partir de las imágenes 2D de dos ángulos diferentes. Implementaron el filtro Frangi, para mejorar las segmentaciones al resaltar las estructuras de los vasos sanguíneos. Les permite determinar el área donde existe el material de contraste para poder determinar la línea central del vaso sanguíneo implementan el algoritmo de *fast-marching* para encontrar la mejor ruta del punto de inicio al punto final.

En la identificación de los puntos finales del vaso sanguíneo en las imágenes obtenidas de dos ángulos diferentes, las imágenes de entrenamiento se crearon etiquetando los dos puntos

finales del vaso. El área proximal del vaso se etiquetó en la punta del catéter y el área distal del vaso se etiqueto en el punto final distal del material de contraste, cuando el material lleno por completo el vaso.

Para afrontar el problema de escorzo, que es la distorsión que puede existir en las imágenes de angiografías al establecer la correspondencia entre los vasos sanguíneos segmentados en 2D desde dos ángulos diferentes, utilizaron modelos de plantillas de las líneas centrales de las arterias coronarias tanto izquierda como derecha. Las líneas centrales representan la estructura del vaso sanguíneo, los modelos de plantillas son construidos a partir de promediar los puntos de la línea central de 10 modelos que eligieron de manera aleatoria. Estos 10 modelos fueron generados usando un software comercial (Autoseg).

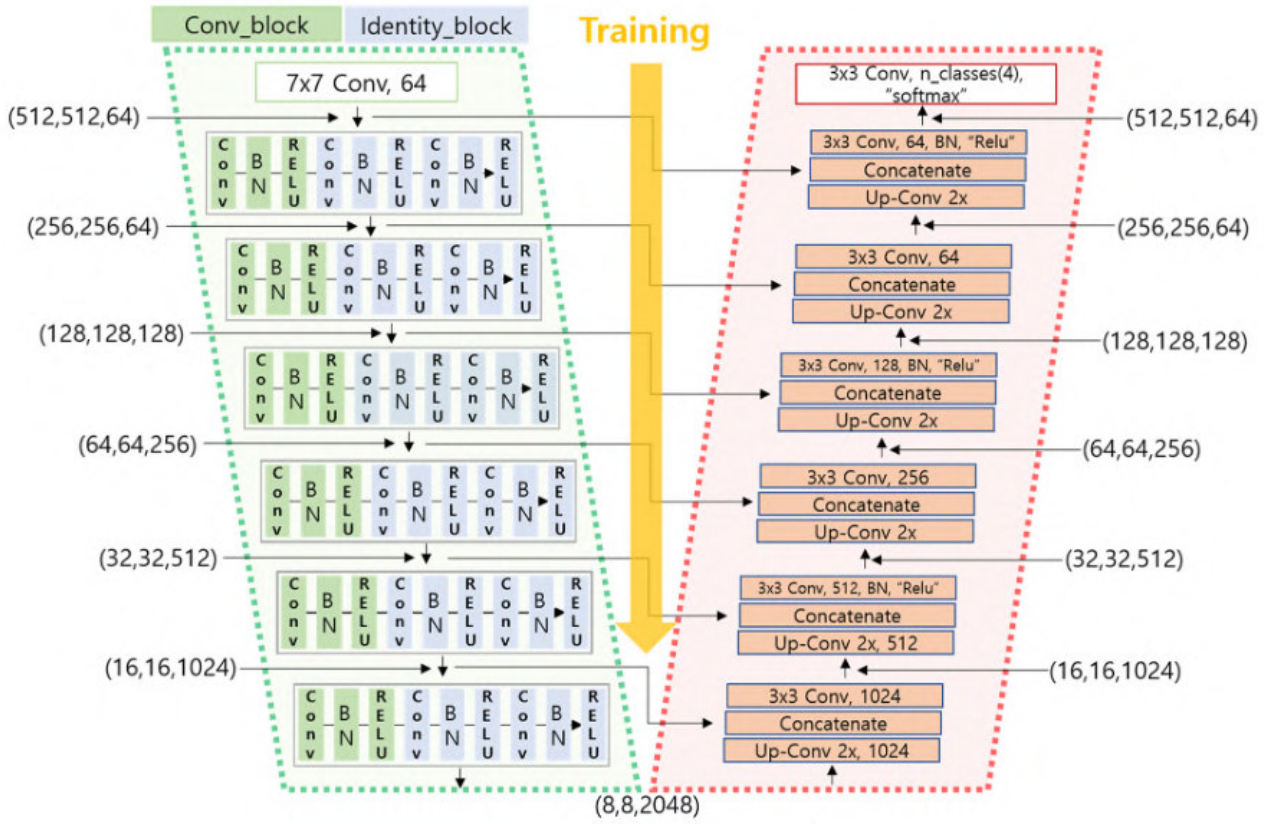


Figura 2.4. En el diagrama recuperado de [Hwang y cols., 2021] se muestra en verde la parte del encoder (Resnet) y en rojo la parte del decoder para el entrenamiento de las imágenes etiquetadas.

Con el objetivo de reconstruir el vaso sanguíneo tridimensional emplearon el método de retro-proyección el cual consiste en que cada punto de la línea central del espacio 3D se obtiene mediante relaciones geométricas entre las coordenadas de los puntos en las imágenes 2D y la

ubicación de la fuente de rayos X en los dos ángulos.

Para analizar y reconocer angiografías coronarias para segmentación como identificación de las morfologías de lesiones como las medidas del diámetro de la lesión de estenosis, la ponderación, la trombosis, la oclusión total y como la detección de disección en una angiografía de entrada [Du y cols., 2021] proponen un sistema DeepDiscern. Para la parte del reconocimiento de arterias coronaria entrenaron una *Deep Neuronal Network* (DNN) mediante la modificación de una *conditional Generative Adversarial Network* (cGAN).

Extrajeron características que contienen información semántica, mediante el generador las angiografías de entrada se muestrean a una escala menor y luego a una escala mayor para obtener características a diferente escala, se juntan mediante concatenación para obtener la información semántica, los resultados obtenidos pasan en el discriminador junto con su *ground truth* y son re-dimensionados a distintas escalas. Los datos pasan por varias capas convolucionales para tener el resultado del discriminador, obteniendo información como los bordes de vasos sanguíneos, la textura del fondo, características de bajo nivel y alto nivel para la segmentación de imágenes implementando la función de pérdida de la red GAN, representada matemáticamente por

$$\min_{G} \max_{D} L_{GAN}(G, D) = \mathbb{E}_{i,o}[\log D(i, o)] + \mathbb{E}_{i}[\log(1 - D(i, G(i)))], \tag{2.3}$$

donde G es el generador, D es el discriminador, i es la imagen de angiografía, o es la imagen resultante de segmentación y $\mathbb{E}$ representa la esperanza de la función objetivo.

En la detección de la morfología de las lesiones, desarrollaron una DNN convolucional que toma la entrada de la angiografía coronaria y genera la localización de las coordenadas superior izquierda e inferior derecha del área rectangular predicha. Utilizan bloques residuales profundos, con capas de muestreo ascendente y conexiones laterales para la extracción de diferentes escalas de características de diferentes lesiones. Al tener los mapas de características utilizan el generador *Region Proposal Network* (RPN) para generar propuestas de regiones donde puedan ocurrir morfologías de lesiones. Posteriormente las características de las regiones propuestas son pasadas por capas convolucionales completamente conectadas y así predecir el tipo y localización de las morfologías de las lesiones.

Siguiendo un enfoque alternativo [Maccagnan y cols., 2023] presentan un conjunto de herramientas habituales de manera alternativa a los métodos de data augmentation habituales. El método que proponen, denominado “Toolbox for vessel x-ray angiography images simulation” representan los vasos sanguíneos mediante la implementación de ecuaciones que

utilizan coordenadas de un plano cartesiano para la anchura y altura, las cuales permiten generar imágenes de angiografía de rayos X sintéticas a través del esbozo de trayectorias de vasos sanguíneos presentando rasgos de estenosis y aneurismas.

Mediante el uso de variables aleatorias, gradientes y ruido blanco, buscan darles versatilidad y autenticidad a las imágenes. Las variables aleatorias permiten la existencia de diferencias entre una imagen y otra, los gradientes pude dar pequeños cambios de iluminación o del entorno simulando el realismo en la imagen y el ruido blanco añade imperfecciones haciendo que se vean más naturales. Todo esto tiene como objetivo el crear imágenes sintéticas y la intención de mejorar las bases de datos para implementar en algoritmos que requieran imágenes médicas para su entrenamiento.

La Tabla 2.1 muestra una recopilación de los diferentes métodos presentados en este capítulo que son empleados para la generación de imágenes y que se han desarrollado en los últimos años.

Al analizar los trabajos relacionados con la generación de imágenes sintéticas, observamos que la mayoría de los modelos generativos como las GANs o los modelos de difusión, dependen en gran medida de bases de datos muy grandes para tener una buena generación de imágenes. Haciendo que los procesos de entrenamiento sean más largos y costosos computacionalmente.

Además, al depender del ruido aleatorio, complica la generación de detalles en las imágenes para representar las formas complejas como son los vasos sanguíneos. Vemos como una oportunidad usar modelos que implementen máscaras y no dependan del ruido aleatorio, con el reto de mejorar la calidad de las imágenes simplificando una red básica como la U-Net. Mientras que los modelos generativos tradicionales se centran en aproximaciones geométricas, nuestra propuesta se basa en la extracción de puntos clave a través de la línea central de la imagen predicha.

Otro aspecto que hemos detectado es que muchos de los modelos generativos se centran en una sola tarea a la vez y hay muy pocos trabajos que se dedican en hacer tareas en simultaneo para imágenes médicas. En esta tesis, presentamos un modelo generativo que implementa una red U-Net modificada para realizar tareas en simultaneo (Multi-task Attention U-Net), generando imágenes de segmentación e imágenes de la línea central de la segmentación (conocida como esqueleto de la segmentación) en angiografías coronarias de rayos X. Estas imágenes se utilizan para la extracción de puntos clave que posteriormente son empleados en la generación de imágenes sintéticas por medio de transformaciones locales en dichos puntos.

Tabla 2.1. Métodos generados para el análisis de imágenes médicas.

Artículo	**Red**	**Parámetros**	**Dataset**
Generative AI for Synthetic Data Across Multiple Medical Modalities [Ibrahim y cols., 2024]	Diferentes GANs, DDPM y Stable Diffusion	Diferentes parámetros en cada tipo de red generativa	ISIC, Fitzpatrick 17k, BCN10000, CBIS-DDSM, MIAS, OPTIMAM y EHR.
The Role of generative adversarial network in medical image analysis: An in-depth survey [AlAmir y AlGhamdi, 2022]	Diferentes GANs	Diferentes parámetros en cada tipo de red generativa	SpineWeb, TCIA, Disease Neuroimaging Initiative (ADNI), BraTS, Iseg2017, MRBrain13
Conditional image-to-image translation generative adversarial network (cGAN) for fabric defect data augmentation [Mohammed y Clarke, 2024]	GAN, Generator (conditional U-Net), Discriminator (PatchGAN)	Learning rate = 0.0001, parámetros de momento $\beta_1 = 0.5$ y $\beta_2 = 0.999$, 100 épocas	El conjunto de datos AITEX original consta de 245 imágenes
Diffusion models for medical image analysis: A comprehensive survey [Kazerouni y cols., 2022]	Diferentes modelos de difusión	No especifica	No especifica
Evaluation of deep convolutional generative adversarial networks for data augmentation of chest x-ray images [Kora Venu y Ravula, 2020]	DCGAN	No especifica	Kermany entrenamiento 5216 (1341 Normal y 3875 Neumonía), 16 validación y 624 test
Generative adversarial networks for the synthesis of chest x-ray images [Ng y Hargreaves, 2023]	DCGAN, WGAN-GP	Optimizador Adam, batch zise 128, leraning rate = 0.0002, $\beta_1 = 0.5$, 500 épocas / learning rate = 0.0001, $\beta_1 = 0.5$ y $\beta_2 = 0.9$, 700 épocas	4 datasets de 500, 1000, 1500 y 2000 radiografía de tórax COVID-19 / 1000 radiografía de tórax COVID-19
Blood vessel geometry synthesis using generative adversarial networks [Wolterink y cols., 2018]	GAN	Mini-batch de 64 muestras reales y 64 sintéticas, Optimizador Adam, learning rate = 0.0001, 200,000 iteraciones	4.412 líneas centrales de arterias coronarias reales con mediciones de radio
A simple method for automatic 3D reconstruction of coronary arteries from X-ray angiography [Hwang y cols., 2021]	U-Net encoder(Resnet) y decoder	Optimizador Nadam, learning rate = 0.0001, Batch size de 4 y 150 épocas	2342 (LAD), 1907 (LCX) y 1523 (RCA) imágenes etiquetadas
Training and validation of a deep learning architecture for the automatic analysis of coronary angiography [Du y cols., 2021]	DeepDiscern DNN, cGAN	Learning rate = 2×10^5 que baja 10^6, pesos $\alpha = 0.5$ y $\beta = 10$, 400 épocas, optimizador SGD, mini-bach de dos imágenes y 256 anclajes por imagen	20.612 angiografías de 10.073 pacientes
Toolbox for vessel X-ray angiography images simulation [Maccagnan y cols., 2023]	No especifica	No especifica	No especifica

Este enfoque tiene como objetivo ofrecer un método alternativo a los métodos convencionales para hacer *data augmentation.*

CAPÍTULO 3

Metodología

Este capítulo se centra en la descripción detallada del modelo generativo propuesto, el cual consta de tres etapas principales:

1. Etapa supervisada: se entrena una CNN para la segmentación de arterias y su esqueletización.
2. Extracción de puntos clave: a partir de la red entrenada, se realizan inferencias para obtener los puntos clave del esqueleto de la arteria.
3. Generación de imágenes sintéticas: se aplican deformaciones locales alrededor de los puntos clave para generar imágenes sintéticas.

Este flujo de trabajo se puede observar en la Figura 3.1.

Los experimentos realizados se ejecutaron en los entornos de Google Colab con la GPU (Graphics Processing Unit) A100, L4 y en un equipo personal de computo con GPU GTX 1650 utilizando Python versión 3.10.13, PyTorch versión 2.1.2 y TorchVision versión 0.16.2

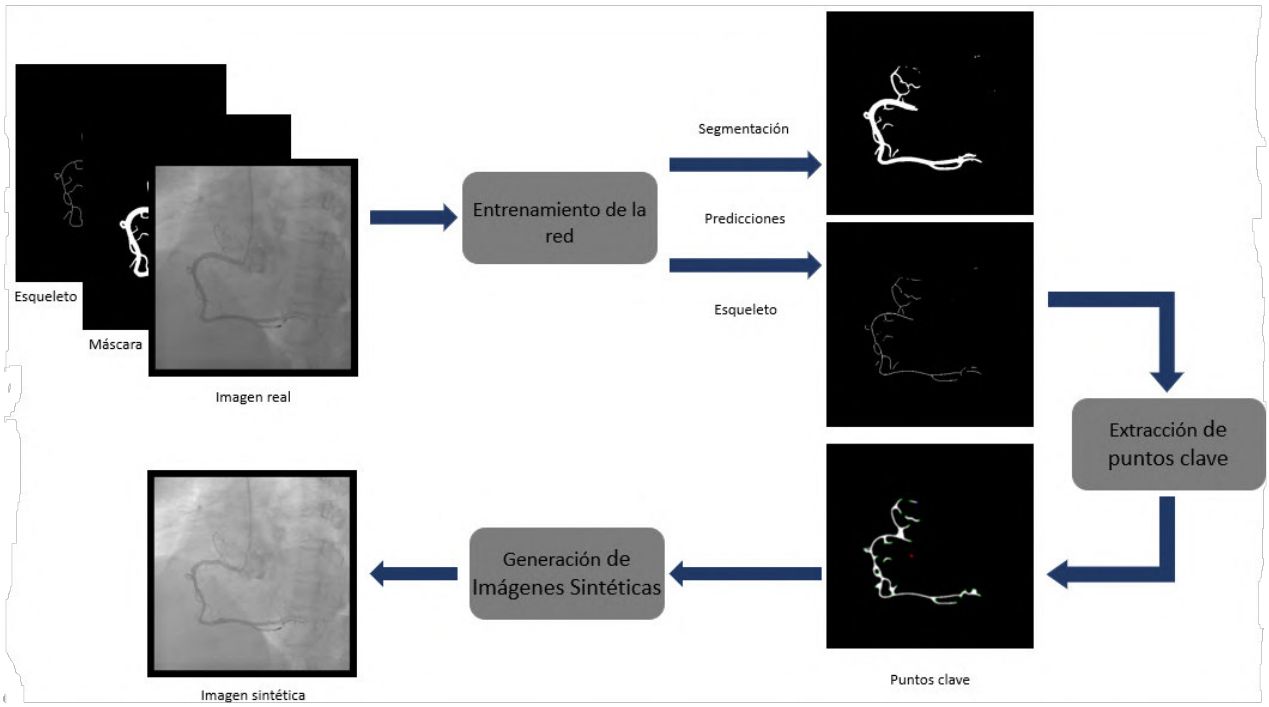
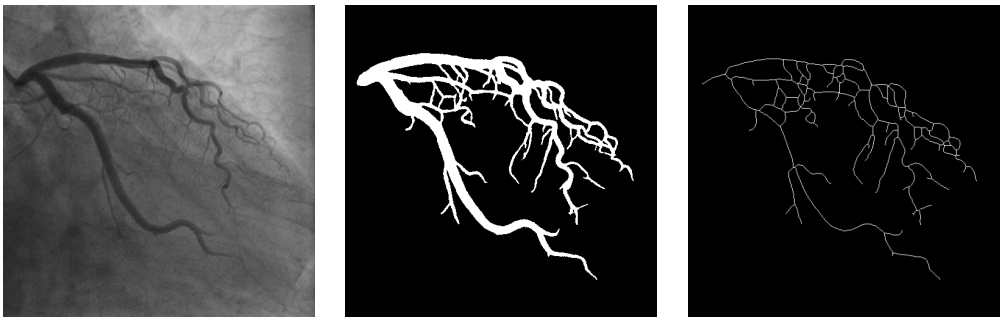


Figura 3.1. Estructura del modelo generativo de imágenes de angiografía coronaria de rayos X.

3.1. Base de datos

La base de datos está constituida de angiografías basadas en rayos X, las cuales fueron liberadas por [Zhao y cols., 2021]. La base de datos corresponden a un estudio que incluyó a 99 pacientes a quienes se les realizó *Invasive Coronary Angiography* (ICA) entre el 26 de febrero y el 18 de julio de 2019. El ICA fue llevado a cabo mediante un sistema de angiografía intervencionista (AXIOM-Artis, Siemens, Múnich) donde se adquirieron a 15 cuadros por segundo en el Hospital Popular de la Provincia de Jiangsu, China. Las imágenes fueron escaneadas con un tamaño de 512×512 pixeles, con un espaciado entre píxeles variable de 0,258 mm a 0,390 mm. El estudio fue aprobado por el comité de ética del Primer Hospital Afiliado de la Universidad Médica de Nanjing.

En total, la base de datos consta de 616 imágenes de angiografías coronarias y sus respectivas 616 máscaras de segmentación o *ground truth*. De las 616 imágenes que angiografías coronarias tenemos 405 LCA y 211 RCA con la misma cantidad de máscaras de segmentación. Además, para nuestro estudio, generamos la esqueletización de las máscaras, como se observa en la Figura 3.2.



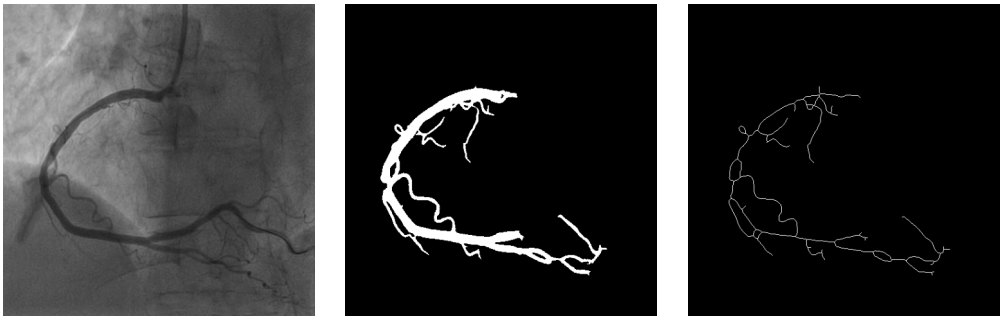


Figura 3.2. Imagen de *Left Coronary Artery* (LCA) e imagen de *Right Coronary Artery* (RCA).

Esta base de datos no cuenta con particiones predefinidas, por lo que se crean de manera aleatoria y estratificada tres *subsets*, uno para entrenamiento, uno para validación y otro para pruebas. La distribución de los subsets es la siguiente: 60 % (369 imágenes) para entrenamiento, 10 % (61 imágenes) para validación y 30 % (186 imágenes) para pruebas.

El *subsets* de entrenamiento (369 imágenes) fue sometida a *data augmentation*, aplicándole técnicas como espejo vertical y horizontal, así como rotación a diferentes grados, generando un total de 2583 imágenes para el entrenamiento. Estas trasformaciones fueron seleccionadas con la finalidad de no modificar la morfología de la artería y crear artefactos no deseados. La Figura 3.3 ilustra un ejemplo de este tipo de técnicas de *data augmentation*.

Es importante resaltar que tanto para el *subset* de validación como para el de pruebas no se les aplicó ninguna técnica de *data augmentation*.

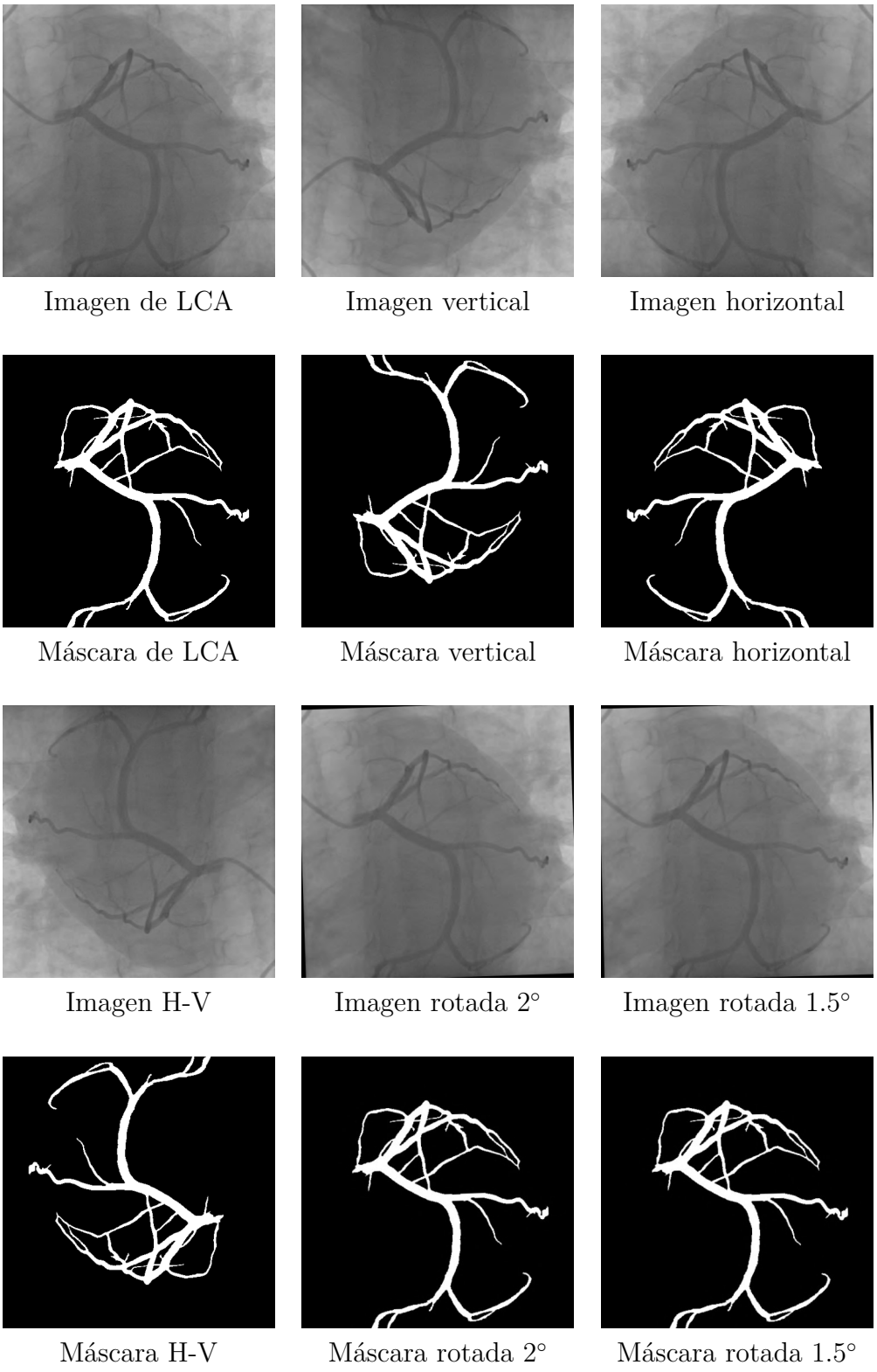


Figura 3.3. Angiografia coronaria y su máscara con data augmentation.

