



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación
y Desarrollo Tecnológico

Tesis de Maestría

Diseño de filtros fraccionarios de orden complejo,
constante, y variable aplicados a la reconstrucción de
imágenes médicas.

presentada por

Ing. Arturo García Morelos

como requisito para la obtención del grado de

**Maestro en Ciencias en Ingeniería
Electrónica**

Director de tesis

Dr. José Francisco Gómez Aguilar

Codirector de tesis

Dr. Jesús Emmanuel Solís Pérez

Cuernavaca, Morelos, México. Septiembre de 2024.



Cuernavaca, Mor.,

26/agosto/2024

No. de Oficio:

DIE/132/2024

Asunto:

Aceptación de documentos de tesis

DR. CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del C. ARTURO GARCÍA MORELOS, con número de control M22CE060 de la Maestría en Ciencias en Ingeniería Electrónica, le informamos que hemos revisado el trabajo de tesis profesional titulado "Diseño de Filtros Fraccionarios de Orden Complejo Constante y Variable Aplicados a la Reconstrucción de Imágenes Médicas", y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS

Dr. José Francisco Gómez Aguilar
Doctor en Física
Cédula profesional 09124781

CODIRECTOR DE TESIS

Dr. Jesús Emmanuel Solís Pérez
Doctor en Ciencias en Ingeniería
Electrónica
Cédula profesional 12057664

REVISOR 1

Dr. Víctor Manuel Alvarado Martínez
Doctor en Ciencias en Ingeniería
Electrónica
Cédula profesional 8031070

REVISOR 2

Dra. Guadalupe López López
Doctora en Ciencias en Ingeniería
Química
Cédula profesional 7980045

C.p.: M.T. María Elena Gómez Torres, Jefa del Departamento de Servicios Escolares
Estudiante
JCM/kmq





Cuernavaca, Mor.,
No. De Oficio:
Asunto:

26/agosto/2024
SAC/245/2024
Autorización de
impresión de tesis

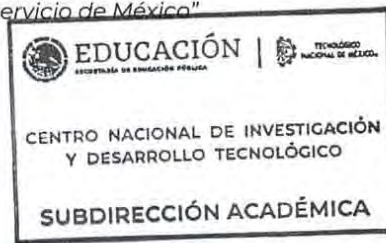
**ARTURO GARCÍA MORELOS
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS
EN INGENIERÍA ELECTRÓNICA
P R E S E N T E**

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **“Diseño de Filtros Fraccionarios de Orden Complejo Constante y Variable Aplicados a la Reconstrucción de Imágenes Médicas”**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE

*Excelencia en Educación Tecnológica®
“Conocimiento y tecnología al servicio de México”*



**CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO**

C. c. p. Departamento de Ingeniería Electrónica
Departamento de Servicios Escolares

CMAZ/lmz



Agradecimientos

Quisiera manifestar mi más sincero agradecimiento a mi querido padre, **Juventino García Cruz**, quien ya no está físicamente con nosotros pero cuyo amor y enseñanzas siguen guiándome cada día. Su ejemplo de perseverancia, bondad y sabiduría ha sido fundamental en mi vida. Agradezco su sacrificio y dedicación para brindarme las mejores oportunidades y por haber sido un modelo de integridad y fortaleza. Que su memoria perdure siempre en mi corazón y en mis acciones. Este logro y cualquier otro éxito que alcance en la vida son en gran parte gracias a él. A ti, papá, te dedico este momento y te llevo conmigo en cada paso que doy.

También deseo manifestar mi más sincero agradecimiento a mi madre, cuya constante dedicación, amor y apoyo han sido cruciales para la finalización de esta tesis. Su sabiduría y aliento constante me han inspirado a superar los desafíos y a perseverar en este arduo camino. Sin su guía y sacrificio, este logro no habría sido posible. Esta tesis está dedicada a ella, en reconocimiento a todo lo que ha hecho por mí.

Quiero manifestar mi más profundo agradecimiento a todas las personas que me han brindado apoyo y compañía durante este trayecto académico.

En primer lugar, expreso mi más sincero agradecimiento a mi director de tesis, el Dr. José Francisco Gómez Aguilar, por su valiosa orientación, paciencia y apoyo constante. Su experiencia y compromiso han sido cruciales para el progreso de este trabajo.

A mi codirector, el Dr. Jesus Emmanuel Solis Perez, le debo también un enorme agradecimiento por su valiosa ayuda y por sus constantes aportaciones que enriquecieron este proyecto.

Agradezco sinceramente a los revisores de mi tesis, la Dra. Ma. Guadalupe López López y al Dr. Víctor Manuel Alvarado Martínez, por sus detalladas revisiones, sus valiosos comentarios y sugerencias que han contribuido significativamente a la mejora de este trabajo.

Al M. en C. Jonathan Diego Pereyra Guzmán, por su colaboración y por compartir su conocimiento y tiempo conmigo. Su ayuda fue fundamental para superar muchos de los desafíos que encontré en el camino.

A mis amigos, Ana Seli, Xopo, Vic, Inri, Cheque, mi tocayo, Elker y Jose Luis Mata, gracias por su amistad, entendimiento y apoyo continuo. Sus palabras de ánimo y su compañía hicieron este viaje mucho más llevadero.

Agradezco a todos aquellos que, de alguna manera, han contribuido a la culminación de esta tesis. A cada uno de ustedes, mi más sincero agradecimiento.

Al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) por el apoyo financiero que hizo posible la realización y finalización de mis estudios de maestría.

Finalmente, me gustaría expresar mi agradecimiento al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) por ofrecerme la oportunidad de estudiar y prepararme, así como por proporcionarme un entorno de trabajo cómodo y una atención excelente en todo momento.

Resumen

La detección de bordes con los métodos tradicionales como los operadores de Sobel, Prewitt y Canny, aunque efectivos en muchos casos, presentan limitaciones en situaciones que requieren mayor precisión y adaptabilidad a diferentes escalas y texturas de la imagen. En este contexto, surge la necesidad de investigar nuevas técnicas capaces de superar estas limitaciones. Las derivadas fraccionarias han demostrado ser herramientas poderosas para modelar fenómenos complejos debido a su capacidad para capturar más información sobre la estructura de la señal. Sin embargo, su aplicación en la detección de bordes no ha sido suficientemente explorada, especialmente en el caso de órdenes fraccionarios y variables. Esta investigación se centra en el desarrollo de una metodología para la detección de bordes mediante el uso de operadores fraccionarios de orden constante y variable, aplicados al diseño de máscaras diferenciales. Esta metodología posee el potencial para proporcionar una mayor precisión y flexibilidad en la detección de bordes, superando las limitaciones de los métodos convencionales. La solución a este problema tiene implicaciones significativas para el campo del procesamiento de imágenes, proporcionando nuevas herramientas que pueden mejorar la calidad y precisión en aplicaciones críticas. La hipótesis central es que el uso de derivadas fraccionarias M-truncadas, tanto de orden constante como variable, en la generalización del Kernel Gaussiano y el gradiente de una imagen, mejorará significativamente la exactitud y efectividad en la identificación de curvas y bordes en imágenes digitales médicas.

Abstract

Edge detection with traditional methods such as Sobel, Prewitt, and Canny operators, while effective in many cases, has limitations in situations requiring greater precision and adaptability to different scales and textures of the image. In this context, there is a need to explore new techniques that can overcome these limitations. Fractional derivatives have proven to be powerful tools for modeling complex phenomena due to their ability to capture more information about the signal's structure. However, their application in edge detection has not been sufficiently explored, especially regarding fractional and variable orders.

This research focuses on developing a methodology for edge detection using fractional operators of constant and variable order, applied to the design of differential masks. This methodology has the potential to offer greater precision and flexibility in edge detection, surpassing the limitations of conventional methods.

Solving this problem has significant implications for the field of image processing, providing new tools that can enhance quality and accuracy in critical applications. The central hypothesis is that the use of M-truncated fractional derivatives, both of constant and variable order, in the generalization of the Gaussian Kernel and the gradient of an image, will significantly improve accuracy and effectiveness in identifying curves and edges in digital medical images.

Índice

Índice	IV
Índice de tablas	VI
Índice de figuras	VII
1. Introducción	1
1.1. Antecedentes	2
1.2. Estudio del estado del arte	2
1.3. Planteamiento del problema	6
1.4. Hipótesis	7
1.5. Objetivos	7
1.5.1. Objetivo general	7
1.5.2. Objetivos específicos	7
1.6. Estructura del documento	7
2. Preliminares matemáticos	9
2.1. Introducción	9
2.2. Gradiente de una imagen	9
2.3. Convolución	11
2.4. Detectores de Bordes	11
2.4.1. Funcionamiento de un Detector de Bordes	11
2.4.2. Ejemplos de Detectores de Bordes	12
2.4.3. Aplicaciones de los Detectores de Bordes	14
2.5. Filtro de Kernel Gaussiano	15
2.6. Cálculo fraccionario	15
2.6.1. La función Gamma (Γ)	16
2.6.2. Definición de la derivada fraccionaria de Riemann-Liouville	17
2.7. Derivada conformable	17
2.8. Derivada \mathcal{M} -truncada fraccionaria	18
2.9. Detector de bordes generalizado	19
2.10. Matriz Hessiana	19
2.11. Índices de desempeño en el procesamiento de imágenes	21
2.12. Función objetivo.	24