3.2. Vanilla U-Net

La arquitectura U-Net, propuesta por [Ronneberger y cols., 2015], es el modelo estándar para tareas de segmentación de imágenes. El modelo base (*Vanilla U-Net*) está conformada por un *encoder* (E) el cual consiste en un camino de contracción encargado de extraer características jerárquicas de la imagen de entrada $\mathbf{X}$, y un *decoder* que es un camino de expansión que reconstruye la salida $\mathbf{Y}$ mediante el aumento de resolución y la combinación de características desde E (usando conexiones de salto).

Tenemos entonces que el modelo U-Net puede entenderse como una función matemática que mapea una imagen de entrada $\mathbf{X}$ a una salida $\mathbf{Y}$. Esto se puede expresar como:

$$\mathbf{Y} = \mathbf{D}(\mathbf{E}(\mathbf{X})). \tag{3.1}$$

El *encoder* E, consiste de una arquitectura convolucional conformada por la aplicación de dos convoluciones simultaneas con un filtro de tamaño 3×3. A cada una le sigue una función de activación *Rectified Linear Unit* (ReLU) y una operación de *max pooling* de tamaño 2×2 con *stride* de 2 para el descenso. En cada bloque de descenso, se duplica el número de filtros. En total, se tienen cuatro bloques de descenso con 64, 128, 256 y 512 filtros, respectivamente.

El proceso de *encoding* se puede expresar como:

$$\mathbf{E}(\mathbf{X}) = \{\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4\}, \tag{3.2}$$

donde $\mathbf{F}_i$ son los mapas de características en el nivel i, obtenidos mediante:

$$\mathbf{F}_i = \begin{cases} f_{\text{pool}}(f_{\text{relu}}(f_{\text{conv}}(\mathbf{X}))) & \text{si } i = 1, \\ f_{\text{pool}}(f_{\text{relu}}(f_{\text{conv}}(F_{i-1}))) & \text{si } i = 2, 3, 4. \end{cases} \tag{3.3}$$

donde: $f_{\text{conv}}(\cdot)$ es la operación de convolución, $f_{\text{relu}}(\cdot)$ es la función de activación ReLU, que introduce no linealidad y $f_{\text{pool}}(\cdot)$ es la operación de reducción de resolución de max-pooling.

La función de activación permite a las redes neuronales aprender patrones complejos de los datos al introducir no linealidad. Si tenemos la salida de la i-ésima capa convolucional, $\mathbf{X_i^{out}}$, pasa por una función de activación, está la mapea a un rango predeterminado $\mathbf{X}_i^{out}$ antes de ser enviada a la siguiente capa convolucional.

La función de activación ReLU entrega a su salida 0 para valores negativos y devuelve la misma entrada para valores positivos. Se expresa matemáticamente de la siguiente forma:

$$f_{relu}(\mathbf{X}_i^{out}) = \mathbf{max}(0, \mathbf{X}_i^{out}) \tag{3.4}$$

donde para cada valor del mapa de características, si su valor es 0 o negativo la función entrega 0 pero si es un valor positivo devuelve la entrada, gráficamente se puede visualizar en la Figura 3.4.

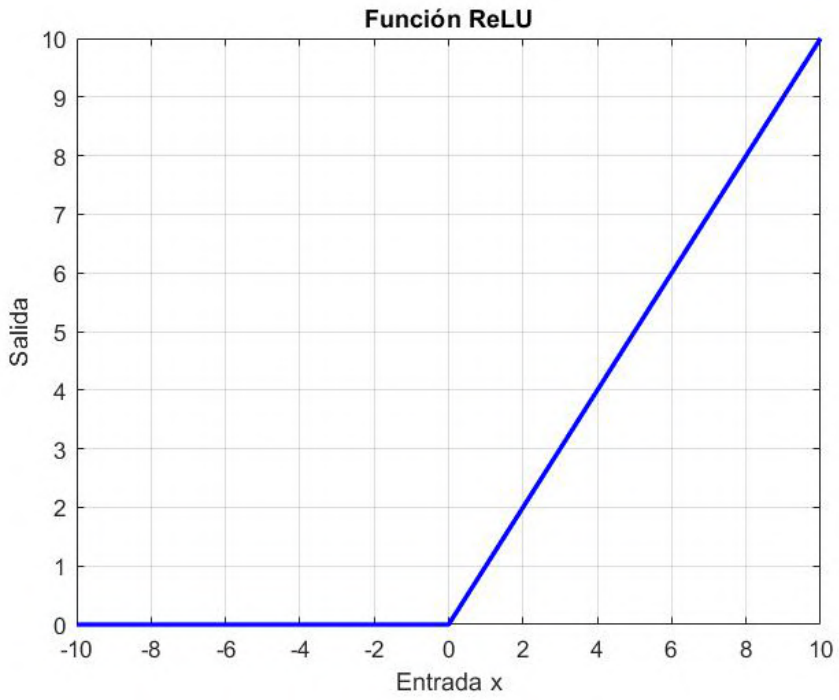


Figura 3.4. Función de activación ReLU.

El *pooling* es una operación que generalmente se aplica sobre el mapa de características **F**, obtenido después del proceso de convolución y activación. Su función es analizar el contenido de **F** mediante regiones con el fin de extraer información representativa. Particularmente, el *max pooling* consiste en dividir en regiones de igual tamaño denominadas kernel, y para cada región, extraer el valor máximo correspondiente a un píxel en el mapa de características **F** resultante. Matemáticamente, para cada canal de **F** se expresa de la siguiente forma:

$$\mathbf{P}[u, v] = \mathbf{max}\left\{\mathbf{F}[i, j] \mid i \in \{u \cdot s, \ldots, u \cdot s + k - 1\},\ j \in \{v \cdot s, \ldots, v \cdot s + k - 1\}\right\} \tag{3.5}$$

donde **P** es la matriz de la salida del *max pooling*.

Una representación gráfica de la operación se muestra en la Figura 3.5 donde se ilustra un ejemplo de una imagen de 5×5 y se le aplica *max pooling* con un kernel de 2×2 y un *stride* de dos.

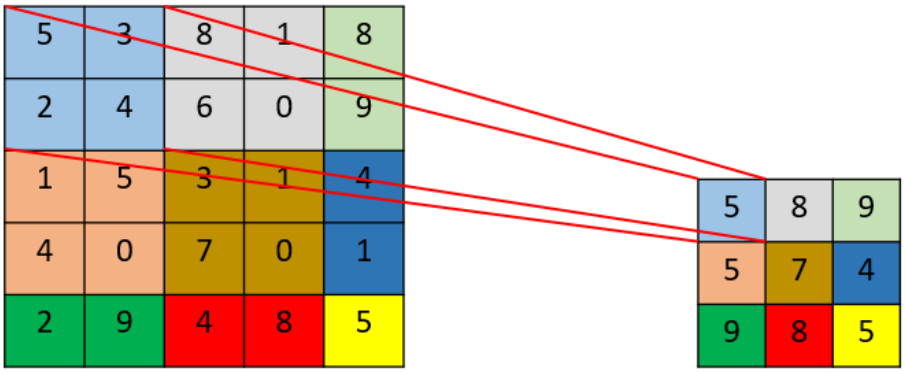


Figura 3.5. *Max pooling.*

Se puede aprecia por los colores las regiones que son utilizadas para la operación de *Max pooling*.

La transición del *encoder* al *decoder* pasa por el cuello de botella de la red o *Bottleneck*, donde la última capa del *encoder* tiene 512 filtros. a esta se le aplica otra convolución para generar 1024 filtros. En este punto, comienza la parte del *decoder*, donde se hace una convolución ascendente, lo que provoca que se reduzcan los filtros a 512. Para compensar esta reducción, se realiza la concatenación del mapa de características del *encoder* que se encuentra en la misma posición con el bloque del *decoder*, recuperando así los 1024 filtros.

Por su parte, el *decoder*, o el camino de expansión, consiste del muestreo del mapa de características obtenido en la parte del *encoder*. Posteriormente, se aplican convoluciones de 2×2 de manera ascendente, reduciendo a la mitad el número de filtros. En cada nivel, se conectan por concatenación con el mapa de características del *encoder* seguidos de dos convoluciones simultaneas, con un filtro de tamaño 3×3 y la función de activación ReLU. En la parte final de la red (head), se implemente una convolución 1×1 que no modifica la resolución, pero reduce los filtros a dos los cuales son las clases para cada pixel de la imagen.

El proceso de *decoding* se puede expresar como:

$$\mathbf{D}(\mathbf{E}) = \{\hat{\mathbf{F}}_4, \hat{\mathbf{F}}_3, \hat{\mathbf{F}}_2, \hat{\mathbf{F}}_1\}, \tag{3.6}$$

donde $\hat{\mathbf{F}}_i$ son los mapas de características reconstruidos en el nivel i, obtenidos mediante:

$$\hat{\mathbf{F}}_i = \begin{cases} f_{\text{conv}}(f_{\text{relu}}(f_{\text{concat}}(f_{\text{up}}(\mathbf{E}_4), \mathbf{F}_4))) & \text{si } i = 4, \\ f_{\text{conv}}(f_{\text{relu}}(f_{\text{concat}}(f_{\text{up}}(\hat{\mathbf{F}}_{i+1}), \mathbf{F}_{i+1}))) & \text{si } i = 3, 2, 1, \end{cases} \tag{3.7}$$

donde $f_{\text{up}}(\cdot)$ es la operación de *up-convolution* (o convolución transpuesta) para aumentar la resolución espacial, $f_{\text{concat}}(\cdot, \cdot)$ denota la concatenación de los mapas de características del *encoder* y el *decoder* en el nivel correspondiente, $f_{\text{relu}}(\cdot)$ es la función de activación ReLU y $f_{\text{conv}}(\cdot)$ es la operación de convolución, que refina los mapas de características reconstruidos.

Por lo que la red base original consta de 27 capas y un total de 31,042,369 parámetros. Se puede ver de mejor manera en la Figura 3.6 como está conformada la estructura de la red.

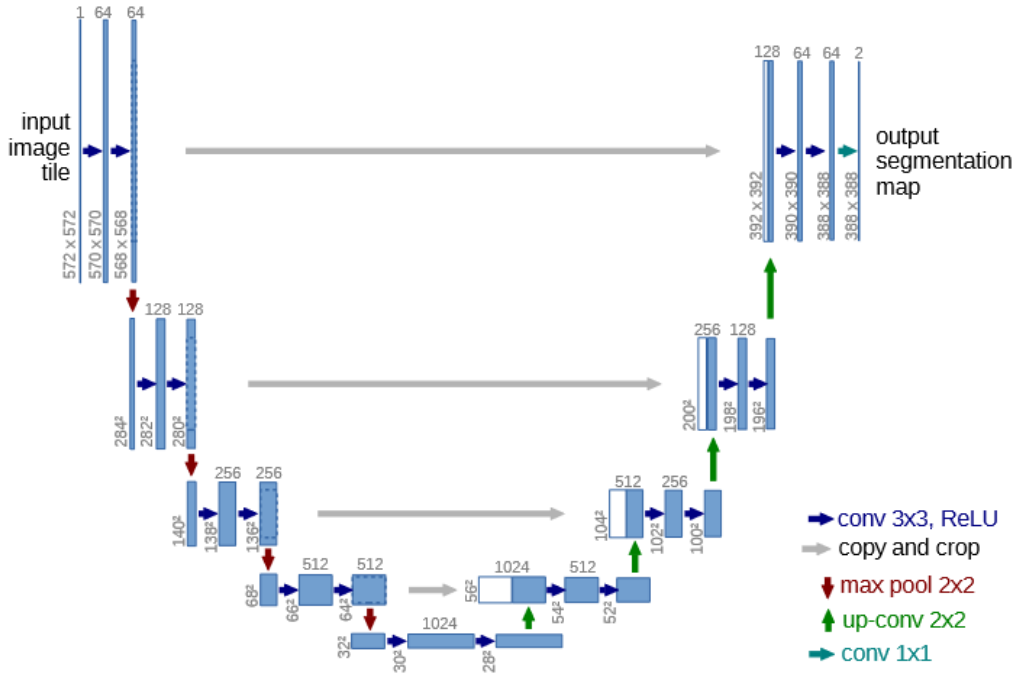


Figura 3.6. Estructura de la red U-Net creada por [Ronneberger y cols., 2015].

3.3. Multi-task Attention U-Net

Partiendo de la arquitectura de la U-Net como base, se pretende robustecer el proceso de segmentación mediante el uso de la imagen esqueletonizada de cada máscara o *ground truth*. De esta manera, la U-Net realizará dos tareas: segmentar la artería y obtener su esqueleto. La arquitectura general propuesta (Multi-task Attention U-Net) se puede observar en la Figura 3.7.

Además, la arquitectura Multi-task Attention U-Net contempla la reducción del número de 64 filtros iniciales hasta un factor de escala, con el objetivo de reducir el costo computacional (31,042,369 de parámetros).

La Multi-task Attention U-Net implementa el mismo *encoder* que la U-Net, pero con ciertas modificaciones, tales como: una capa de normalización, la modificación de la función de activación, el cambio de *max pooling* y la incorporación de un módulo de atención. Por su parte, el *encoder* se conecta de manera independiente a dos *decoders*, uno utilizado para la segmentación de la arteria y otro para obtener el esqueleto. Por lo tanto, usando 16 filtros de inicio, la arquitectura está conformada de un total de 2,793,546 parámetros. Cada uno de los componentes se describen a detalle en las sub-secciones siguientes.

Podemos definir la arquitectura Multi-task Attention U-Net de la siguiente manera,

$$\mathbf{Y_{Mt}} = \begin{bmatrix} \mathbf{Y}_{S1} \\ \mathbf{Y}_{S2} \end{bmatrix} = \begin{bmatrix} \mathbf{D}_{S1}(\mathbf{E}(\mathbf{X})) \\ \mathbf{D}_{S2}(\mathbf{E}(\mathbf{X})) \end{bmatrix} \tag{3.8}$$

donde $\mathbf{Y}_{S1}$ representa la segmentación de la imagen, $\mathbf{Y}_{S2}$ es el esqueleto, $\mathbf{D}_{S1}$ es el *decoder* para la segmentación de la imagen y $\mathbf{D}_{S2}$ es el *decoder* para el esqueleto.

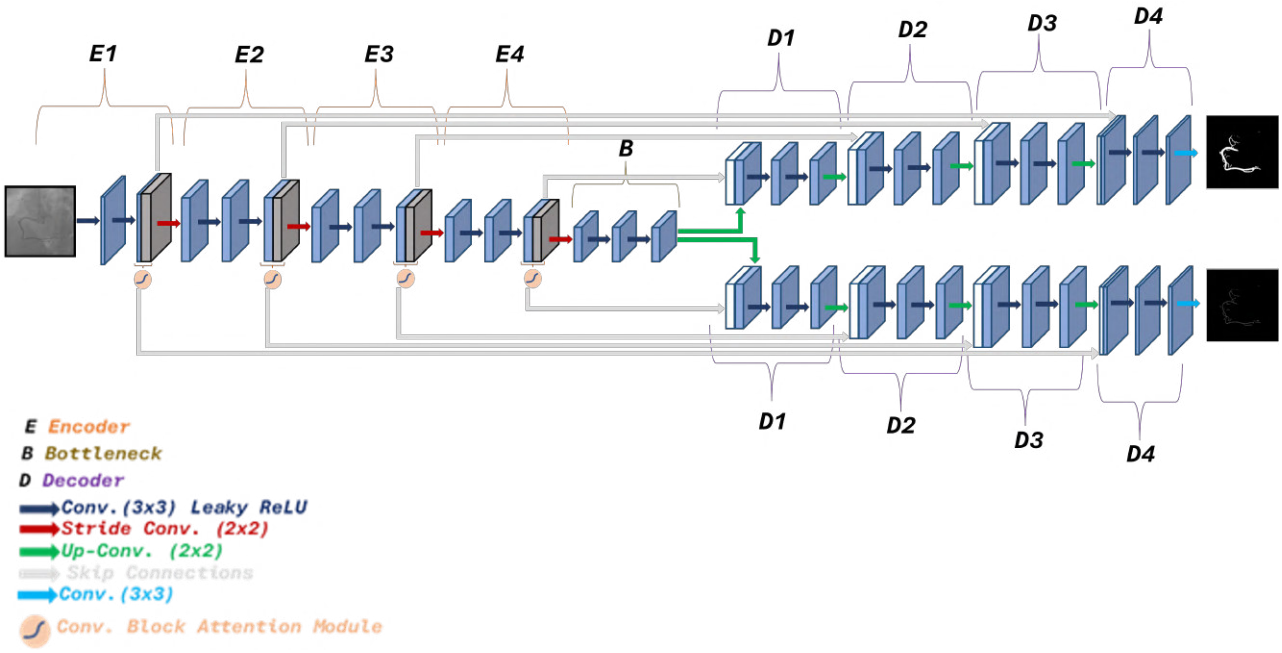


Figura 3.7. Multi-task Attention U-Net

3.3.1. Módulo de Normalización

Es muy común encontrar modificaciones a la U-Net que mejoran su desempeño. Tal es el caso de la inclusión de *Batch normalization* [Ioffe, 2015]. Esta función permite normalizar las salidas de una capa antes de pasarlas a la siguiente. Esto facilita que la siguiente capa reciba una distribución de salida más estable, lo que permite analizar los datos de manera más efectiva. La descripción matemática de *Batch normalization* (BN) normaliza la entrada $\mathbf{X}$ de la forma:

$$\mathrm{BN}(\mathbf{X}) = \gamma \frac{\mathbf{X} - \mathbb{E}_{B,H,W}[\mathbf{X}]}{\sqrt{\mathrm{Var}_{B,H,W}[\mathbf{X}] + \epsilon}} + \beta, \tag{3.9}$$

donde la entrada $\mathbf{X} \in \mathbb{R}^{B \times C \times H \times W}$ representa un *batch* de imágenes, B es el tamaño del *batch*, C es el número de canales, H es la altura y W es el ancho, $\mathbb{E}_{B,H,W}[\mathbf{X}]$ es la esperanza del *batch* y $\mathrm{Var}_{B,H,W}[\mathbf{X}]$ su varianza. Los parámetros $\gamma \in \mathbb{R}^C$ y $\beta \in \mathbb{R}^C$ permiten escalar y desplazar las activaciones normalizadas, mientras que ϵ es un pequeño valor constante que se introduce para garantizar la estabilidad numérica.

[Isensee y cols., 2021] proponen para mejorar el desempeño de la U-Net remplazando *Batch normalization* por *Instance normalization*, ya que ambas técnicas funcionan de manera similar, pero con la diferencia de que *Instance normalization* (IN) normaliza cada canal de cada imagen de entrenamiento de manera independiente, en lugar de normalizar por *batch*, como ocurre en Batch Normalization. Esto se puede observar en la Figura 3.8 donde se ilustra la diferencia entre *Batch normalization* contra *Instance normalization*. El uso de Instance Normalization nos permite trabajar con *batches* más pequeños, y así podemos optimizar la red haciendo que ocupe menos memoria. La ecuación que describe *Instance Normalization*, normaliza la entrada $\mathbf{X}$ mediante:

$$\mathrm{IN}(\mathbf{X}) = \gamma \frac{\mathbf{X} - \mathbb{E}_{H,W}[\mathbf{X}]}{\sqrt{\mathrm{Var}_{H,W}[\mathbf{X}] + \epsilon}} + \beta \tag{3.10}$$

donde también la entrada $\mathbf{X} \in \mathbb{R}^{B \times C \times H \times W}$ es un *batch* de representaciones de imágenes y B es el tamaño del *batch*, C es el número de canales, H es la altura y W es el ancho, las estimaciones de la media $\mathbb{E}_{H,W}[\mathbf{X}]$ y la varianza $\mathrm{Var}_{H,W}[\mathbf{X}]$ son por imagen, $\gamma \in \mathbb{R}^C$ and $\beta \in \mathbb{R}^C$ son parámetros que indican si se escala y desplaza el valor normalizado, ϵ un valor constante para estabilidad numérica.

3.3.2. Función de activación

Otro de los cambios que realizamos para la reducción de las imágenes y extracción de características fue el uso de *stride convolution* en lugar de *max pooling*. Esta técnica aplica convoluciones a pasos, lo que permite omitir ciertos pixeles mientras se realiza el mapeo de las

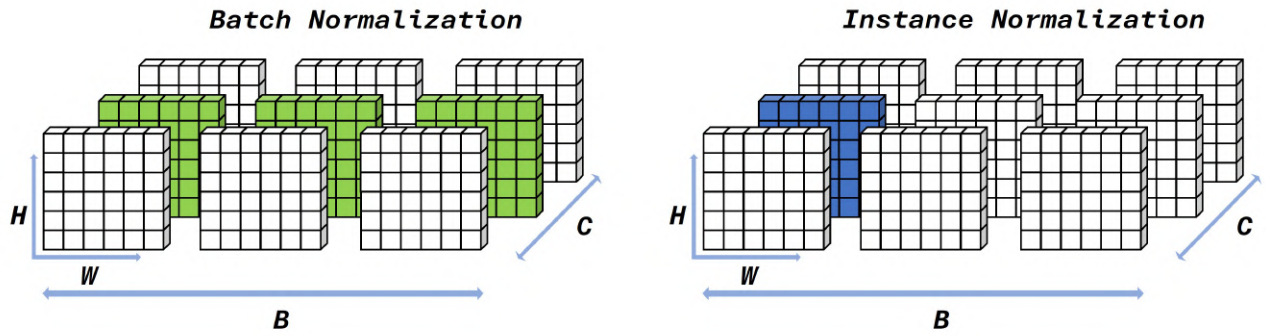


Figura 3.8. Los colores muestran la diferencia entre *Batch normalization* que normaliza a través de las características de todo el batch e *Instance normalization* normaliza por elemento del batch .

características, aplicando un filtro de convolución y reduciendo el tamaño de la imagen como se muestra en la Figura 3.9. La ventaja de esta técnica es que permite una extracción más controlada y detallada de las características, ya que la red aprende la operación de pooling de manera óptima, en lugar de seleccionar únicamente los valores máximos; lo que favorece una mejor generalización del modelo. La relación matemática que mencionan [Dumoulin y Visin, 2016] de *stride convolution* establece que la reducción de las dimensiones de la capa de salida están dadas por H_{out}, W_{out}:

$$H_{out} = \left\lfloor \frac{H_{in} + 2P - K}{S} \right\rfloor + 1, \tag{3.11}$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2P - K}{S} \right\rfloor + 1, \tag{3.12}$$

donde H_{in}, W_{in} son las dimensiones de la capa de entrada, K es el tamaño de kernel, S es el *stride* o paso que se desplazara el kernel y P es el *Padding* que es el número de píxeles que se agregan alrededor de la entrada.

De igual manera, la función de activación $ReLU$ puede ser reemplazada por *Leaky ReLU*. Algunas de las razones para realizar este cambio han sido presentadas por [Xu y cols., 2020], donde explican que *Leaky ReLU* puede ayudar a evitar el *Dying ReLU*, el cual consiste de la muerte de las neuronas, es decir, algunas neuronas dejan de activarse debido a que reciben valores de entrada negativos. *Leaky ReLU* provee una salida que es diferente de cero para

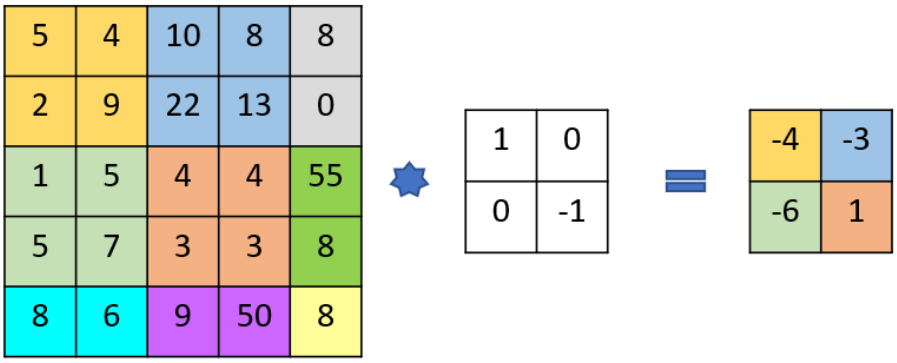


Figura 3.9. Representación de *stride convolution* de una imagen 5×5 con un kernel de 2×2 con un stride de 2.

valores negativos de entrada con el objetivo de evitar el descarte de información que puede ser relevante para la segmentación. *Leaky ReLU* se define de la forma:

$$f(x) = \begin{cases} x, & \text{si } x \geq 0, \\ \alpha \cdot x, & \text{si } x < 0, \end{cases} \tag{3.13}$$

donde x representa la entrada y si su valor es 0 o negativo la función $f(x)$ es $\alpha \cdot x$ y α controla la pendiente para los valores negativos de entrada. Por defecto, $\alpha = 1 \times 10^{-1}$. Gráficamente se puede visualizar en la Figura 3.10.

3.3.3. Módulo de atención

[Nguyen, 2021] describe el proceso de esqueletización como un proceso para representar un objeto mediante la extracción de los pixeles de una imagen binaria. Además señala que este proceso sigue siendo un tema novedoso en el ámbito del *deep learning*. Para abordar este problema, proponen un método para el desafío Pixel SkelNetOn de la tercera edición del taller *"Deep Learning for Geometric Computing"* en ICCV 2021. En su propuesta, modifica la arquitectura U-Net agregando módulos de atención para segmentar el esqueleto utilizando *Convolutional Block Attention Module* (CBAM).

El CBAM es un módulo diseñado para redes convolucionales, presentado por [Woo y cols., 2018] con la finalidad de mejorar la capacidad de la red para procesar las características

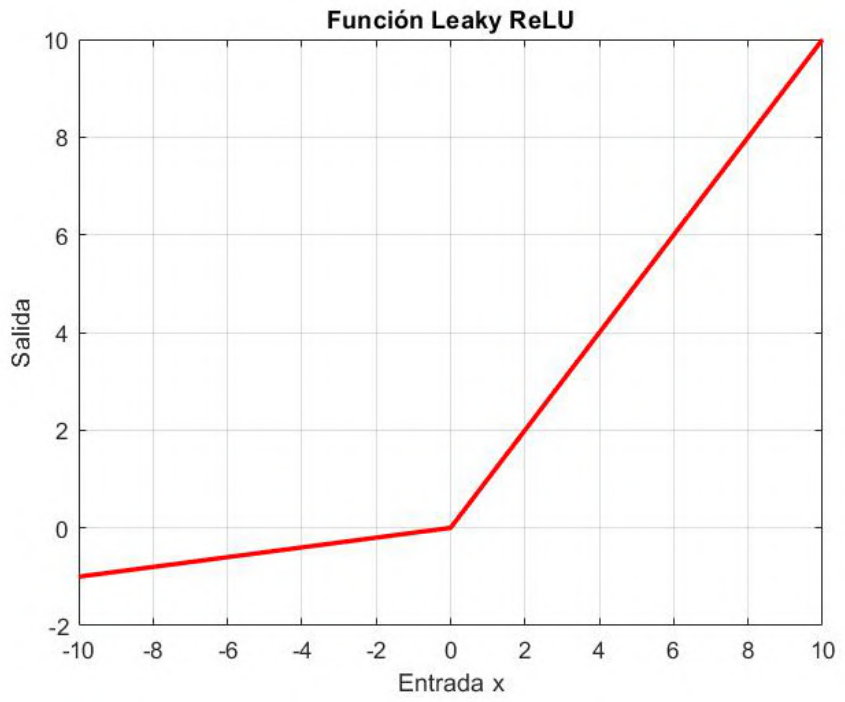


Figura 3.10. Función de activación Leaky ReLU con un $\alpha = 1 \times 10^{-1}$.

relevantes. Este modulo se compone principalmente de dos módulos secuenciales de dos tipos de atención: *Channel Attention Module* y *Spatial Attention Module*. Entonces dado como entrada un mapa de características intermedio $\mathbf{F} \in \mathbb{R}^{C \times H \times W}$, se obtiene un mapa secuencial de *Channel attention* $1D$ $\mathbf{M}_c \in \mathbb{R}^{C \times 1 \times 1}$ y un mapa de *Spatial attention* $2D$ $\mathbf{M}_s \in \mathbb{R}^{1 \times H \times W}$, los cuales refinan el mapa de características de entrada. Este proceso se ilustra en la Figura 3.11 y matemáticamente se expresa de la forma:

$$\mathbf{F}^{'} = \mathbf{M}_c(\mathbf{F}) \otimes \mathbf{F}, \tag{3.14}$$

y

$$\mathbf{F}^{''} = \mathbf{M}_s(\mathbf{F}^{'}) \otimes \mathbf{F}^{'} \tag{3.15}$$

donde $\otimes$ implica multiplicar elemento por elemento, en la multiplicación los valores de atención se copian acorde los valores de atención del canal y se transmiten a través de la dimensión espacial y viceversa. $\mathbf{F}^{''}$ es la salida final refinada.

El *Channel Attention Module* genera un mapa de *Channel Attention* explotando la relación entre canales de las características. Para ello, emplea *Average Pooling* y *Max Pooling* produciendo dos diferentes descriptores de contexto espacial distintos: $\mathbf{F}^{C}_{\mathbf{avg}}$ y $\mathbf{F}^{C}_{\mathbf{max}}$. Ambos

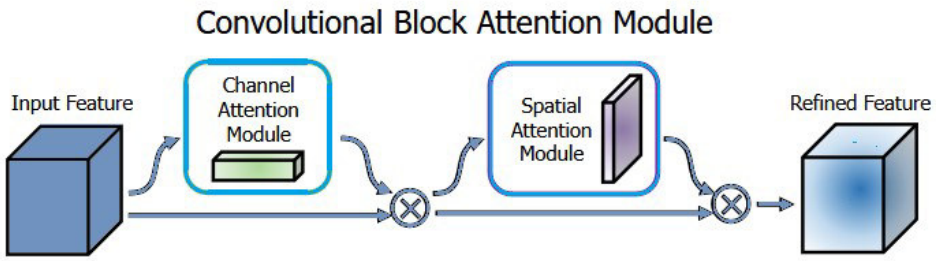


Figura 3.11. Estructura del modulo CBAM recuperado de [Woo y cols., 2018].

descriptores comparten la misma red para generar el mapa de *Channel Attention* $\mathbf{M}_C \in \mathbb{R}^{C\times 1\times 1}$. La red que comparten en común es un *Multi-Layer Perceptron* (*MLP*), con una capa oculta que reduce la sobrecarga de parámetros dada por un radio r de reducción de la forma:

$$\begin{aligned}\mathbf{M}_C(\mathbf{F}) &= \sigma\left(MLP(AvgPool(\mathbf{F}))\right) + MLP(MaxPool(\mathbf{F}))\\ &= \sigma\left(\mathbf{W}_1\left(\mathbf{W}_0\left(\mathbf{F}^C_{\mathbf{avg}}\right)\right) + \mathbf{W}_1\left(\mathbf{W}_0\left(\mathbf{F}^C_{\mathbf{max}}\right)\right)\right),\end{aligned} \tag{3.16}$$

donde σ indica que es la función *sigmoid*, $\mathbf{W}_0 \in \mathbb{R}^{\frac{C}{r}\times C}$ y $\mathbf{W}_1 \in \mathbb{R}^{C\times\frac{C}{r}}$, los pesos del *MLP* $\mathbf{W}_0$ y $\mathbf{W}_1$ se comparten para ambas entradas y la función de activación Leaky ReLU le sigue $\mathbf{W}_0$, como se ve en la Figura 3.12.

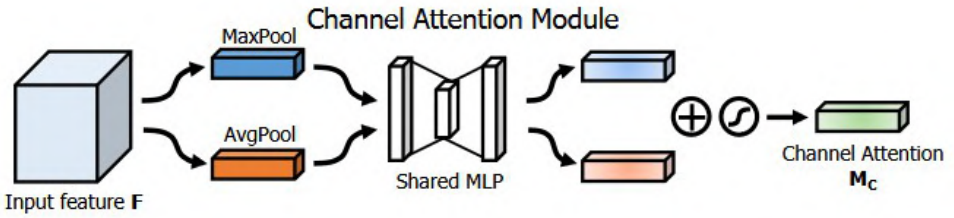


Figura 3.12. Estructura de *Channel Attention Module* recuperado de [Woo y cols., 2018].

Por otro lado, el *Spatial Attention Module* se enfoca en generar un mapa de *Spatial Attention* mediante operaciones de *Average Pooling* y *Max Pooling* a lo largo del eje del canal. Posteriormente estos se concatenan para obtener un descriptor de características que permite resaltar regiones de mayor interés espacial. Finalmente, se aplica una capa convolucional sobre el descriptor de características concatenado para generar un mapa de *Spatial Attention* $\mathbf{M}_S \in \mathbb{R}^{H\times W}$.

Este módulo agrega información del canal del mapa de características al usar dos

operaciones de *Pooling* dando dos mapas $2D$ uno de $\mathbf{F}^S_{\mathbf{avg}} \in \mathbb{R}^{1\times H\times W}$ y otro $\mathbf{F}^S_{\mathbf{max}} \in \mathbb{R}^{1\times H\times W}$ que resaltan características de *Average Pooling* y características de *Max Pooling* a lo largo del canal. Posteriormente, estos mapas se concatenan y convolucionan por medio de una capa de convolución estándar, obteniendo así el mapa de *Spatial Attention* $2D$, matematicamente se expresa de la forma:

$$\begin{aligned}\mathbf{M}_S(\mathbf{F}) &= \sigma\left(\mathbf{f}^{7\times 7}\left([AvgPool(\mathbf{F}); MaxPool(\mathbf{F})]\right)\right)\\ &= \sigma\left(\mathbf{f}^{7\times 7}\left([\mathbf{F}^S_{\mathbf{avg}}; \mathbf{F}^S_{\mathbf{max}}]\right)\right),\end{aligned} \tag{3.17}$$

donde σ indica que es la función *sigmoid* y $\mathbf{f}^{7\times 7}$ indica una convolución con un filtro de tamaño 7×7 como se ve en la Figura 3.13.

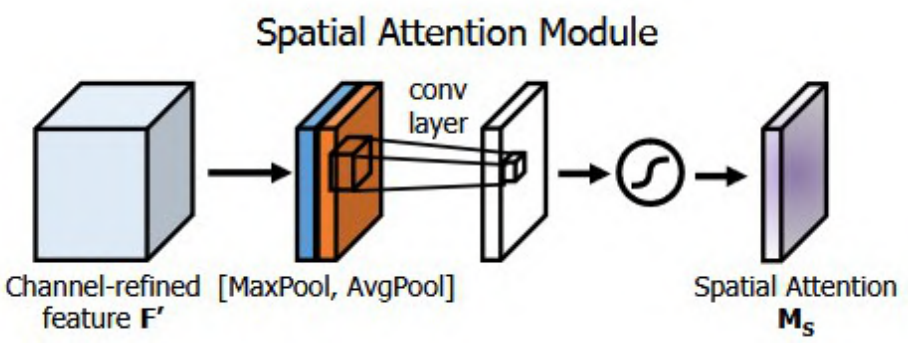


Figura 3.13. Estructura de *Spatial Attention Module* recuperado de [Woo y cols., 2018].

3.3.4. Función de perdida

En el caso de la segmentación de imágenes, de acuerdo a lo planteado por [Azad y cols., 2023] el método que se tiene para evaluar los píxeles de una imagen en una clase en particular es fundamental para definir el rendimiento del modelo. Las funciones de perdida son esenciales en los algoritmos de segmentación ya que tiene como objetivo mejorar el desempeño de la red.

Para el entrenamiento de la U-Net la función de pérdida más utilizada es la *Binary Cross-Entropy* (BCE). Esta función se emplea en tareas de clasificación binaria y compara la diferencia entre las probabilidades predichas y las etiquetas o *ground truth* binarias reales. Calcula el error que existe entre la probabilidad predicha de que una instancia sea de la clase positiva y la etiqueta binaria real, si es positiva es 1 y si es negativa 0. Mediante la comparación

del logaritmo de la probabilidad predicha de la clase positiva y el logaritmo de la probabilidad complementaria de la clase negativa, con la intención de penalizar a los modelos cuando las predicciones se alejan de su *ground truth*. Matemáticamente se expresa de la forma:

$$\text{BCE}(p, y) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)\right], \tag{3.18}$$

donde N son las muestras del conjunto de datos, y_i es su *ground truth* para la muestra i, con $y_i \in \{0, 1\}$, p_i es la predicción para la clase positiva, con $0 \leq p_i \leq 1$.

[Gong y cols., 2019] discuten que en los modelos de aprendizaje multitarea, la combinación adecuada de las funciones de pérdidas es un problema crucial en el aprendizaje multitarea ya que se tiene que buscar una estrategia para que la función de perdida global se adapte y mejore el rendimiento del modelo para las múltiples tareas.

En general, el desempeño de los modelos de *deep learning* depende de los ajustes de los hiperparámetros, lo que impacta el rendimiento y precisión del modelo. [Montazerolghaem y cols., 2023] realizaron un estudio en el que utilizaron diferentes funciones de pérdida en la segmentación de imágenes de próstata utilizando una red U-Net, comparando su rendimiento al utilizar diversas funciones de pérdida, entre las que se encuentran BCE, IoU, Dice y combinaciones de estas entre otras funciones. Concluyeron que el rendimiento en la segmentación de las imágenes de próstata al utilizar funciones de pérdida combinadas era superior al de utilizar funciones de pérdida individuales.

Teniendo esto en cuenta, se elige usar una función de pérdida combinada entre las métricas *Intersection over Union* (IoU) y *Dice score* (DSC). Para ello, se definieron IoUloss y Diceloss para las tareas de segmentación y esqueletización con la intención de mejorar el rendimiento del modelo.

De acuerdo a [Rahman y Wang, 2016] la métrica IoU evalua que tan bueno es el comportamiento del modelo para el problema de segmentación de objetos. IoU proporciona una medida de similitud entre la predicación y el *ground truth*, definida como:

$$\text{IoU}(p, y) = \frac{|p_i \cap y_i|}{|p_i \cup y_i|}, \tag{3.19}$$

donde $p_i \cap y_i$ representa la intersección de la predicción con su *ground truth*, $p_i \cup y_i$ indica su unión.

Por otro lado, se define IoUloss como:

$$\text{IoUloss} = 1 - \text{IoU}(p, y). \tag{3.20}$$

Asimismo, [Sudre y cols., 2017] propone la métrica *Dice Score*, que mide la superposición del conjunto de píxeles de la predicción con el conjunto de píxeles del *ground truth*. Esto permite evaluar que también el modelo genera la segmentación. Matemáticamente, se expresa de la forma:

$$\text{DSC}(p, y) = \frac{2|p_i \cap y_i|}{|p_i| + |y_i|}, \tag{3.21}$$

donde 2 es un factor de ponderación que indica el doble de la intersección de $p_i \cap y_i$ entre la suma de los conjuntos de pixeles de la predicción y *ground truth* para darle peso a la superposición. Entonces el Diceloss se define como:

$$\text{Diceloss} = 1 - \text{DSC}(p, y). \tag{3.22}$$

Aprovechando las ventajas de IoU Loss y Dice Loss, se definió una combinación lineal ponderada de ambas, tal que:

$$\text{IouDiceLoss} = \alpha \cdot \text{Diceloss} + (1 - \alpha) \cdot \text{IoUloss}, \quad \alpha \in (0, 1), \tag{3.23}$$

donde α es un parámetro para darle más peso a una perdida que a otra, entre más grande sea el valor de α, más peso tendrá Diceloss y entre más pequeño sea, más peso tendrá IoUloss.

3.4. Procesamiento del esqueleto

Una vez generada la predicción del esqueleto, es importante recordar que esta es una línea a nivel píxel de 1 píxel de ancho, por lo que pueden generarse pequeños espacios que corten la continuidad de la linea. Para solucionar este problema, procesamos las imágenes de esqueletos mediante operaciones morfológicas, basándonos en el trabajo de [Heijmans y Ronse, 1990], quienes emplean dilatación y erosión como se puede ver en la Figura 3.14.

La dilatación representa la suma de Minkowski, que consiste en expandir un conjunto A mediante la aplicación de un elemento estructurante B de la siguiente manera:

$$A \oplus B = \{a + b \mid a \in A, b \in B\}, \tag{3.24}$$

donde a son los elementos que pertenecen al conjunto A y b son los elementos que pertenecen al elemento estructurante B.

Por otro lado la erosión representa la reducción del conjunto A mediante la aplicación de un elemento estructurante B definida como:

$$A \ominus B = \{z \mid B_z \subseteq A\}, \tag{3.25}$$

donde B_z es la traslación del elemento estructurante B ubicada en el punto z de un conjunto A, pero B_z debe estar contenido dentro de A de lo contrario z no se toma en cuenta.

(a) Predicción (b) Predicción procesada

Figura 3.14. (a) Predicción antes de aplicar los procesos morfológicos, (b) Predicción después de aplicar dilatación y erosión.

3.5. Extracción de puntos de clave

Una vez que se ha obtenido la segmentación y esqueletización mejoradas por elementos estructurantes de las imágenes, se procede a extraer los puntos clave a partir del esqueleto de las imágenes. Para ello realizamos una adaptación del método Shi-Tomasi [Shi y cols., 1994] para la detección de esquinas. El método consiste en calcular la matriz de autocorrelación para evaluar los autovalores λ_1 y λ_2 para obtener la respuesta de esquina como el mínimo de

los autovalores por medio de los gradientes de intensidad de los píxeles utilizando el filtro de Sobel, con el propósito de analizar las regiones donde existan cambios máximos de intensidad en diferentes direcciones. Para ello, se realiza la convolución entre la imagen y 21 kernels de rotación.

Los kernels de rotación son de la forma,

$$\begin{aligned}
\text{kernel } 1 &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad \text{kernel } 2 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{kernel } 3 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
\text{kernel } 4 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \text{kernel } 5 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad \text{kernel } 6 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \\
\text{kernel } 7 &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{kernel } 8 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{kernel } 9 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\
\text{kernel } 10 &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{kernel } 11 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{kernel } 12 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\
\text{kernel } 13 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{kernel } 14 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{kernel } 15 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \\
\text{kernel } 16 &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{kernel } 17 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{kernel } 18 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
\text{kernel } 19 &= \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{kernel } 20 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{kernel } 21 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}
\end{aligned} \tag{3.26}$$

Las imágenes de entrada son normalizadas en el rango de 0 a 255 para tener uniformidad

en los cálculos. Posteriormente, se aplican los kernels de rotación para resaltar la intensidad de los píxeles y así ejecutar el método de Shi-Tomasi, lo que permite detectar los puntos iniciales, finales y bifurcaciones del esqueleto.

Para ello, se calculan los gradientes por medio del filtro de Sobel, el cual calcula los gradientes de las imágenes tanto en la dirección vertical como horizontal, mediante la aplicación de kernels de convolución 3×3. Matemáticamente se expresa de la manera:

$$K_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \tag{3.27}$$

$$K_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{3.28}$$

donde K_x y K_y son los kernels para el operador Sobel, incluyendo un filtro Gaussiano para reducir el ruido.

Los kernels de convolución se aplican a una imagen $I(x, y)$ de la siguiente manera:

$$I_x(x, y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I(x + i, y + j) \cdot K_x(i, j) \tag{3.29}$$

$$I_y(x, y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I(x + i, y + j) \cdot K_y(i, j) \tag{3.30}$$

donde I_x y I_y son los gradientes en las direcciones horizontal y vertical, respectivamente e $I(x, y)$ representa la intensidad de los píxeles.

Posteriormente, se calcula la matriz de segundo momento M:

$$M = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{3.31}$$

donde I_x^2 y I_y^2 representan cada uno el producto de sus gradientes al cuadrado, $I_x I_y$ es el producto de ambos gradientes.

Para encontrar los autovalores λ_1 y λ_2 a partir de la matriz M se calculan de la ecuación cuadrática,

$$\lambda^2 - (\text{trace}(M))\lambda + \det(M) = 0 \tag{3.32}$$

donde trace(M) representa $I_x^2 + I_y^2$, que es la suma del producto de los gradientes al cuadrado y det(M) representa $I_x^2 \cdot I_y^2$ - $(I_x I_y)^2$, siendo el producto de los productos de los gradientes al cuadrado menos el producto de ambos gradientes al cuadrado.

Resolviendo la ecuación cuadrática se obtienen los autovalores,

$$\lambda_{1,2} = \frac{\text{trace}(M) \pm \sqrt{\text{trace}(M)^2 - 4 \cdot \det(M)}}{2} \tag{3.33}$$

para encontrar las respuestas de esquinas $R(y, x)$ mediante el valor mínimo de la forma,

$$R(y, x) = \text{mín}(\lambda_1, \lambda_2) \tag{3.34}$$

se procede a normalizar los valores de $R(y, x)$ en 0 y 1 y aplicar un umbral para considerar los valores altos de respuesta de los píxeles. Seguido de una supresión no máxima, que deja los valores de $R(y, x)$ que son máximos en una vecindad local para detectar bien las esquinas.

La selección de los puntos clave se basa en los puntos $R(y, x)$ de mayor valor, asegurando una distancia mínima para evitar superposiciones.

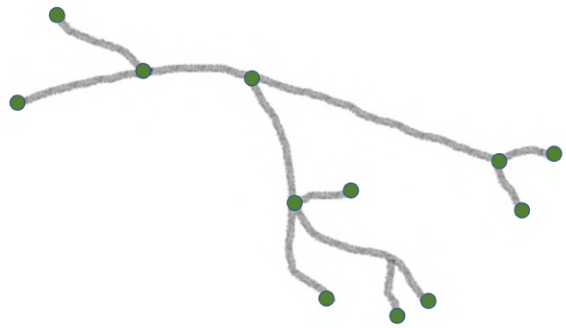
Una vez determinados los puntos clave, se procede a determinar su distribución. Para ello, se calcula el centroide de la imagen a partir de la media aritmética de los puntos en direcciones horizontal y la media de los puntos en vertical,

$$\text{centroide} = (\mu_x, \mu_y), \tag{3.35}$$

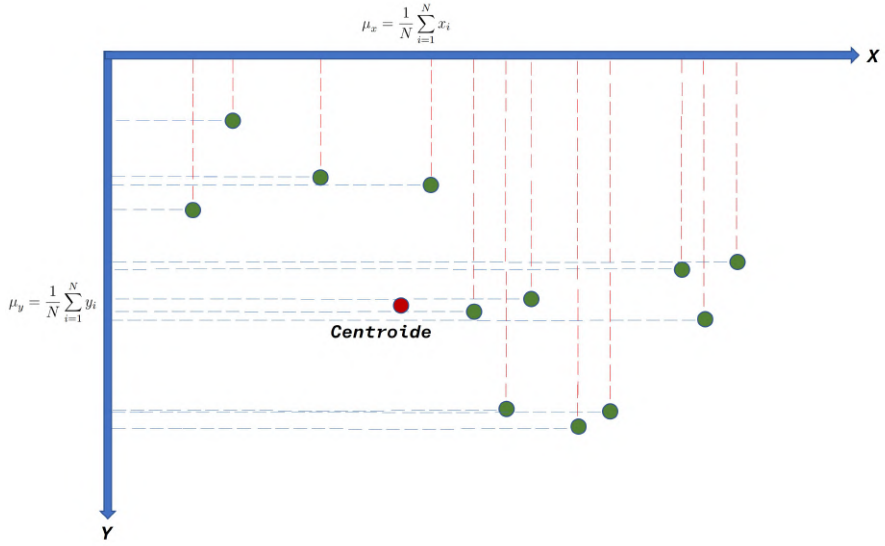
$$\mu_x = \frac{1}{N} \sum_{i=1}^{N} x_i, \tag{3.36}$$

$$\mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i, \tag{3.37}$$

cómo se puede ver en la Figura 3.15,



(a) Puntos clave



(b) Centroide

Figura 3.15. (a) Se extraen los puntos clave, (b) Se obtiene el centroide.

Para determinar la orientación de la imagen, es decir, si corresponde aun lado izquierda o de derecho, se realiza un análisis por región. Para ello se calcula la dispersión de los puntos clave por medio de la media, varianza, desviación estándar y densidad de la siguiente manera:

$$\text{varianza}_x = \frac{1}{n}\sum_{i=1}^{n}(x_i - \text{mean}_x)^2, \tag{3.38}$$

$$\text{varianza}_y = \frac{1}{n}\sum_{i=1}^{n}(y_i - \text{mean}_y)^2, \tag{3.39}$$

la desviación estándar σ

$$\sigma_x = \sqrt{\text{varianza}_x}, \tag{3.40}$$

$$\sigma_y = \sqrt{\text{varianza}_y}. \tag{3.41}$$

Para determinar qué región predomina y así orientar la imagen, utilizamos la densidad de puntos, la cual indica la proporción de puntos clave que se encuentran a un lado del centroide con respecto al otro, Esto se define mediante:

$$\text{densidad}_{\mathbf{izq}} = \frac{N_{\mathbf{izq}}}{N}, \tag{3.42}$$

$$\text{densidad}_{\mathbf{der}} = \frac{N_{\mathbf{der}}}{N}, \tag{3.43}$$

$$\text{densidad}_{\mathbf{izq}} < \text{densidad}_{\mathbf{der}}, \tag{3.44}$$

donde $\text{densidad}_{\mathbf{izq}} < \text{densidad}_{\mathbf{der}}$ nos indica que al haber una menor distribución de puntos del lado izquierdo del centroide determinamos que el sentido de la imagen es de izquierda a derecha, en caso contrario la imagen parte de derecha a izquierda como se puede ver en la Figura 3.16.

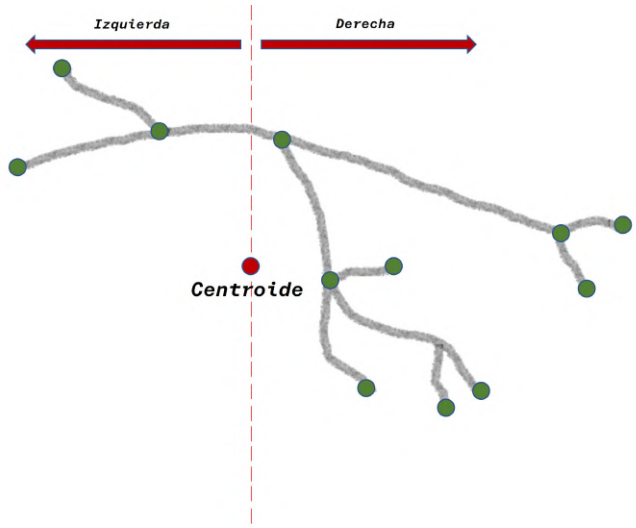


Figura 3.16. Se determina el sentido de la imagen por la cantidad de puntos dispersos a la izquierda o derecha del centroide.

3.6. Transformaciones locales

Las técnicas de *data augmentation* tradicionales implican rotaciones, espejeado y traslaciones sobre toda la imagen. En el caso de la rotación y espejeado, los parámetros utilizados son un ángulo en el rango $[0, 2\pi]$ y un valor booleano, respectivamente. De manera similar, existen técnicas que producen variaciones a nivel local, pero influenciadas por el tamaño de la imagen, como el caso de la transformación elástica [Simard y cols., 2003].

Esta transformación consiste en generar deformaciones en la imagen de manera controlada mediante alteraciones en los valores de las coordenadas, como se puede ver en la Figura 3.17. Matemáticamente se puede expresar como:

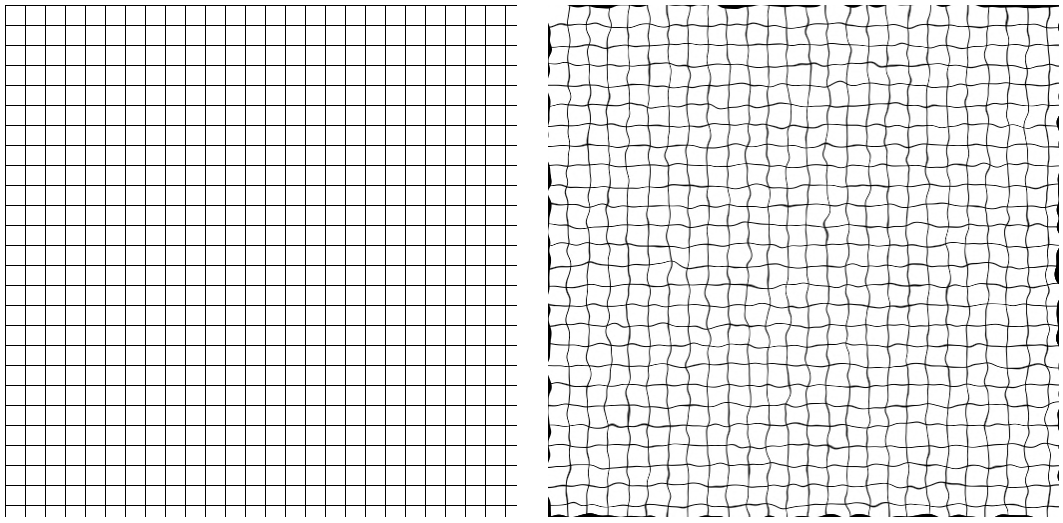
$$T(x, y) = \Big(x + \alpha \cdot G(\Delta_x),\ y + \alpha \cdot G(\Delta_y)\Big) \tag{3.45}$$

donde x y y son las coordenadas de los píxeles, mientras que Δ_x y Δ_y son alteraciones generadas de manera aleatoria mediante ruido gaussiano, que a su vez es filtrado por una convolución gaussiana regulada por un parámetro α. El parámetro α controla la intensidad que tendrá la deformación, si α es un valor alto, la deformación será mayor y para un valor bajo de α, la deformación será menor.

Para el filtrado gaussiano de las alteraciones Δ_x y Δ_y se emplea la siguiente forma:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{3.46}$$

donde σ es la desviación estándar que modifica el filtrado.



(a) Campo original (b) Campo deformado

Figura 3.17. (a) La imagen original, (b) La imagen después de aplicar la deformación.

Por ejemplo, en imágenes de la base de datos MNIST, de tamaño 28×28 píxeles, esta técnica permite generar imágenes sintéticas con similitudes a la imagen original pero con variaciones controladas, como se puede ver en la Figura 3.18.

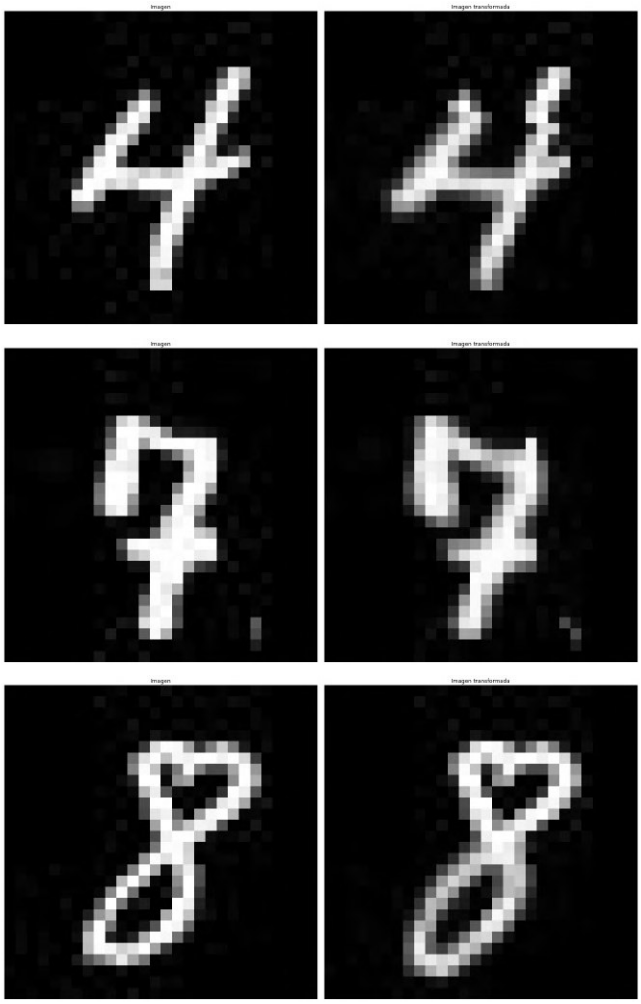


Figura 3.18. Transformación elástica en imágenes MNIST.

La deformación elástica ha sido aplicada de manera supervisada (indicando manualmente la región donde aplicarla) en regiones 76×76 en imágenes de mamografía con la finalidad de alterar únicamente regiones de interés [Castro y cols., 2018].

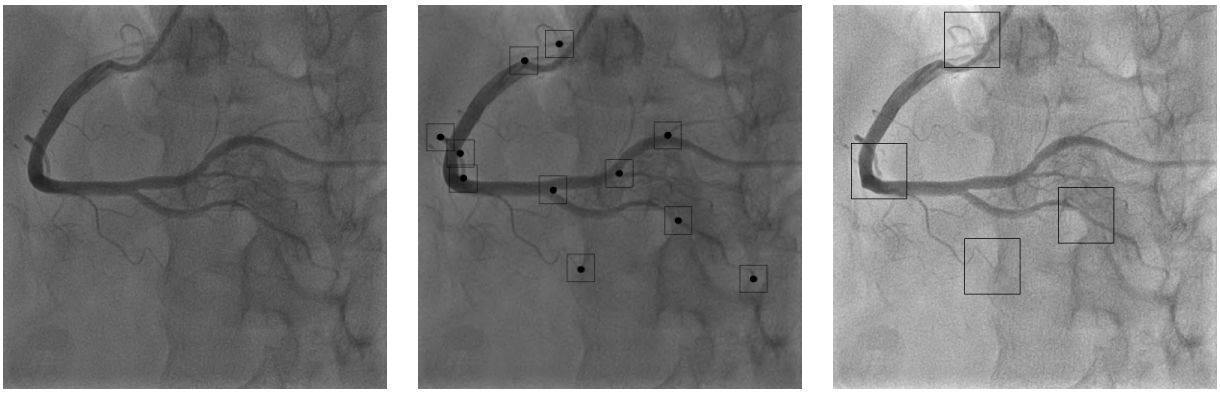
Inspirados en esté trabajo, proponemos una deformación elástica automática para imágenes de angiografía coronaria, las cuales han pasado por la *Multi-task Attention U-Net* (previamente entrenada) para producir su segmentación y su esqueleto. Posteriormente, las imágenes de esqueletos se procesan para eliminar pequeñas regiones aisladas y mejorar su estructura de manera que se puedan extraer puntos clave de ellos.

Una vez obtenidos los puntos clave, se toman sus coordenadas sobre la imagen original para aplicar transformaciones de manera local alrededor de ellos. Para estas transformaciones locales se emplea un kernel que permite realizar las deformaciones en una región controlada al

rededor del punto clave.

Si consideramos un punto clave como centro de un kernel de tamaño $(k \times k)$ como se ve en la Figura 3.19, el valor de k tiene que ser impar. El procedimiento consiste en:

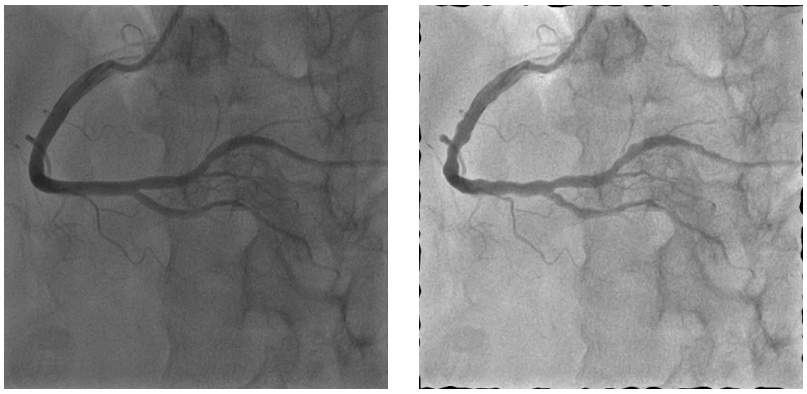
1. Recortar de la imagen original una región del tamaño del kernel.
2. Aplicar las transformaciones a dicha región.
3. Sustituir la región original en la imagen por la versión transformada.



(a) Imagen original (b) Kernels (c) Imagen Transformada

Figura 3.19. Una vez que a la imagen original (a) se le extraen los puntos clave para crear los kernels (b) con centro en esos puntos, se procede a hacer transformaciones locales en algunos de esos puntos (c).

Esto permite controlar la cantidad de puntos utilizados, seleccionándolos de manera aleatoria para generar variabilidad en las imágenes. El tamaño de kernel controla la deformación de una región sin afectar el resto de la imagen de lo contrario la imagen se vería como en la Figura 3.20.



(a) Imagen original (b) Imagen Transformada

Figura 3.20. Imagen original (a), Imagen sin controlar que regiones transformar (b).

El parámetro α controla el grado de deformación en la región modificada. Al final, se obtiene una imagen sintética con similitudes a la imagen original, pero con variaciones controladas.

CAPÍTULO 4

Resultados experimentales

Dentro de este capítulo, presentamos los resultados obtenidos de las diferentes versiones de U-Net que utilizamos la Vanilla U-Net-64, Vanilla U-Net-32 y Vanilla U-Net-16, para la generación de imágenes sintéticas. Para el entrenamiento se utilizó un *batch size* de 4 en 150 épocas, con un *learning rate* de 0,01. El optimizador eempleado fue AdamW, con un *weight decay* de 0,01.

Comenzamos determinando si la reducción de la cantidad de filtros que usa la U-Net original podía mantener el desempeño del modelo, pero con un menor costo computacional en la segmentación de imágenes. Para ello, la Tabla 4.1 presenta las métricas comparativas para cada configuración de U-Net, en la cual probamos 64 filtros, 32 filtros y 16 filtros en cada bloque del *encoder*, reduciendo así el número de parámetros, de 31M, a 7.7M y 1.9M, respectivamente.

Los resultados indican que Vanilla U-Net-16 logra el mejor desempeño en la mayoría de las métricas, alcanzando el menor Loss (0.03855), el mayor IoU (0.77315), Dice (0.87010), Accuracy (0.98554), Recall (0.86267) y F1 Score (0.87470). Además, obtiene la menor Distancia de Hausdorff (69.25949), lo que sugiere una segmentación más precisa en términos de la forma de los vasos sanguíneos. Por otro lado, Vanilla U-Net-32 presenta la mayor

Tabla 4.1. Resultados de segmentación con Vanilla U-Net variando el número de filtros.

Modelo	**Loss**	**IoU**	**Dice**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Hausdorff Distance**
Vanilla U-Net-64	0.04049	0.76198	0.86277	0.98489	0.89085	0.84741	0.86773	80.31830
Vanilla U-Net-32	0.04077	0.76645	0.86572	0.98526	**0.89698**	0.84778	0.87082	80.79632
Vanilla U-Net-16	**0.03855**	**0.77315**	**0.87010**	**0.98554**	0.88879	**0.86267**	**0.87470**	**69.25949**

precisión (0.89698), aunque con un ligero descenso en Recall en comparación con U-Net-16, lo que indica una segmentación más conservadora. Mientras tanto, Vanilla U-Net-64, aunque competitivo, muestra valores inferiores en las métricas de segmentación en comparación con las configuraciones más compactas. Estos resultados sugieren que reducir el número de filtros en la arquitectura Vanilla U-Net puede mejorar la segmentación sin comprometer el rendimiento, optimizando al mismo tiempo la eficiencia del modelo.

Visualmente comparamos los resultados de segmentación que se obtuvo para cada modelo mediante las imágenes reales, las máscaras y las predicciones obtenidas en los modelos para imágenes RCA y LCA como se puede apreciar en la Figura 4.1 y Figura 4.2.

Los resultados de los diferentes modelos de Vanilla U-Net en imágenes RCA no tienen una segmentación limpia en comparación con las máscaras. Se pueden observar secciones de arterias en tonalidades grises donde a la red tuvo dificultades para aprender. También hay variaciones entre la máscara y los resultados de los modelos. Algunas ramificaciones no terminan de la misma forma que la imagen original. A diferencia de la máscara, los resultados de los modelos intentan segmentar secciones de la arteria que están en la imagen original, pero no en la máscara.

En el caso de los modelos de Vanilla U-Net en imágenes LCA ocurre algo similar, se observan secciones con tonalidades grises y bosquejos de ramificaciones que no aparecen en la máscara, pero si en la imagen original. Al analizar los resultado de cada modelo, se nota que existen pequeñas discrepancias entre un modelo y el otro en regiones que no aparecen en el *ground truth*, lo que nos da la pauta para elegir el modelo de Vanilla U-Net-16 como modelo base para las siguientes etapas de este trabajo. Este modelo tiene un costo computacional menor para el entrenamiento de la red.

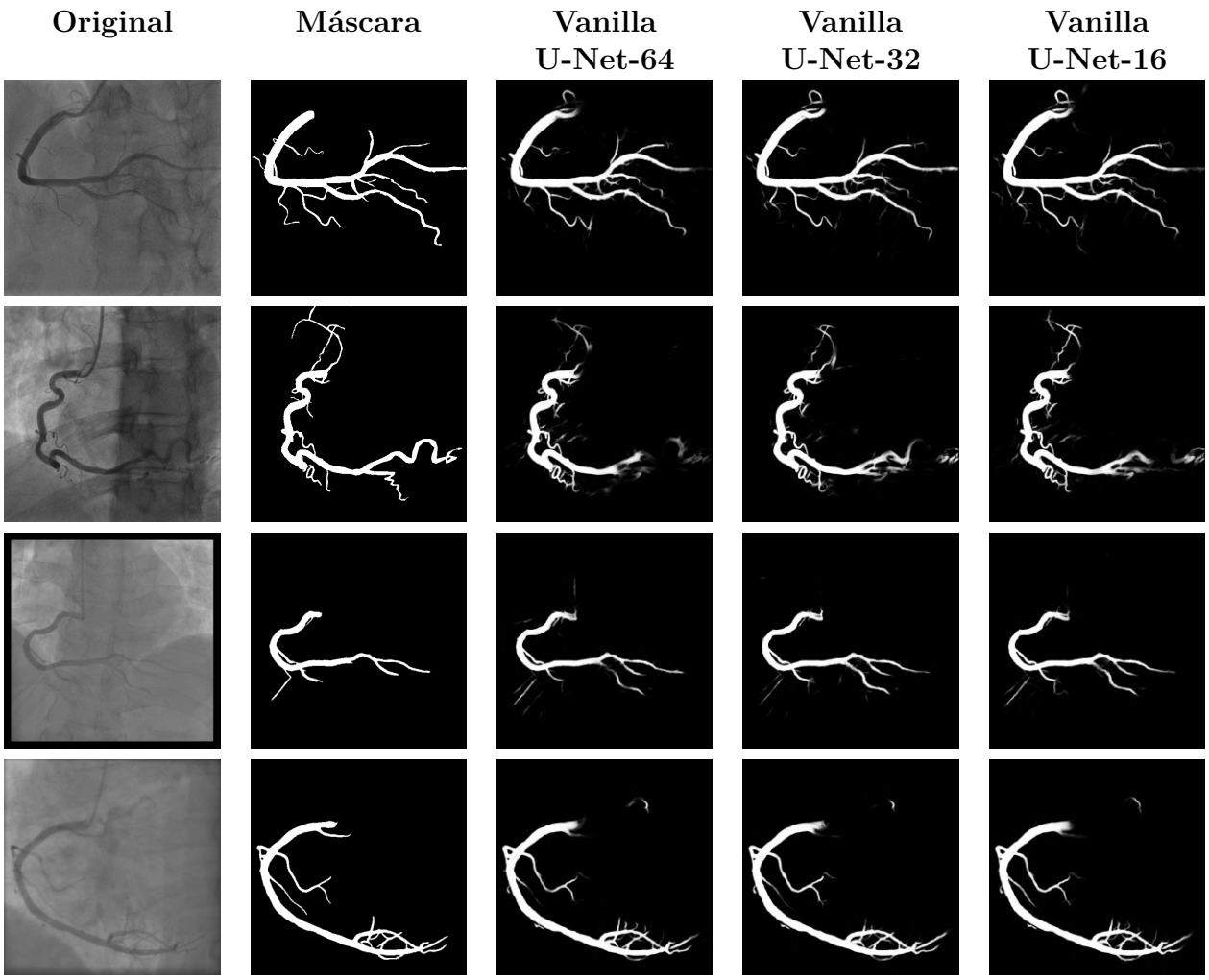


Figura 4.1. Matriz de imágenes RCA : Original, Máscara y Resultado de los modelos Vanilla U-Net-64,Vanilla U-Net-32 y Vanilla U-Net-16.

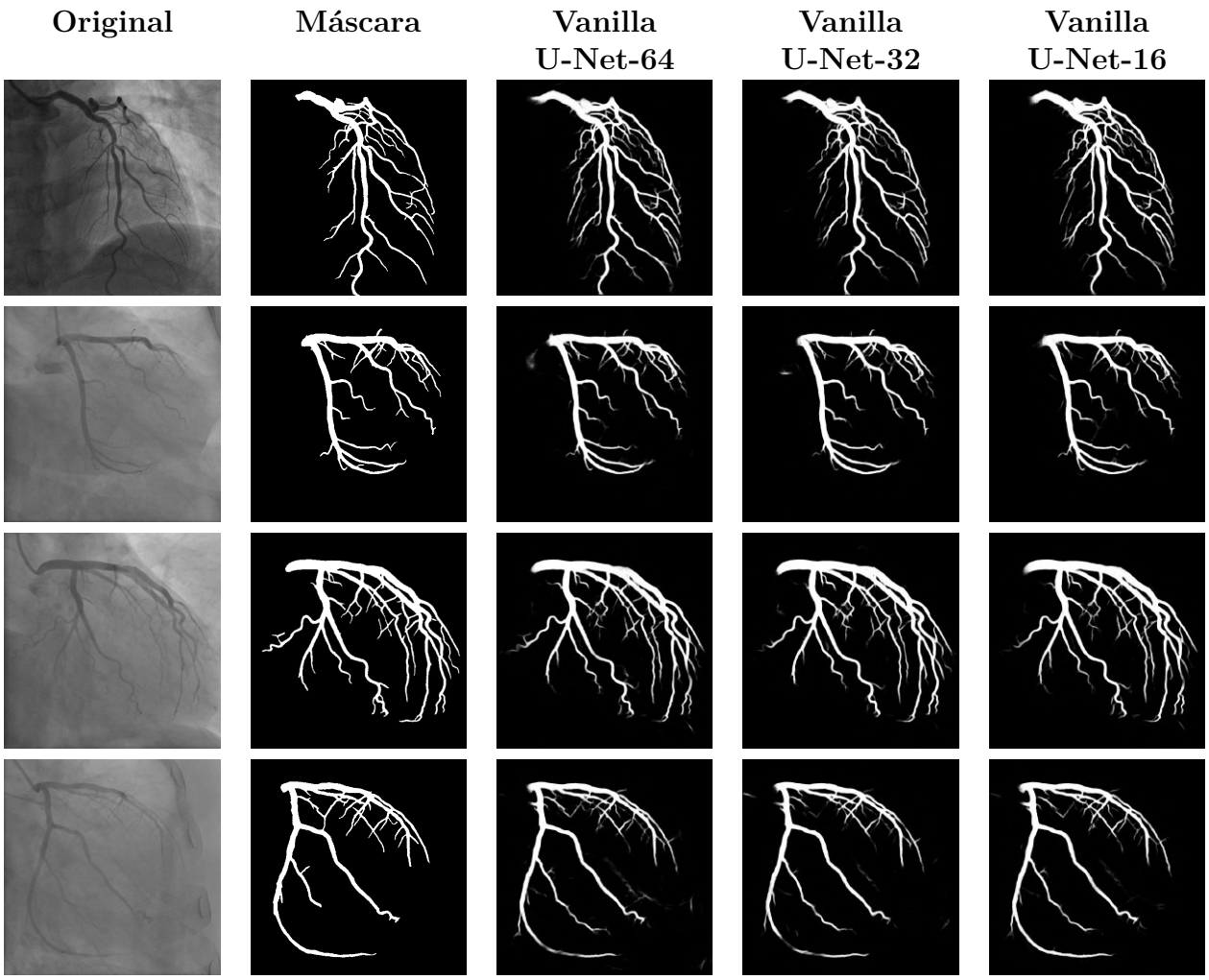


Figura 4.2. Matriz de imágenes LCA : Original, Máscara y Resultado de los modelos Vanilla U-Net-64,Vanilla U-Net-32 y Vanilla U-Net-16.

Ahora bien, aplicamos las modificaciones planteadas en la metodología para nuestro mejor modelo, Vanilla U-Net-16, con el objetivo de mejorar su desempeño y tener una mayor generalización. Los resultados que se muestran a continuación fueron obtenidos tras la aplicación de estas modificaciones, convirtiendo el modelo en Multi-task, cambiar *Normalization Layer* (NL), BN por IN y utilizando diferentes funciones de pérdida BCE por IouDiceLoss (IDLoss) pero probando diferentes combinaciones como se muestra en la Tabla 4.2 y la Tabla 4.3.

Tabla 4.2. Segmentación con Multi-task U-Net con modificaciones.

Modelo	NL	Loss Function	Loss	IoU	Dice	Accuracy	Precision	Recall	F1 Score	Hausdorff Distance
Multitask U-Net	BN	BCE	**0.03855**	**0.77315**	**0.87010**	**0.98554**	0.88879	**0.86267**	**0.87470**	69.25949
Multitask U-Net	IN	BCE	0.04070	0.76948	0.86733	0.98533	**0.89860**	0.84704	0.87130	**67.5348**
Multitask U-Net	BN	IDLoss	0.22072	0.74713	0.85241	0.98365	0.87483	0.84471	0.85828	79.03499
Multitask U-Net	IN	IDLoss	0.21605	0.75662	0.85841	0.98413	0.87251	0.85684	0.86358	74.29862

Tabla 4.3. Esqueleto con Multi-task U-Net con modificaciones.

Modelo	NL	Loss Function	Loss	IoU	Dice	Accuracy	Precision	Recall	F1 Score	Hausdorff Distance
Multitask U-Net	BN	BCE	**0.01663**	0.36561	0.52950	0.99393	0.67607	0.45044	0.53955	70.04714
Multitask U-Net	IN	BCE	0.01670	0.36939	0.53405	**0.99396**	**0.68006**	0.45283	0.54263	**68.59376**
Multitask U-Net	BN	IDLoss	0.12469	0.39259	0.55922	0.99299	0.55569	0.58512	0.56904	83.34329
Multitask U-Net	IN	IDLoss	0.09529	**0.40152**	**0.56793**	0.99322	0.57104	**0.58577**	**0.57754**	98.57278

Como se puede ver en la Tabla 4.2 y la Tabla 4.3, los primeros modelos que implementan BN con BCE e IN con BCE presentan mejores métricas en comparación con los modelos que implementan BC con IDLoss e IN con IDLoss. Esto se debe a que los ajustes están diseñados en mejorar la predicción de las imágenes de segmentación, como se observa en la Figura 4.3 y de las imágenes del esqueleto, como se muestra en la Figura 4.4 para imágenes RCA y en la Figura 4.5 y Figura 4.6 para imágenes LCA.

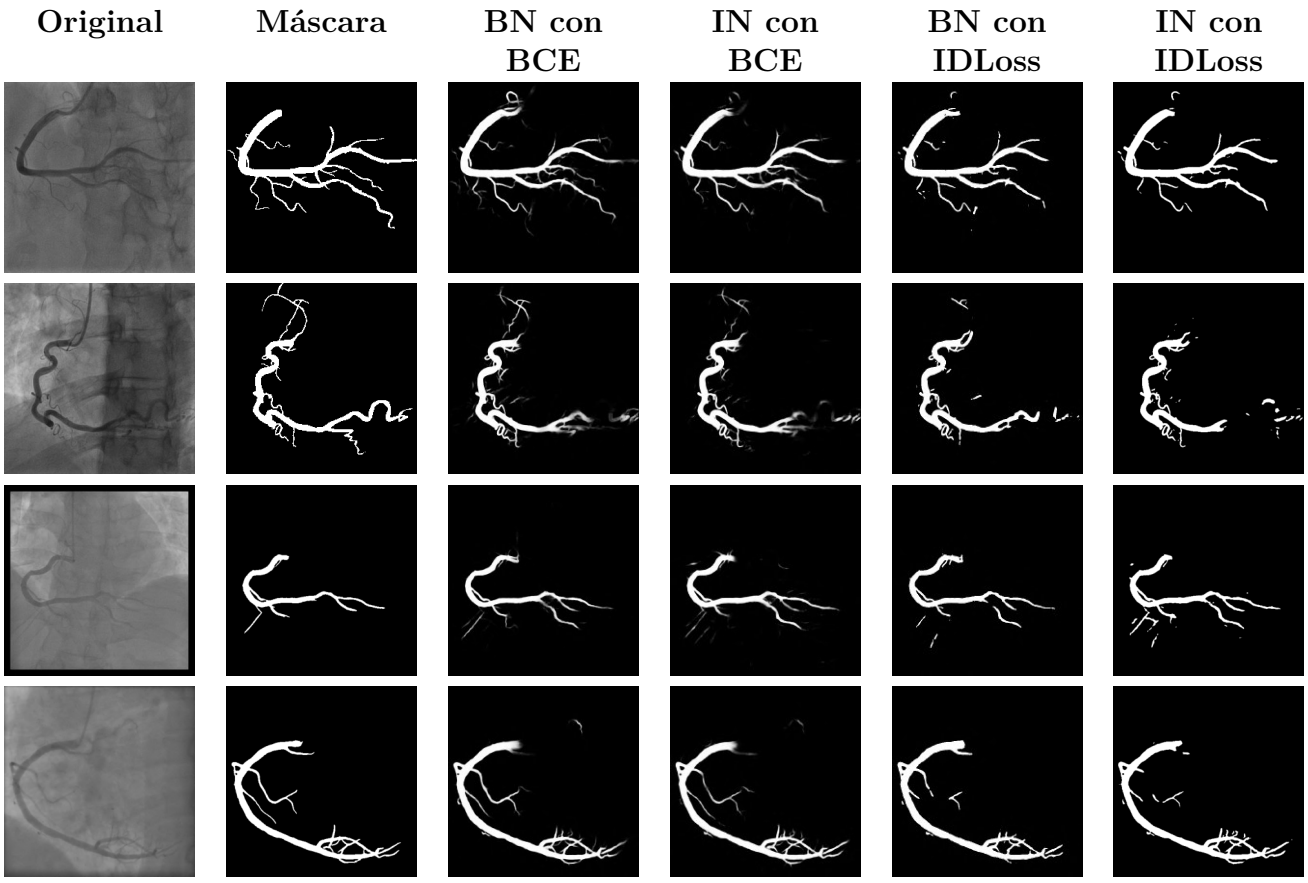


Figura 4.3. Matriz de imágenes RCA : Original, máscara y resultado de la segmentación para las diferentes combinaciones de normalización y función de pérdida.

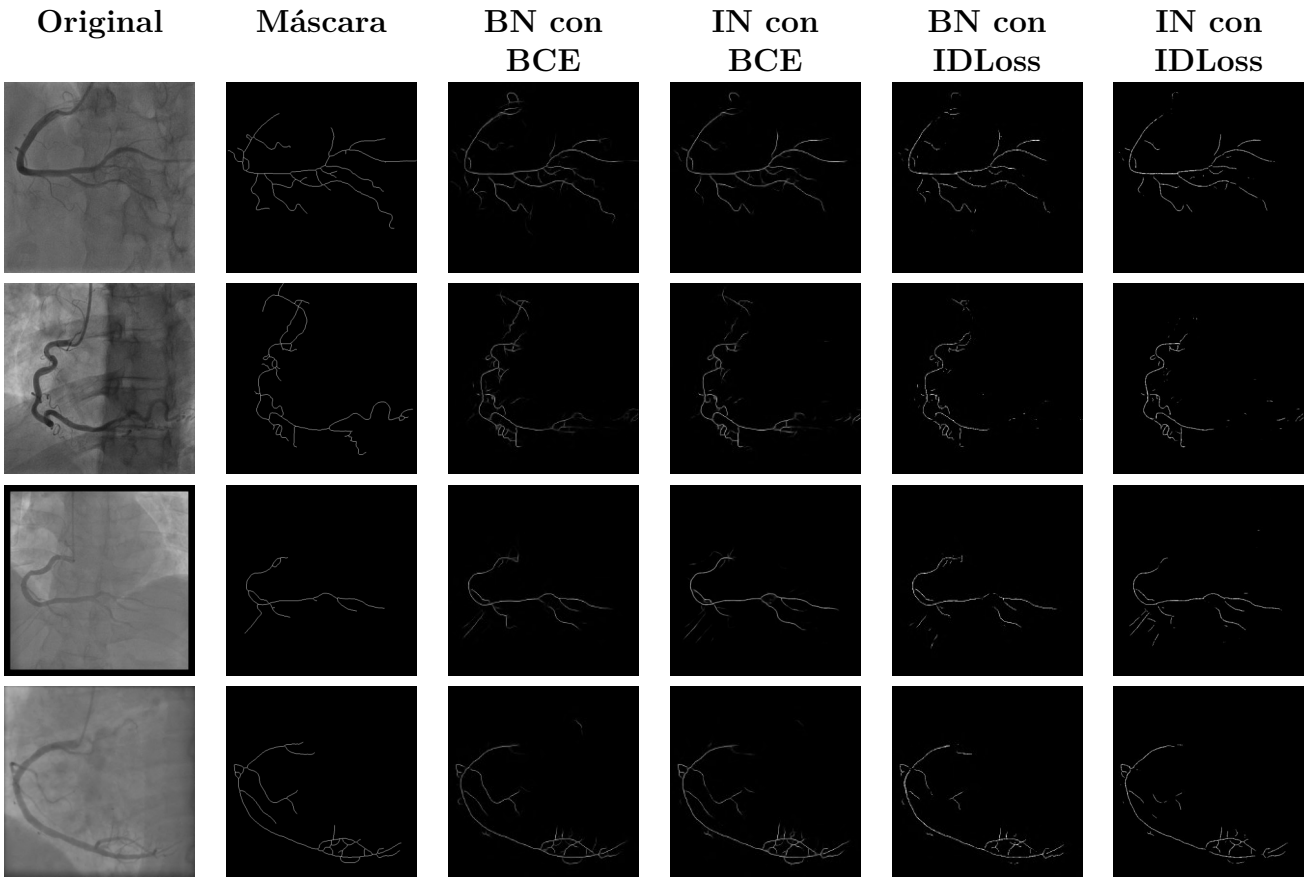


Figura 4.4. Matriz de imágenes RCA : Original, esqueleto y resultado del esqueleto para las diferentes combinaciones de normalización y función de pérdida.

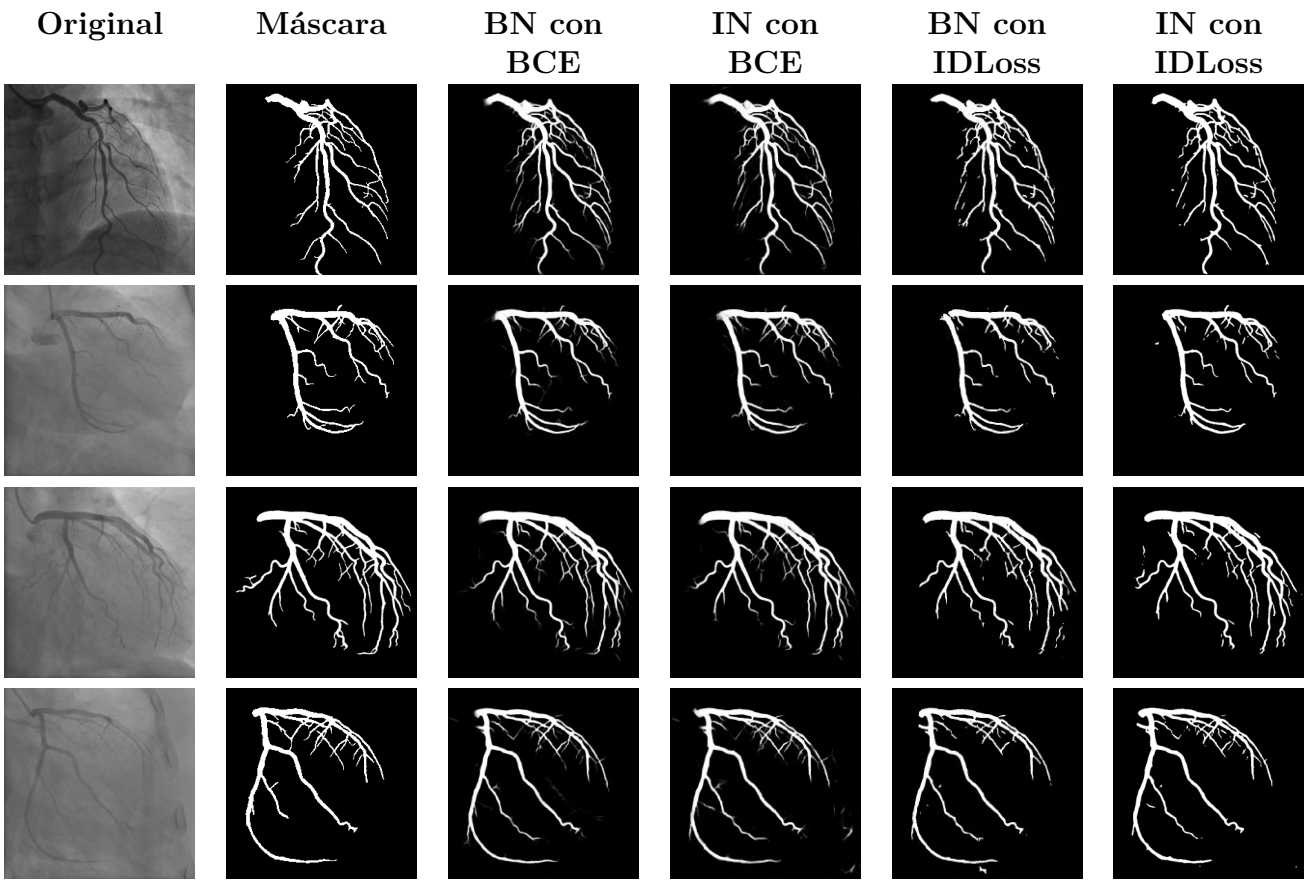


Figura 4.5. Matriz de imágenes LCA : Original, máscara y resultado de la segmentación para las diferentes combinaciones de normalización y función de pérdida.

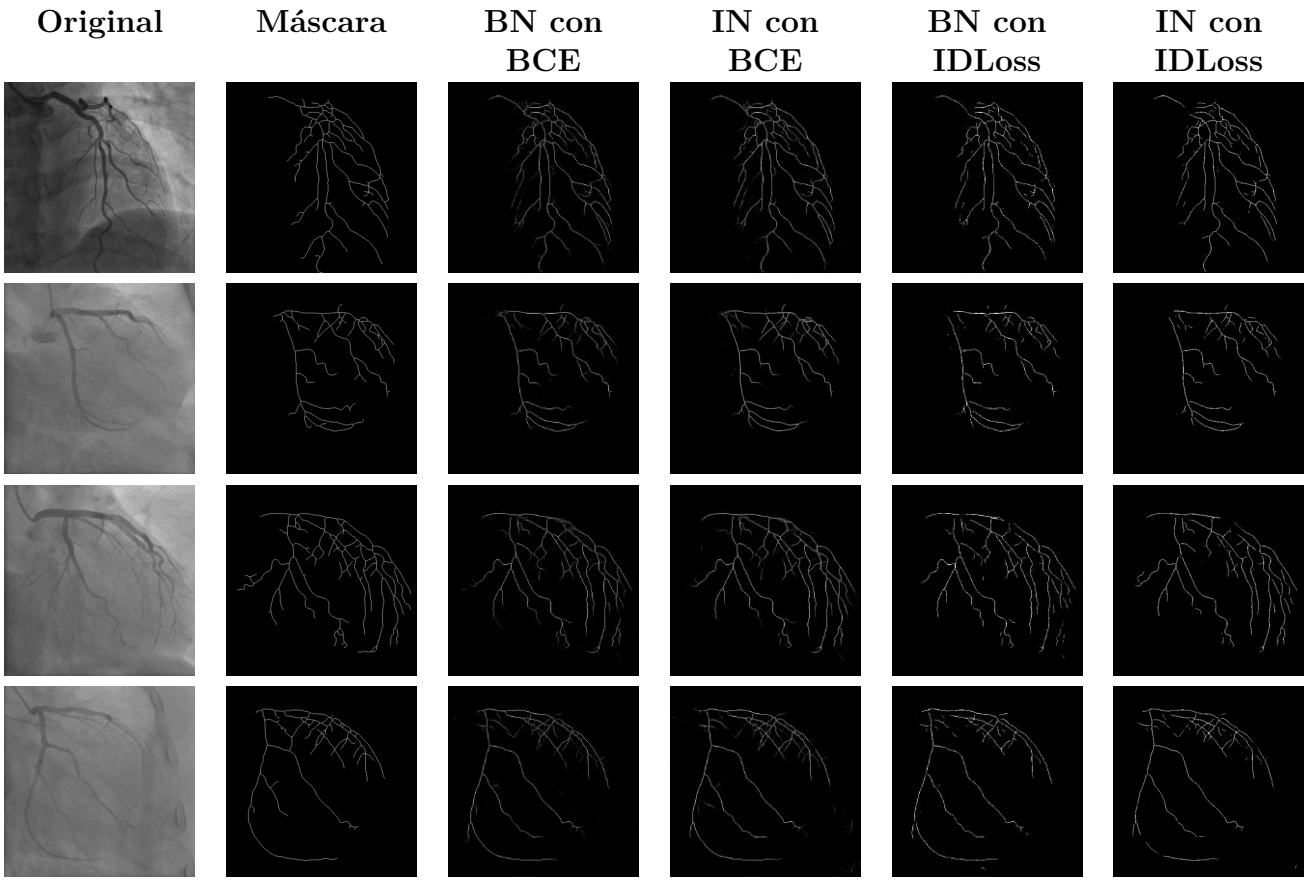


Figura 4.6. Matriz de imágenes LCA : Original, esqueleto y resultado del esqueleto para las diferentes combinaciones de normalización y función de pérdida.

Podemos observar, tanto en las Figura 4.3 como en la Figura 4.5, que los resultados de la segmentación, presentan pequeñas diferencias en comparación con la máscara, al utilizar diferentes combinaciones de normalización y funciones de pérdida. De este modo, determinamos que, para la segmentación, cualquier combinación de parámetros funciona adecuadamente.

Ahora bien, en los resultados en imágenes del esqueleto como se puede ver en la Figura 4.4 y Figura 4.6, se observan diferencias en las combinaciones evaluadas. En las combinaciones de BN con BCE y IN con BCE se obtiene una línea central semejante pero gruesa en comparación al esqueleto original que es una línea central más delgada, más fina de tamaño píxel. Y si vemos las combinaciones de BN con IDLoss e IN con IDLoss los resultados son una línea central delgada de tamaño píxel que es lo que estamos buscando, pero tienen algunos detalles, como al ser líneas delgadas las predicciones generan algunos espacios.

A partir de los resultados de la Tabla 4.2 y la Tabla 4.3, aplicamos módulos de atención CBAM para mejorar el resultado de la predicción de la segmentación y el esqueleto, seleccionando el modelo que implementa IN con IDLoss. Las métricas obtenidas tras esta mejora se presentan en la Tabla 4.4 y la Tabla 4.5.

Tabla 4.4. Segmentación con Multi-task Attention U-Net

Modelo	**Attention layer**	**Loss**	**IoU**	**Dice**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Hausdorff Distance**
Multi-task U-Nett	N/A	**0.21605**	0.75662	0.85841	0.98413	0.87251	0.85684	0.86358	**74.29862**
Multi-task Attention U-Net	CBAM	0.22418	**0.77335**	**0.86954**	**0.98518**	**0.87302**	**0.87634**	**0.87393**	87.02019

Tabla 4.5. Esqueleto con Multi-task Attention U-Net

Modelo	**Attention layer**	**Loss**	**IoU**	**Dice**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Hausdorff Distance**
Multi-task U-Nett	N/A	**0.09529**	0.40152	0.56793	**0.99322**	**0.57104**	0.58577	0.57754	98.57278
Multi-task Attention U-Net	CBAM	0.09958	**0.40551**	**0.57252**	0.99309	0.55999	**0.60339**	**0.58014**	**85.43610**

Los resultados obtenidos en imágenes para la segmentación y el esqueleto con el módulo de atención CBAM se compararon contra los resultados sin módulo de atención, con el objetivo de evaluar si se observan mejoras en las imágenes. Estos resultados se presentan en la Figura 4.7 y Figura 4.8 para imágenes RCA, así como en la Figura 4.9 y Figura 4.10 para imágenes LCA.

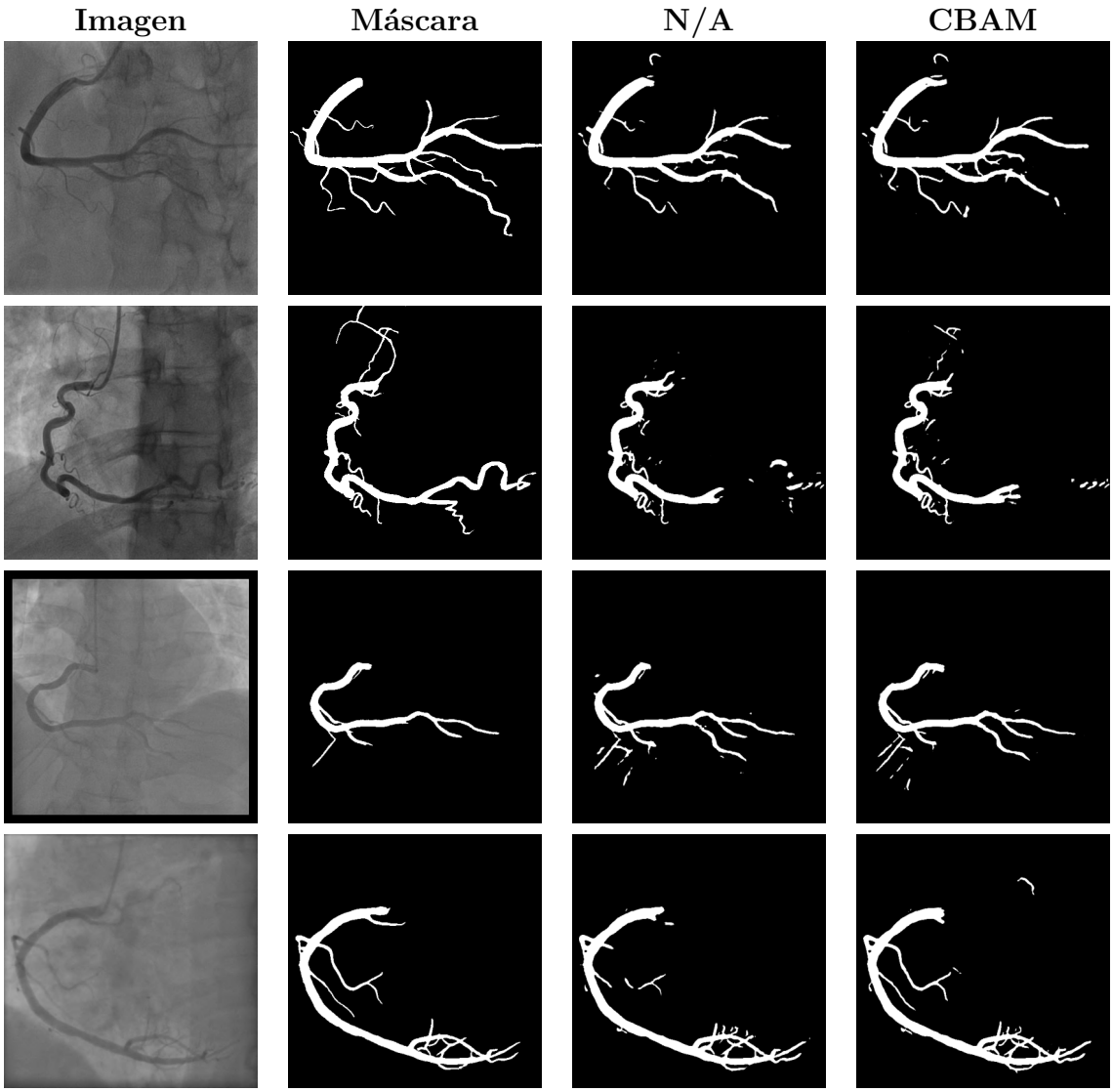


Figura 4.7. Matriz de imágenes RCA : Imagen, máscara y resultado de la segmentación sin módulo de atención y con CBAM.

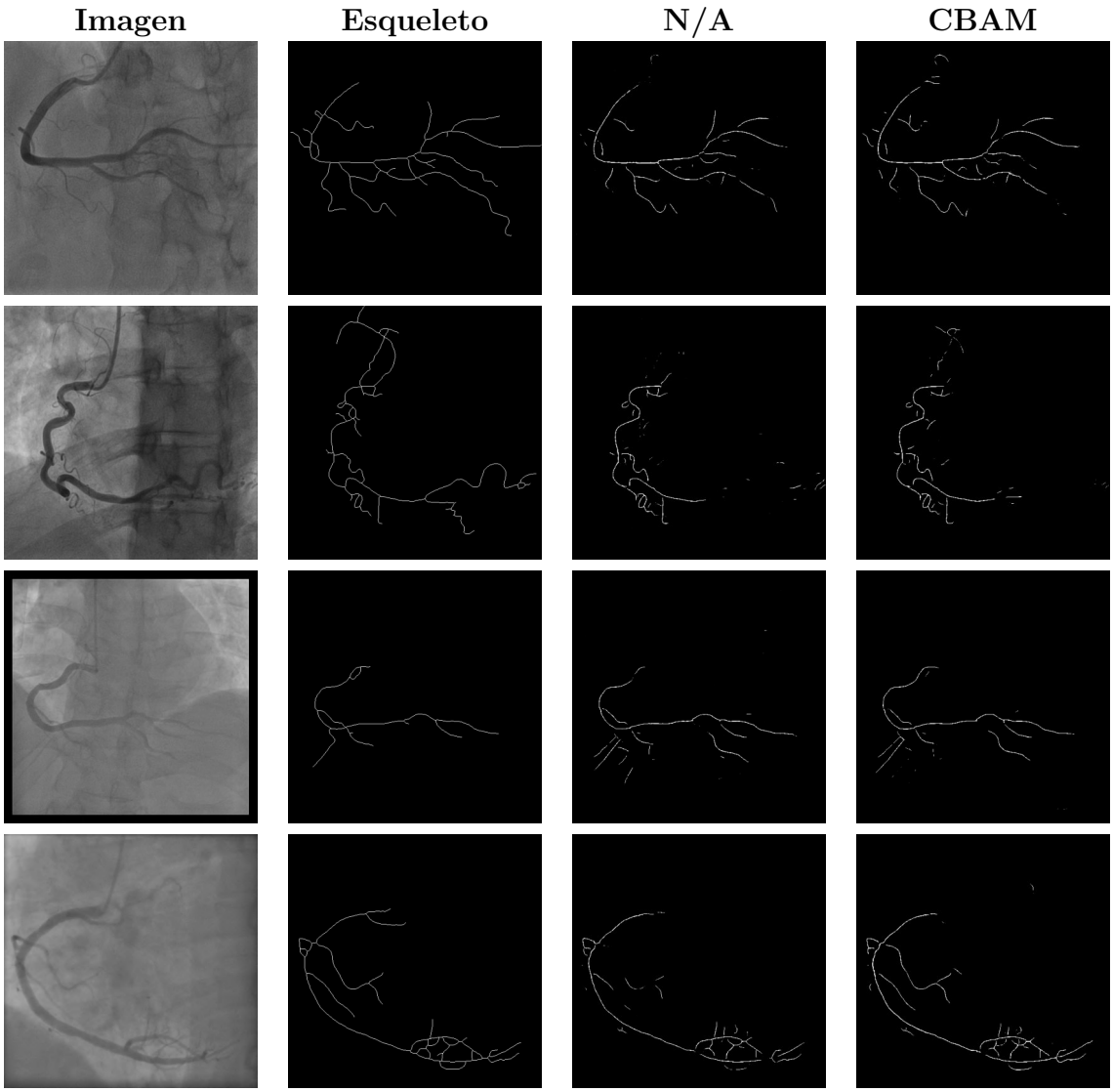


Figura 4.8. Matriz de imágenes RCA : Imagen, esqueleto y resultado del esqueleto sin módulo de atención y con CBAM.

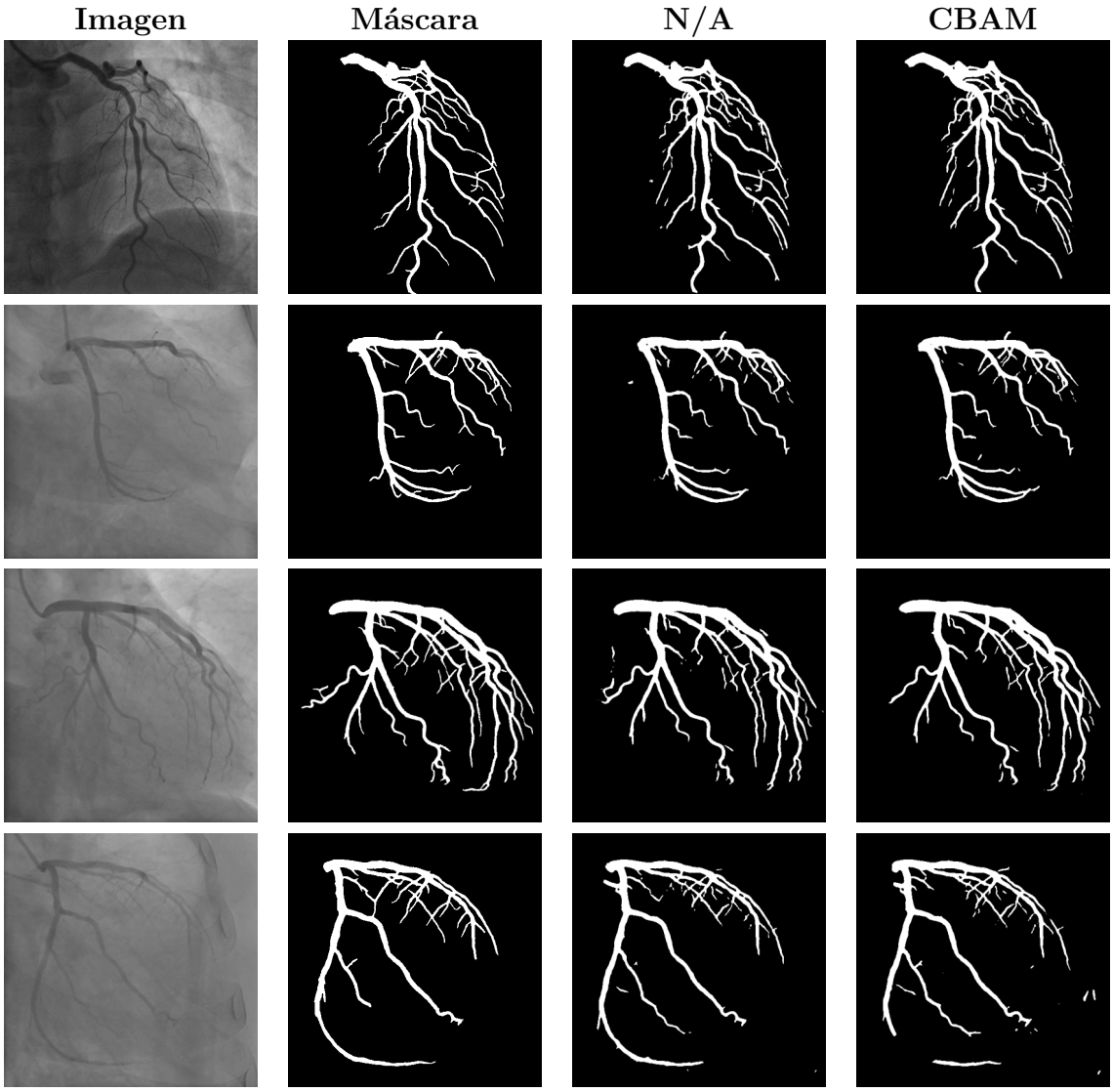


Figura 4.9. Matriz de imágenes LCA : Imagen, máscara y resultado de la segmentación sin módulo de atención y con CBAM.

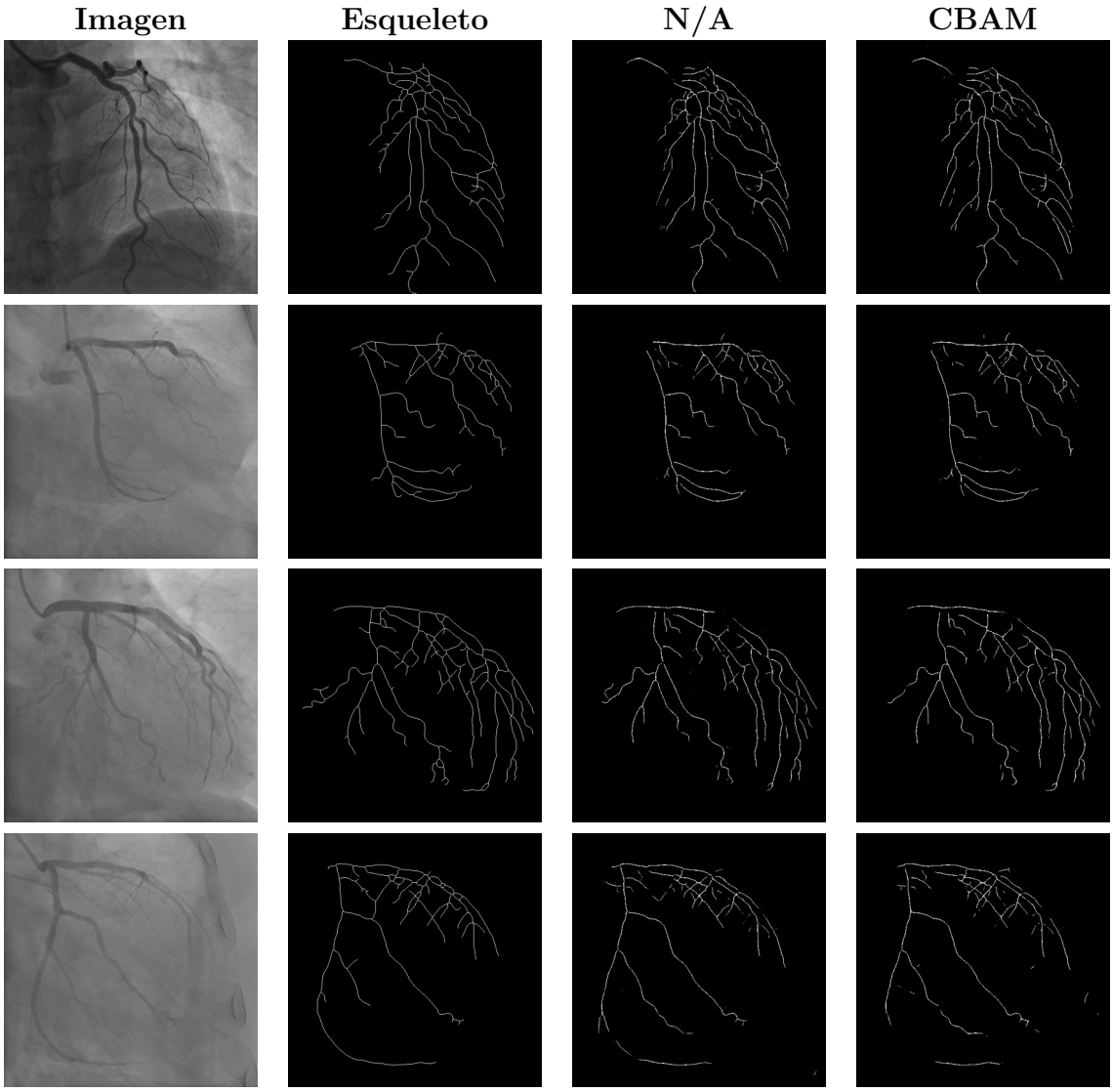


Figura 4.10. Matriz de imágenes LCA : Imagen, esqueleto y resultado del esqueleto sin módulo de atención y con CBAM.

Si nos centramos en la Figura 4.8 y Figura 4.10, al utilizar el módulo de atención CBAM, se observa una mejoría en la línea central de la predicción del esqueleto, en comparación con los resultados sin CBAM. Una vez entrenada la red y obtenido el mejor modelo, este se utilizó con la base de pruebas para generar las predicciones tanto de segmentación como del esqueleto de las angiografías coronarias. A partir de los esqueletos obtenidos, procedimos con la extracción de puntos clave.

El objetivo es identificar los puntos iniciales, finales y bifurcaciones presentes en el esqueleto. Sin embargo, debido a la naturaleza de la predicción del esqueleto, se generan espacios a lo largo de la línea central, lo que provoca la aparición de una gran cantidad de

puntos, como se muestra en la Figura 4.11 y Figura 4.12. Este comportamiento se presenta en todas las combinaciones evaluadas, independientemente de la normalización, la función de pérdida y el uso o no de los módulos de atención.

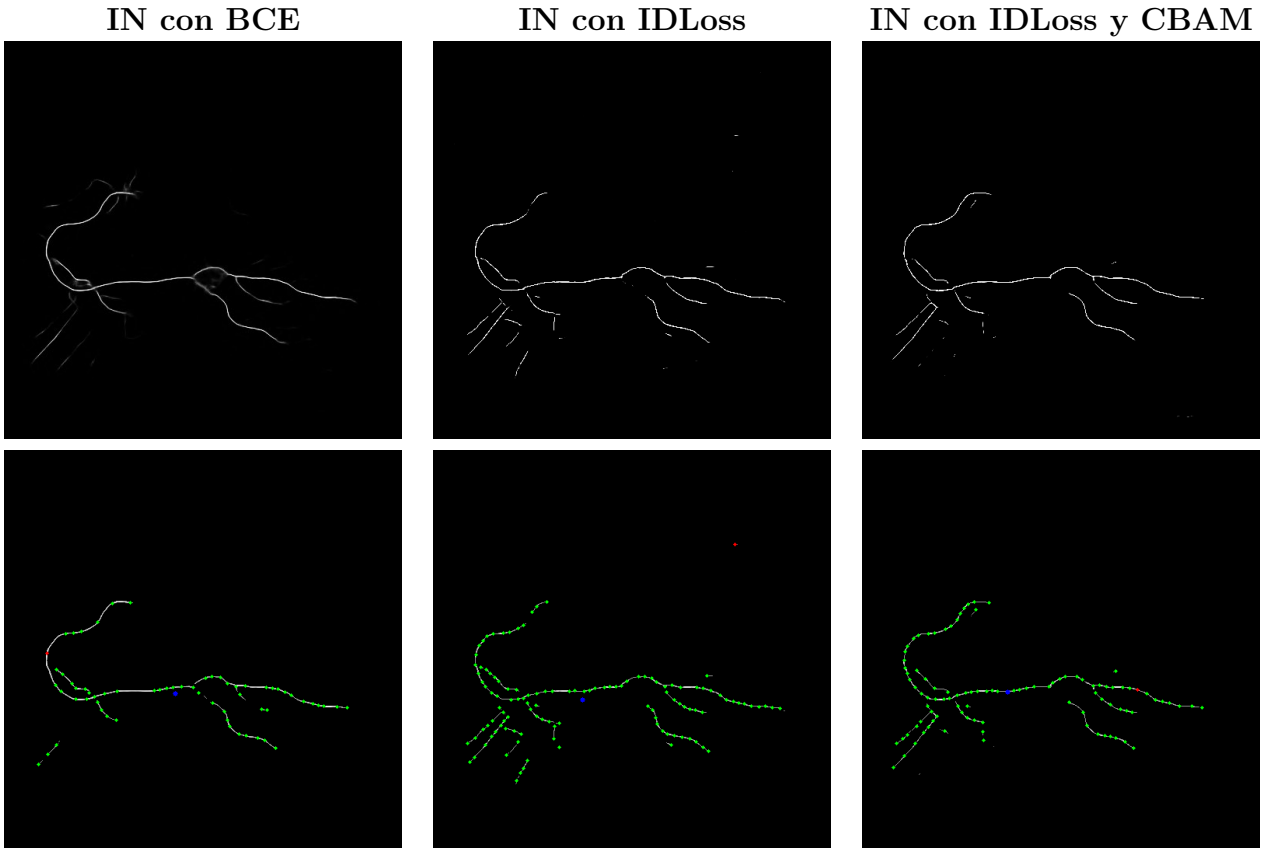


Figura 4.11. Extracción de puntos en esqueletos de imágenes RCA sin procesar.

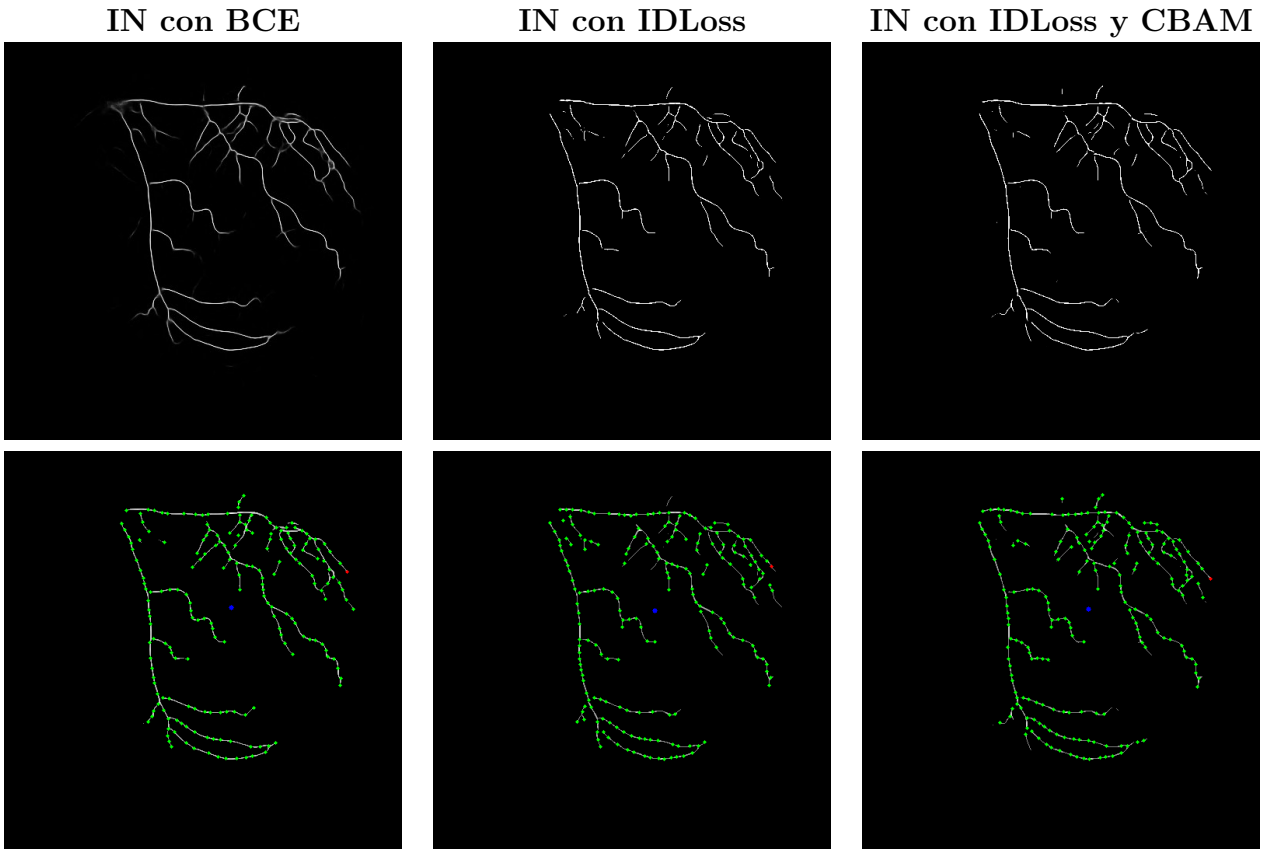


Figura 4.12. Extracción de puntos en esqueletos de imágenes LCA sin procesar.

Por lo tanto realizamos un procesamiento morfológico a los esqueletos descrito en la metodología para mejorar su línea central como se puede ver en la Figura 4.13 y Figura 4.14.

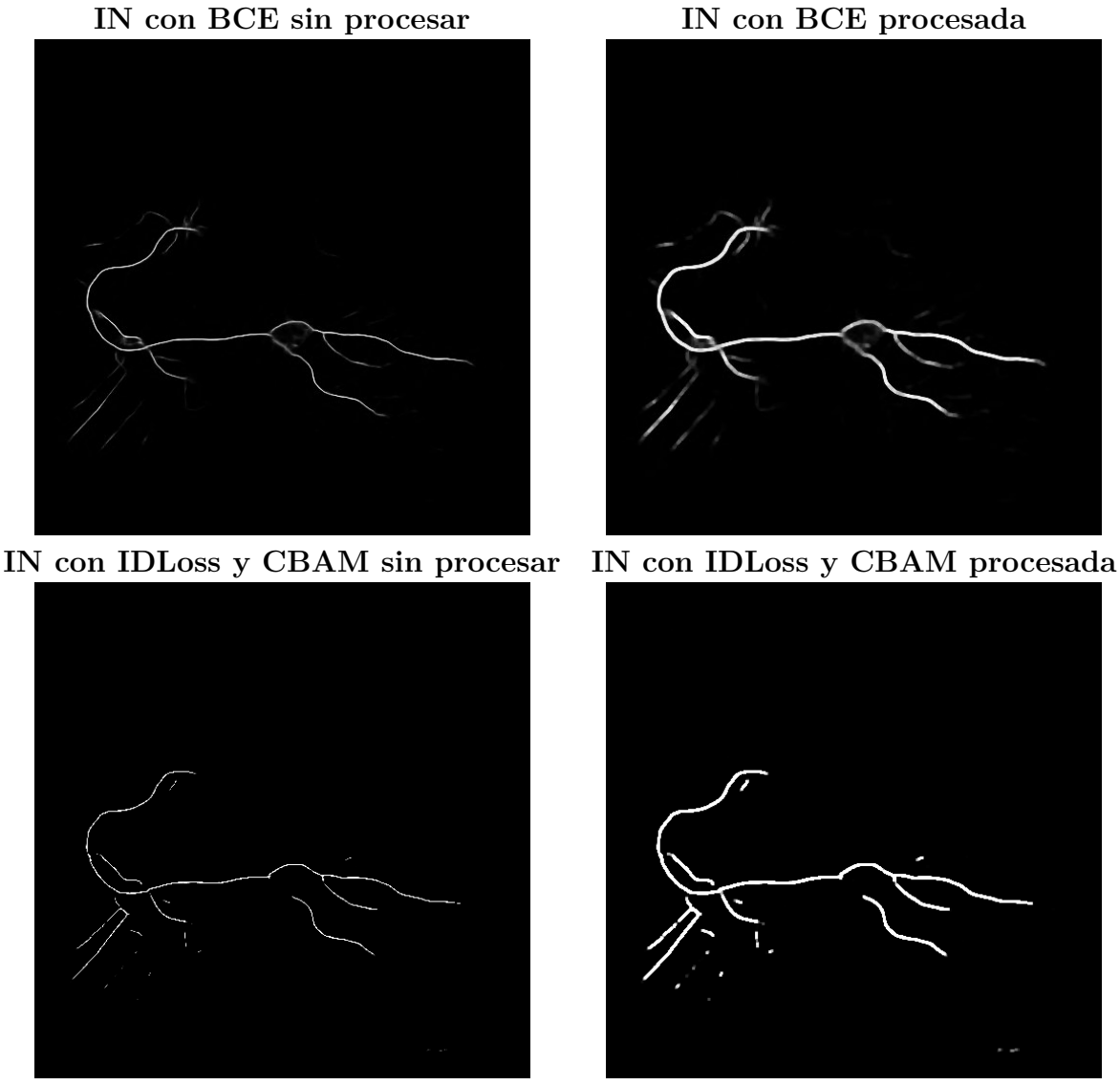


Figura 4.13. Comparación de imágenes RCA sin procesar y procesadas.

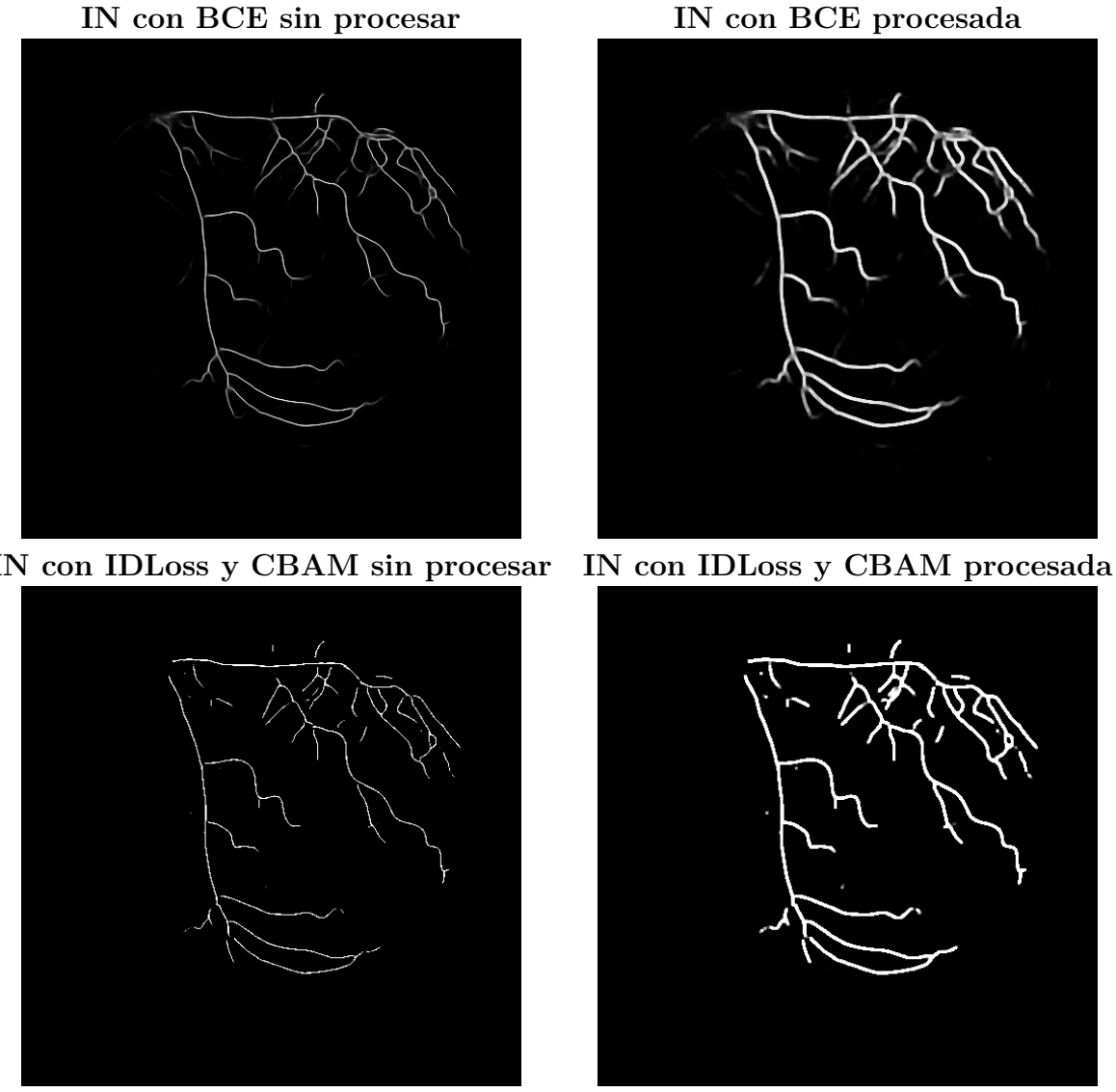


Figura 4.14. Comparación de imágenes LCA sin procesar y procesadas.

Se aprecian de manera más amplia las imágenes en la Figura 4.13 y Figura 4.14, donde se muestran tras el procesamiento las imágenes con IN, IDLoss y CBAM. Presentan una línea central más sólida y limpia en comparación de las imágenes que tiene IN y BCE, las cuales, al ser procesadas aparecen más difusas con ruido de fondo. Por lo tanto, para la extracción de puntos clave las imágenes con CBAM procesadas son las que usamos con los parámetros de cantidad máxima de puntos a detectar de 100, la calidad mínima de 0.09 y distancia mínima entre puntos de 11 como se ve en la Figura 4.15 y Figura 4.16.

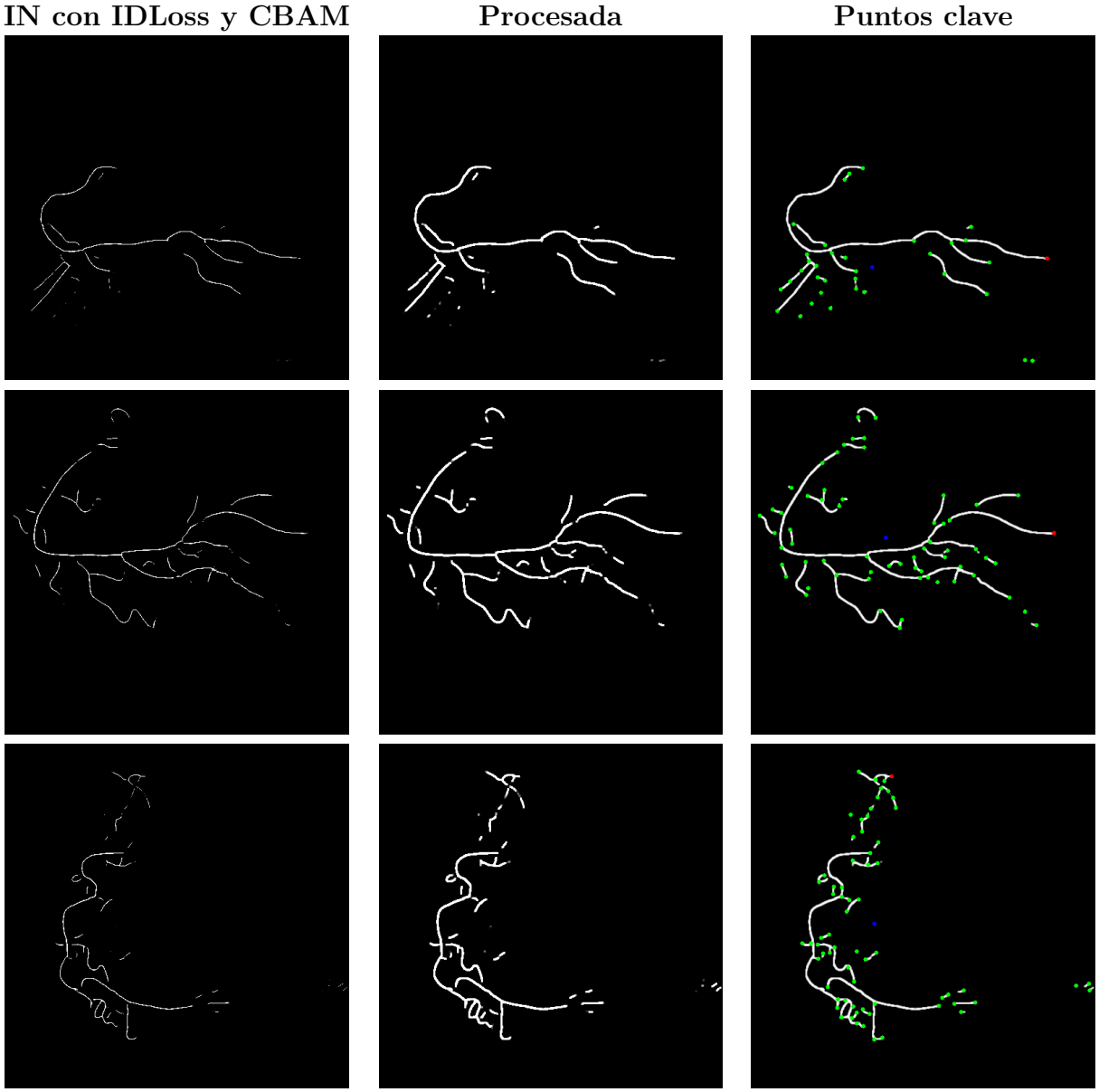


Figura 4.15. Extracción de puntos en esqueletos de imágenes RCA procesadas.

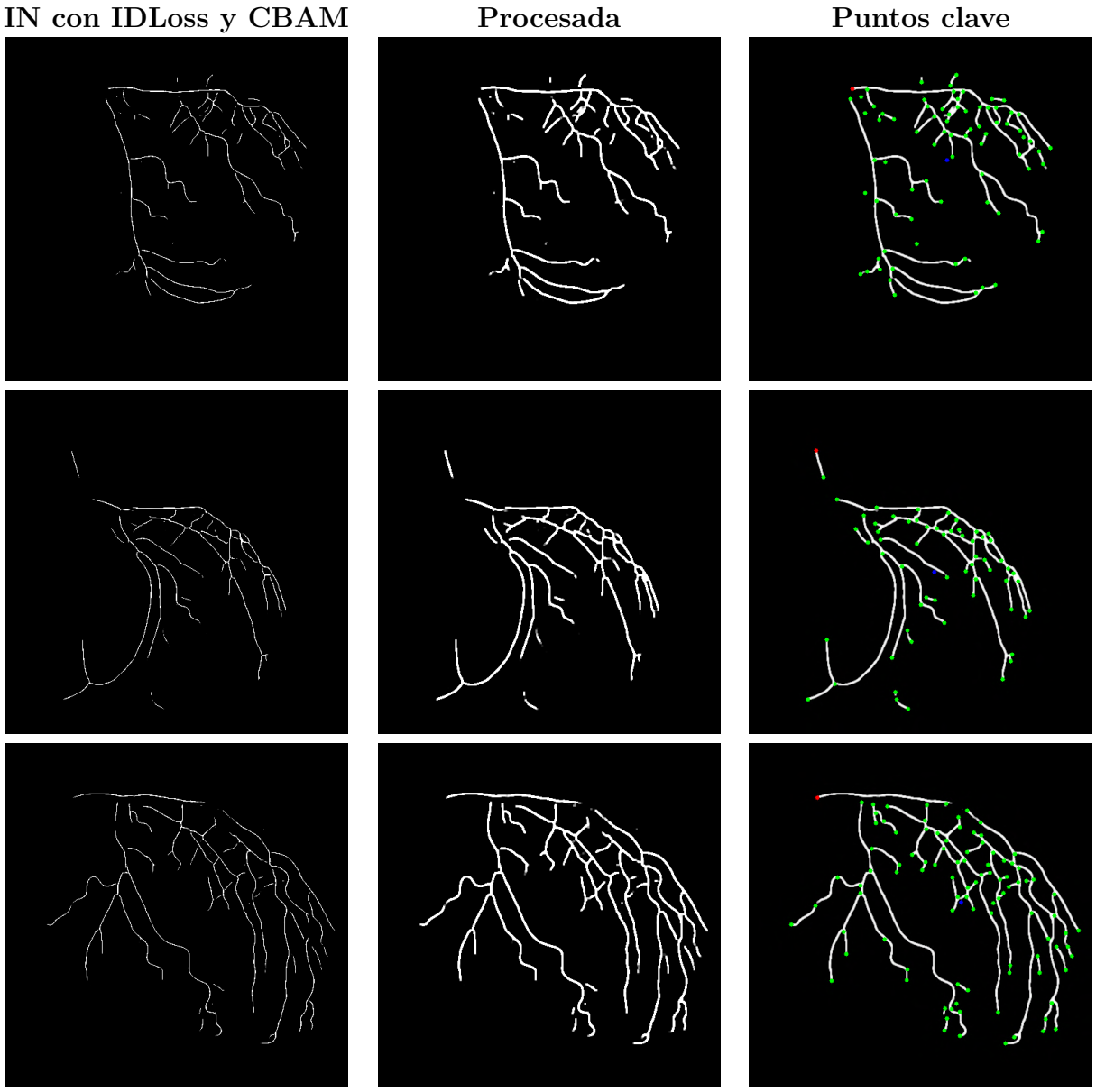


Figura 4.16. Extracción de puntos en esqueletos de imágenes LCA procesadas.

Tanto en la Figura 4.15 como en la Figura 4.16, se muestran los puntos clave en verde, mientras que el punto rojo representa la orientación de la imagen: si la imagen es RCA el punto rojo se encuentra a la derecha; si es LCA el punto rojo está a la izquierda del centroide. Además, el punto azul indica la posición del centroide del esqueleto.

La siguiente fase del proceso consiste en seleccionar de manera aleatoria algunos de los puntos clave obtenidos previamente para generar transformaciones locales en ciertas regiones alrededor de dichos puntos. Para ello, se define un tamaño de región a transformar dentro de

la arteria coronaria utilizando un kernel de 55×55. Para determinar qué tan suave o fuerte es la transformación utilizamos el parámetro α, el cual nos permite regular la intensidad de la transformación. De esta manera controlamos que tanto transformamos la región de la arteria coronaria.

Probando diferentes transformaciones desde algo ligeramente perceptibles con un α de 55 y subiendo el valor poco a poco a un α de 105, nos permite hacer ligeros cambios para generar distinción entre la imagen real y la imagen sintética como se observa en la Figura 4.17 y la Figura 4.19 donde se muestra el acercamiento de una de las regiones transformadas en las imágenes.

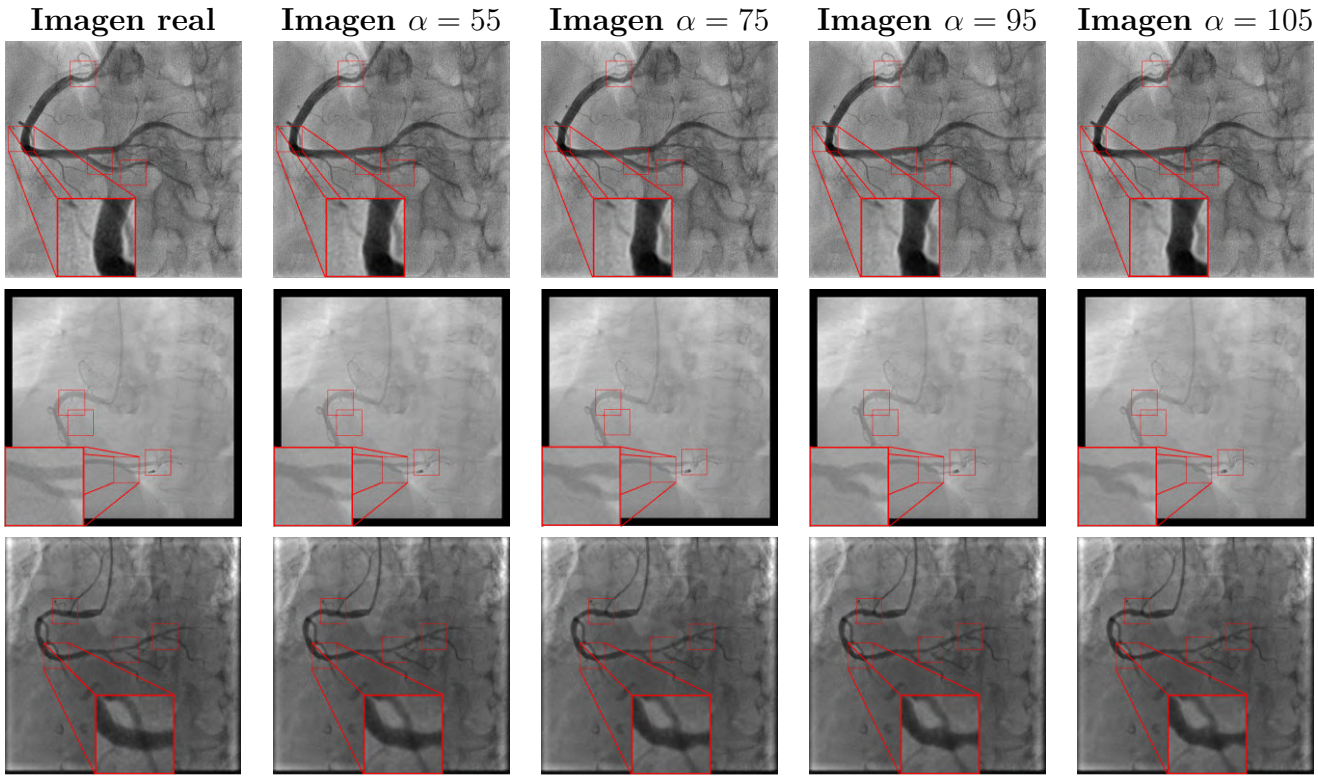


Figura 4.17. Comparación de imágenes RCA entre imagen real contra imágenes sintéticas con diferentes valores de α.

Además, al modificar las regiones de la imagen real ajustando el valor α es posible simular una condición médica como la estenosis, que consiste en el estrechamiento de las paredes de la arteria.

Mediante este método de *data augmentation* podemos alterar la forma de la arteria en ciertas regiones, generando imágenes sintéticas nuevas para aumentar la base de datos. esto incluye la posibilidad de modificar la imagen al punto de generar principios de estenosis

en algunas imágenes. En particular, como se observar en la Figura 4.17 y la Figura 4.19, en algunas regiones donde se aplicó el método las arterias fueron reduciendo su tamaño generando el efecto de cuello de botella.

De igual manera, es posible aplicar las transformaciones en las máscaras reales para generar máscaras sintéticas, como se muestra en la Figura 4.18 y Figura 4.20.

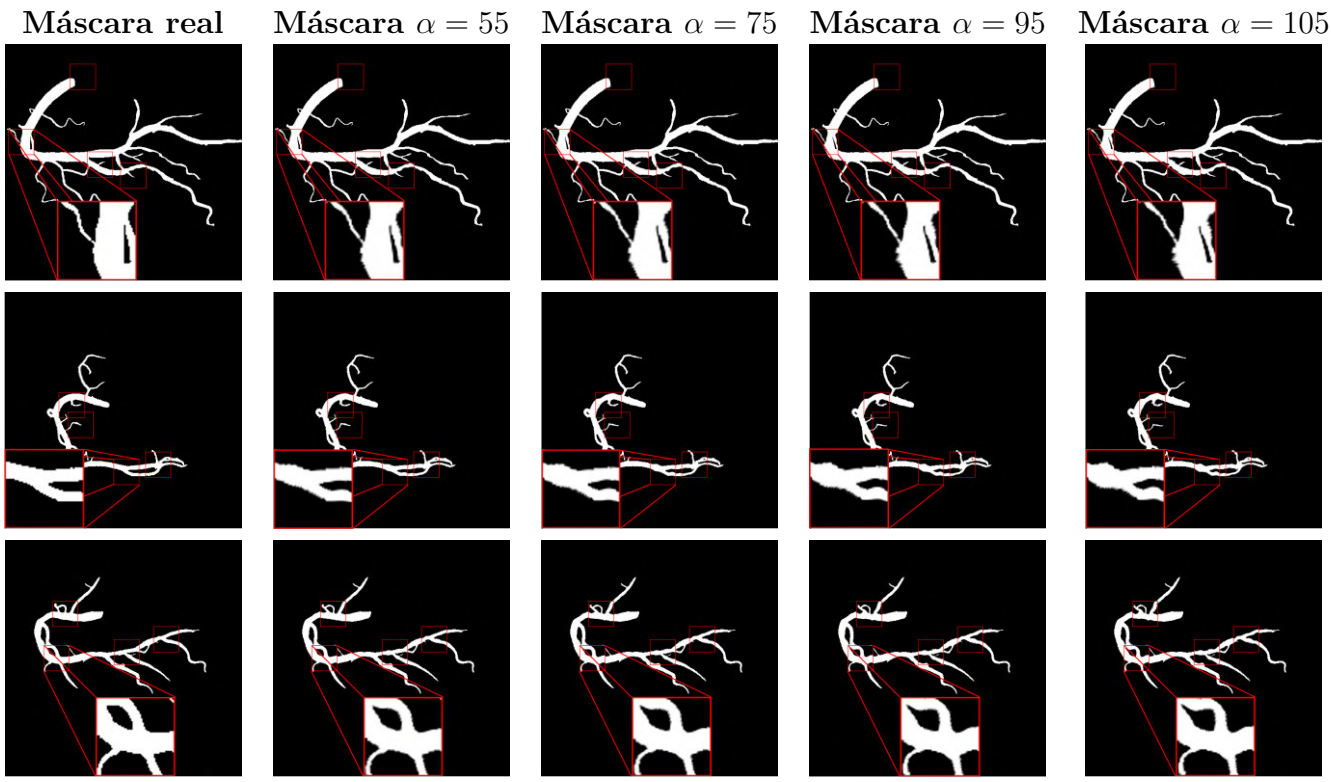


Figura 4.18. Comparación de máscaras RCA entre máscara real contra máscaras sintéticas con diferentes valores de α.

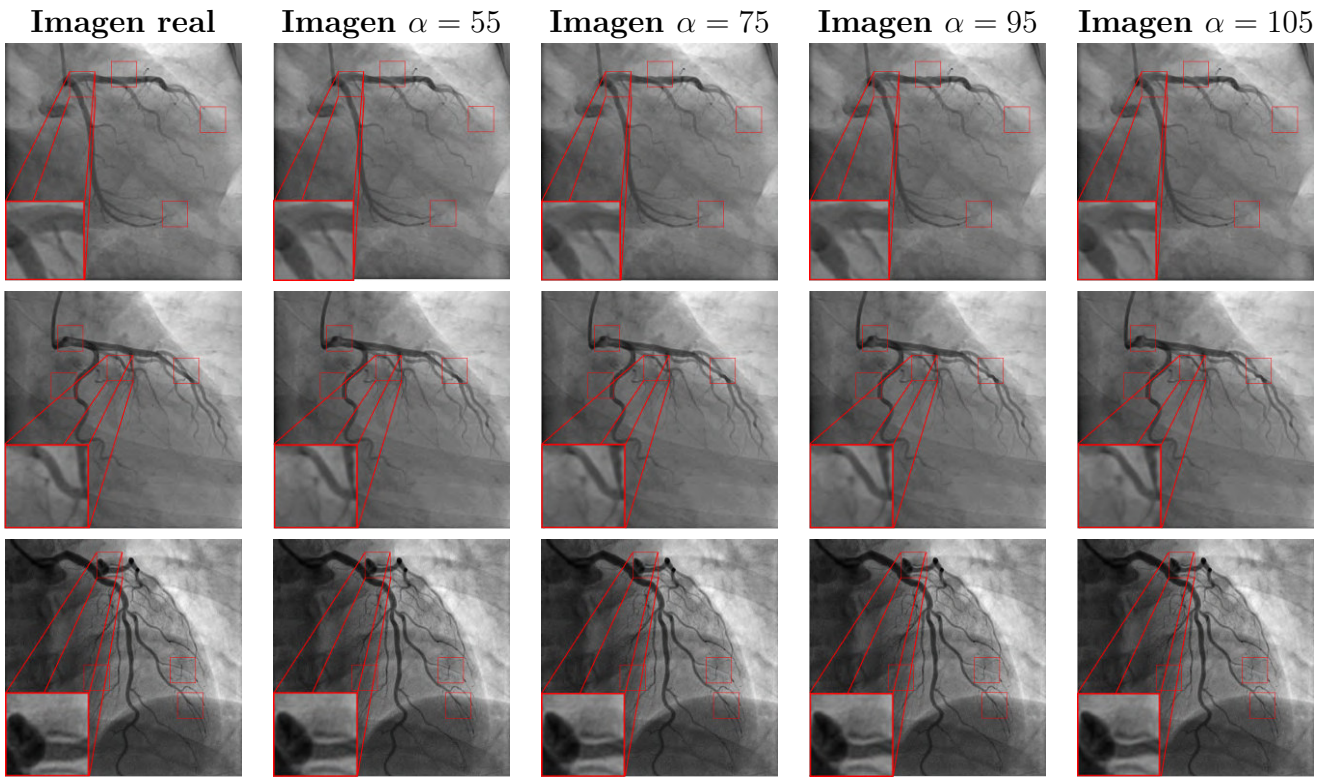


Figura 4.19. Comparación de imágenes LCA entre imagen real contra imágenes sintéticas con diferentes valores de α.

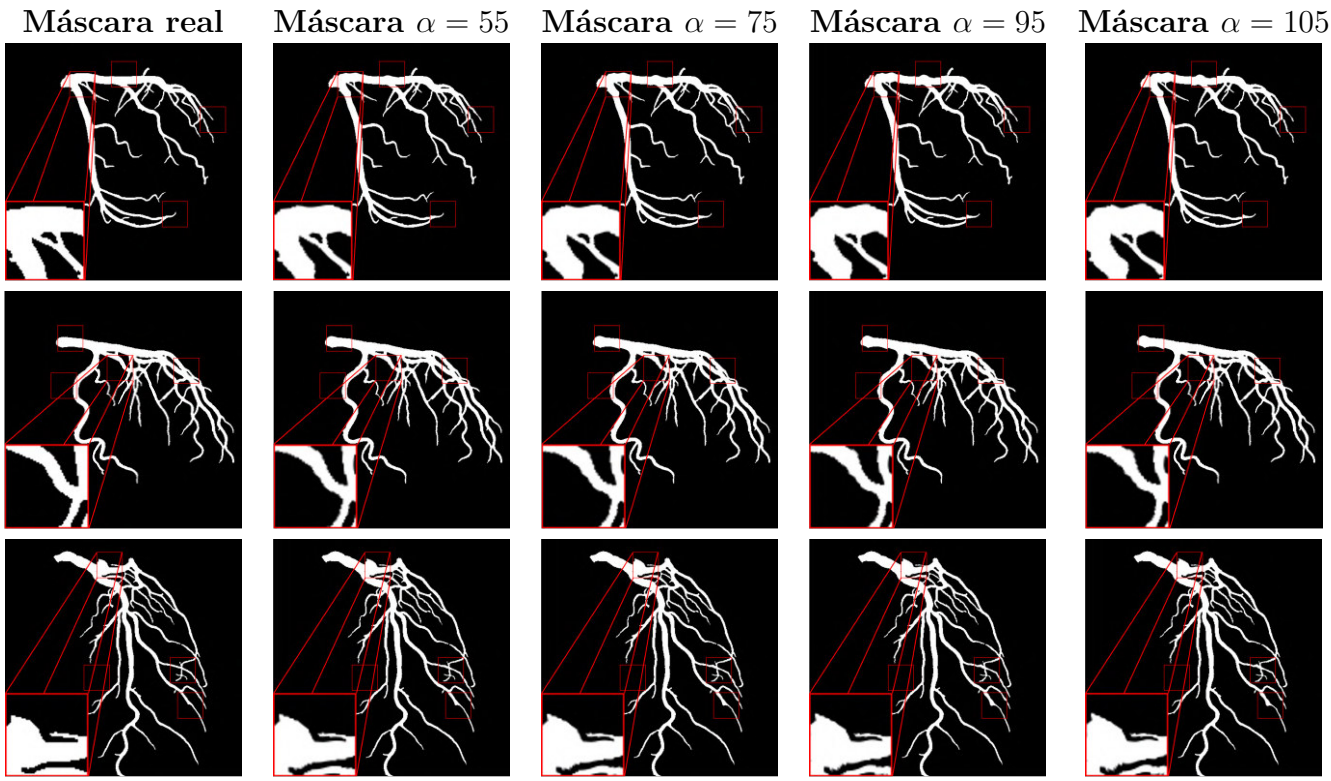


Figura 4.20. Comparación de máscaras LCA entre máscara real contra máscaras sintéticas con diferentes valores de α.

Podemos decir que la metodología del modelo generativo cumple con el objetivo de crear imágenes sintéticas a partir de la extracción de puntos clave, obtenidos del esqueleto generado mediante la Multi-task Attention U-Net previamente entrenada con las imágenes reales. El enlace al repositorio del código [Ramos, 2025].

CAPÍTULO 5

Conclusiones

Dentro de este capítulo de conclusiones, presentamos las observaciones y reflexiones hechas a lo largo de este trabajo, abarcando tanto el desarrollo como la culminación de los objetivos planteados, además de discutir los trabajos futuros relacionados con esta investigación.

Principalmente, podemos concluir que se cumple con el objetivo general de este trabajo al completar los objetivos específicos, lo que llevó al desarrollo de un modelo generativo capaz de generar imágenes sintéticas de angiografías coronarias de rayos X.

La generalización de la segmentación y esqueletización mediante *deep learnign* nos condujo a la implementación de un modelo basado en la arquitectura U-Net. Se observó que al reducir los filtros de la U-Net de 64 a 16, el modelo mantiene la generalización con un costo computacional menor, al reducir el número de parámetros de 31M a 1.9M, logrando además mejores resultados en las métricas de Loss (0.03855), el mayor IoU (0.77315), Dice (0.87010), Accuracy (0.98554), Recall (0.86267) y F1 Score (0.87470). Además, obtiene la menor Distancia de Hausdorff (69.25949). La U-Net de 16 filtros se utilizó para el modelo Multi-task de 2.7M de parámetros para hacer la generación simultanea de la segmentación y esqueletización. A este modelo se le agregaron módulos de atención, para enfocarse en la generalización del esqueleto.

En este caso algunas de las métricas para la segmentación mejoraron en comparación del modelo sin atención con un Loss (0.22418), el mayor IoU (0.77335), Dice (0.86954), Accuracy (0.98518),Precision (0.87302), Recall (0.87634) y F1 Score (0.87393), Distancia de Hausdorff (87.02019). Para el esqueleto si se tuvo una mejora en la mayoría de las métricas con un Loss (0.0.09958), el mayor IoU (0.40551), Dice (0.57252), Accuracy (0.99309),Precision (0.55999), Recall (0.60339) y F1 Score (0.58014), Distancia de Hausdorff (85.43610) y visualmente en la generalización de las imágenes. A los esqueletos se les procesa morfológicamente antes de la extracción de puntos clave, aunque la generalización de los esqueletos por medio de la Multi-task Attention U-Net es buena, las predicciones a nivel píxel dejan espacios entre un píxel y otro para reducir esos espacios se hace este proceso morfológico. La extracción de puntos clave se llevó acabo por medio del método Shi-Tomasi el cual se caracteriza por ser rápido y eficiente ya que se enfoca en los valores mínimos de los autovalores de la matriz de segundo momento. Posteriormente la realización de imágenes sintéticas se realizó por medio de transformaciones locales en ciertas regiones de las imágenes a través de kernels que son creados alrededor de los puntos clave. Estos kernels permiten modificar las regiones de la imagen sin alterar el resto de ella y así generar imágenes sintéticas, las cuales son imágenes parecidas a las originales, pero con modificaciones en algunos puntos clave. Dado que es difícil conseguir imágenes médicas para el entrenamiento de modelos *deep learning*, este método generativo permite expandir la base de datos de imágenes.

Como última observación este método esta enfocado principalmente a imágenes médicas de angiografías coronarias pero podemos teorizar en que otros campos puede ser empleado, siguiendo la linea de imágenes médicas tenemos las *Digital Subtraction Angiography* (DSA) las cuales proporcionan una imagen de los vasos sanguíneos del cerebro y la retinografía proporciona una imagen de los vasos sanguíneos de la retina. En otra área como imágenes satelitales de ríos ya que tiene una topología semejante a los vasos sanguíneos, se puede obtener un esqueleto fluvial donde existan puntos clave, para generar imágenes sintéticas con alteraciones geográficas, les pueden dar uso para entrenar una red para el reconocimiento de zonas geográficas o la predicción de zonas de inundación. En botánica las venas de las hojas se parecen a los vasos sanguíneos, entonces se pueden generar imágenes sintéticas con variaciones entre sus venas, con ello entrenar una red para clasificar especies de plantas.

Para mejorar este estudio, se proponen las siguientes líneas de investigación:

1. Optimizar el modelo Multi-task mediante diferentes configuraciones en los módulos de atención.

2. Desarrollar una nueva estrategia para el ordenamiento y clasificación de puntos clave.
3. Focalización y reducción de kernels para mejorar la generación de imágenes.
4. Explorar la aplicabilidad de este método en otros tipos de imágenes.
5. Desarrollar una red punto a punto, aparte de generalizar la segmentación y el esqueleto que pueda generar los puntos claves.

Referencias

AlAmir, M., y AlGhamdi, M. (2022). The role of generative adversarial network in medical image analysis: An in-depth survey. *ACM Computing Surveys*, *55*(5), 1–36. Descargado de https://doi.org/10.1145/352784 doi: 10.1145/352784

Azad, R., Heidary, M., Yilmaz, K., Hüttemann, M., Karimijafarbigloo, S., Wu, Y., ... Merhof, D. (2023). Loss functions in the era of semantic segmentation: A survey and outlook. *arXiv preprint arXiv:2312.05391*. Descargado de https://doi.org/10.48550/arXiv.2312.05391

Castro, E., Cardoso, J. S., y Pereira, J. C. (2018). Elastic deformations for data augmentation in breast cancer mass detection. En *2018 ieee embs international conference on biomedical & health informatics (bhi)* (pp. 230–234). Descargado de https://doi.org/10.1109/BHI.2018.8333411 doi: 10.1109/BHI.2018.8333411

Dodge, S., y Karam, L. (2017). A study and comparison of human and deep learning recognition performance under visual distortions. En *2017 26th international conference on computer communication and networks (icccn)* (pp. 1–7). Descargado de https://doi.org/10.1109/ICCCN.2017.8038465 doi: 10.1109/ICCCN.2017.8038465

Du, T., Xie, L., Zhang, H., Liu, X., Wang, X., Chen, D., ... others (2021). Training and validation of a deep learning architecture for the automatic analysis of coronary angiography:

Automatic recognition of coronary angiography. *EuroIntervention*, *17*(1), 32. Descargado de https://doi.org/10.4244%2FEIJ-D-20-00570 doi: 10.4244/EIJ-D-20-00570

Dumoulin, V., y Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*. Descargado de https://doi.org/10.48550/arXiv.1603.07285

Gong, T., Lee, T., Stephenson, C., Renduchintala, V., Padhy, S., Ndirango, A., ... Elibol, O. H. (2019). A comparison of loss weighting strategies for multi task learning in deep neural networks. *IEEE Access*, *7*, 141627–141632. Descargado de https://doi.org/10.1109/ACCESS.2019.2943604

Heijmans, H. J., y Ronse, C. (1990). The algebraic basis of mathematical morphology i. dilations and erosions. *Computer Vision, Graphics, and Image Processing*, *50*(3), 245–295. Descargado de https://doi.org/10.1016/0734-189X(90)90148-O

Hwang, M., Hwang, S.-B., Yu, H., Kim, J., Kim, D., Hong, W., ... others (2021). A simple method for automatic 3d reconstruction of coronary arteries from x-ray angiography. *Frontiers in Physiology*, *12*, 724216. Descargado de https://doi.org/10.3389/fphys.2021.724216 doi: doi.org/10.3389/fphys.2021.724216

Ibrahim, M., Khalil, Y. A., Amirrajab, S., Suna, C., Breeuwer, M., Pluim, J., ... Dumontiera, M. (2024). Generative ai for synthetic data across multiple medical modalities: A systematic review of recent developments and challenges. *arXiv preprint arXiv:2407.00116*. Descargado de https://doi.org/10.1016/j.compbiomed.2025.109834

INEGI. (2022). *Estadísticas de defunciones registradas (edr).* https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2024/EDR/EDR2023_En-Jn.pdf.

Ioffe, S. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*. Descargado de https://doi.org/10.48550/arXiv.1502.03167

Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., y Maier-Hein, K. H. (2021). nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, *18*(2), 203–211. Descargado de https://doi.org/10.1038/s41592-020-01008-z

Kazerouni, A., Aghdam, E. K., Heidari, M., Azad, R., Fayyaz, M., Hacihaliloglu, I., y Merhof, D. (2022). Diffusion models for medical image analysis: A comprehensive survey.

arXiv preprint arXiv:2211.07804. Descargado de https://doi.org/10.48550/arXiv.2211.07804 doi: 10.48550/arXiv.2211.07804

Kora Venu, S., y Ravula, S. (2020). Evaluation of deep convolutional generative adversarial networks for data augmentation of chest x-ray images. *Future Internet*, *13*(1), 8. Descargado de https://doi.org/10.3390/fi13010008

Maccagnan, G. C., Schmith, J., Santos, M., y de Figueiredo, R. M. (2023). Toolbox for vessel x-ray angiography images simulation. En *Anais do xxiii simpósio brasileiro de computação aplicada à saúde* (pp. 59–70). Descargado de https://doi.org/10.5753/sbcas.2023.229439 doi: 10.5753/sbcas.2023.229439

Mayo Clinic. (2022). *Enfermedad de las arterias coronarias.* https://www.mayoclinic.org/es/diseases-conditions/coronary-artery-disease/symptoms-causes/syc-20350613.

Mayo Clinic. (2024). *Angiografía coronaria.* https://www.mayoclinic.org/es/tests-procedures/coronary-angiogram/about/pac-20384904.

Mohammed, S. S., y Clarke, H. G. (2024). Conditional image-to-image translation generative adversarial network (cgan) for fabric defect data augmentation. *Neural Computing and Applications*. Descargado de https://doi.org/10.1007/s00521-024-10179-1 doi: 10.1007/s00521-024-10179-1

Montazerolghaem, M., Sun, Y., Sasso, G., y Haworth, A. (2023). U-net architecture for prostate segmentation: the impact of loss function on system performance. *Bioengineering*, *10*(4), 412. Descargado de https://doi.org/10.3390/bioengineering10040412

Mumuni, A., y Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array*, *16*, 100-258. Descargado de https://doi.org/10.1016/j.array.2022.100258 doi: 10.1016/j.array.2022.100258

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., ... Mian, A. (2023). A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*. Descargado de https://doi.org/10.48550/arXiv.2307.06435

Ng, M. F., y Hargreaves, C. A. (2023). Generative adversarial networks for the synthesis of chest x-ray images. *Engineering Proceedings*, *31*(1), 84. Descargado de https://doi.org/10.3390/ASEC2022-13954

Nguyen, N. H. (2021). U-net based skeletonization and bag of tricks. En *Proceedings of the ieee/cvf international conference on computer vision* (pp. 2105–2109). Descargado de https://doi.org/10.1109/ICCVW54120.2021.00238 doi: 10.1109/ICCVW54120.2021.00238

Rahman, M. A., y Wang, Y. (2016). Optimizing intersection-over-union in deep neural networks for image segmentation. En *International symposium on visual computing* (pp. 234–244). Descargado de https://doi.org/10.1007/978-3-319-50835-1_22 doi: 10.1007/978-3-319-50835-1_22

Ramos, J. S. (2025). *Repositorio del proyecto de tesis.* Descargado de https://github.com/Saldor11/ModeloGenerativoICA

Ronneberger, O., Fischer, P., y Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. En *Medical image computing and computer-assisted intervention–miccai 2015: 18th international conference, munich, germany, october 5-9, 2015, proceedings, part iii 18* (pp. 234–241). Descargado de https://doi.org/10.1007/978-3-319-24574-4_28 doi: 10.1007/978-3-319-24574-4_28

Secretaría de Salud. (2022). *Cada año, 220 mil personas fallecen debido a enfermedades del corazón.* https://www.gob.mx/salud/prensa/490-cada-ano-220-mil-personas-fallecen-debido-a-enfermedades-del-corazon. (Ultimo acceso Enero 2024)

Shi, J., y cols. (1994). Good features to track. En *1994 proceedings of ieee conference on computer vision and pattern recognition* (pp. 593–600). Descargado de https://doi.org/10.1109/CVPR.1994.323794 doi: 10.1109/CVPR.1994.323794

Shorten, C., y Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, *6*(1), 1–48. Descargado de https://doi.org/10.1186/s40537-019-0197-0

Simard, P. Y., Steinkraus, D., Platt, J. C., y cols. (2003). Best practices for convolutional neural networks applied to visual document analysis. En *Proceedings of the international conference on document analysis and recognition (icdar)* (Vol. 3). Descargado de https://doi.org/10.1109/ICDAR.2003.1227801 doi: 10.1109/ICDAR.2003.1227801

Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., y Jorge Cardoso, M. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. En *Deep learning in medical image analysis and multimodal learning for clinical decision support: Third international workshop, dlmia 2017, and 7th international workshop, ml-cds 2017,*

held in conjunction with miccai 2017, québec city, qc, canada, september 14, proceedings 3 (pp. 240–248). Descargado de https://doi.org/10.1007/978-3-319-67558-9_28 doi: 10.1007/978-3-319-67558-9_28

Wolterink, J. M., Leiner, T., y Isgum, I. (2018). Blood vessel geometry synthesis using generative adversarial networks. *arXiv preprint arXiv:1804.04381*. Descargado de https://doi.org/10.48550/arXiv.1804.04381

Woo, S., Park, J., Lee, J.-Y., y Kweon, I. S. (2018). Cbam: Convolutional block attention module. En *Proceedings of the european conference on computer vision (eccv)* (pp. 3–19). Descargado de https://doi.org/10.48550/arXiv.1807.06521 doi: 10.48550/arXiv.1807.06521

Xu, J., Li, Z., Du, B., Zhang, M., y Liu, J. (2020). Reluplex made more practical: Leaky relu. En *2020 ieee symposium on computers and communications (iscc)* (pp. 1–7). Descargado de https://doi.org/10.1109/ISCC50000.2020.9219587 doi: 10.1109/ISCC50000.2020.9219587

Zhao, C., Vij, A., Malhotra, S., Tang, J., Tang, H., Pienta, D., ... Zhou, W. (2021). Automatic extraction and stenosis evaluation of coronary arteries in invasive coronary angiograms. *Computers in biology and medicine*, *136*, 104667. Descargado de https://doi.org/10.1016/j.compbiomed.2021.104667

Zhou, S. K., Greenspan, H., Davatzikos, C., Duncan, J. S., Van Ginneken, B., Madabhushi, A., ... Summers, R. M. (2021). A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proceedings of the IEEE*, *109*(5), 820–838. Descargado de https://doi.org/10.1109/JPROC.2021.3054390