2.12.1. Definición y Propósito	24
3. Gradiente Gaussiano \mathcal{M}-truncado fraccionario	25
3.1. Generalización del kernel Gaussiano vía \mathcal{M} -truncada fraccionaria	25
3.2. Gradiente Gaussiano \mathcal{M} -truncado fraccionario de orden constante	27
3.3. Gradiente Gaussiano \mathcal{M} -truncado fraccionario de orden variable	28
4. Gradiente Gaussiano \mathcal{M}-truncado fraccionario aplicado a imágenes digitales	30
4.1. Gradiente Gaussiano \mathcal{M} -truncado fraccionario de orden constante aplicado a imágenes médicas	30
4.1.1. Función objetivo	30
4.1.2. Resultados	30
4.1.3. Análisis de resultados	42
4.2. Gradiente Gaussiano \mathcal{M} -truncado fraccionario de orden variable aplicado a imágenes médicas	43
4.2.1. Función objetivo	43
4.2.2. Resultados	43
4.2.3. Análisis de resultados	61
4.3. Matriz Hessiana	62
4.3.1. Función objetivo	62
4.3.2. Resultados	62
4.3.3. Análisis de resultados	71
5. Conclusiones	72
5.1. Trabajos Futuros	72
Bibliografía	74
Anexos	78
A. Pseudocódigos	79
B. Productos obtenidos	89

Índice de tablas

4.1. Resultados de las métricas de la imagen “Lung abscess and pleural empyema”	32
4.2. Resultados de las métricas de la imagen “covid2”	34
4.3. Resultados de las métricas de la imagen “covid3”	36
4.4. Resultados de las métricas de la imagen “covid4”	38
4.5. Resultados de las métricas de la imagen “covid5”	40
4.6. Resultados de las métricas de la imagen “covid6”	42
4.7. Resultados de las métricas de la imagen “Lung abscess and pleural empyema” procesada con funciones suaves	45
4.8. Resultados de las métricas de la imagen “covid2” procesada con funciones suaves	49
4.9. Resultados de las métricas de la imagen “covid3” procesada con funciones suaves	52
4.10. Resultados de las métricas de la imagen “covid4” procesada con funciones suaves	55
4.11. Resultados de las métricas de la imagen “covid5” procesada con funciones suaves	58
4.12. Resultados de las métricas de la imagen “covid6” procesada con funciones suaves	61
4.13. Comparación de resultados	61

Índice de figuras

3.1.	Función de distribución gaussiana y el kernel gaussiano.	25
3.2.	respuestas a la ecuación diferencial fraccionaria con $\mu = 0$ y $\sigma = 0.3$	27
3.3.	Metodología para el procesamiento de imagenes usando la derivada- \mathcal{M} truncada fraccionaria.	28
3.4.	Funciones suaves. (a) tangente hiperbolica, (b) Sigmoidal.	29
4.1.	(a) Imagen “Lung abscess and pleural empyema”. (b) Imagen procesada con el detector de bordes generalizado.	31
4.2.	Detectores de bordes clásicos a)Prewitt, b)Sobel, c)Canny	31
4.3.	(a) suma de la imagen “Lung abscess and pleural empyema” con el detector de bordes generalizado. (b) resta de la imagen “Lung abscess and pleural empyema” con el detector de bordes generalizado.	32
4.4.	(a) Imagen “covid2”. (b) Imagen procesada con el detector de bordes generalizado. . . .	33
4.5.	Detectores de bordes clásicos a)Prewitt, b)Sobel, c)Canny	33
4.6.	Resultados al sumar y restar la imagen “covid2” con el detector de bordes generalizado	34
4.7.	(a) Imagen “covid3”. (b) Imagen procesada con el detector de bordes generalizado. . . .	35
4.8.	Detectores de bordes clásicos a)Prewitt, b)Sobel, c)Canny	35
4.9.	Resultados al sumar y restar la imagen “covid3” con el detector de bordes generalizado	36
4.10.	(a) Imagen “covid4”. (b) Imagen procesada con el detector de bordes generalizado. . . .	37
4.11.	Detectores de bordes clásicos a)Prewitt, b)Sobel, c)Canny	37
4.12.	Resultados al sumar y restar la imagen “covid4” con el detector de bordes generalizado	38
4.13.	(a) Imagen “covid5”. (b) Imagen procesada con el detector de bordes generalizado. . . .	39
4.14.	Detectores de bordes clásicos a)Prewitt, b)Sobel, c)Canny	39
4.15.	Resultados al sumar y restar la imagen “covid5” con el detector de bordes generalizado	40
4.16.	(a) Imagen “covid6”. (b) Imagen procesada con el detector de bordes generalizado. . . .	41
4.17.	Detectores de bordes clásicos a)Prewitt, b)Sobel, c)Canny	41
4.18.	Resultados al sumar y restar la imagen “covid6” con el detector de bordes generalizado	42
4.19.	(a) Imagen “Lung abscess and pleural empyema”. (b) Imagen “Lung abscess and pleural empyema” procesada con el detector de bordes con parámetros variables.	43
4.20.	(a) suma de la imagen “Lung abscess and pleural empyema” con la imagen procesada con el detector con parámetros variables. (b) resta de la imagen “Lung abscess and pleural empyema” con la imagen procesada con el detector de bordes con parámetros variables.	44
4.21.	(a) Imagen “Lung abscess and pleural empyema”. (b) Imagen “Lung abscess and pleural empyema” procesada con el detector de bordes con parámetros variables.	44

4.22. (a) suma de la imagen “Lung abscess and pleural empyema” con el detector de bordes con parámetros variables. (b) resta de la imagen “Lung abscess and pleural empyema” con el detector de bordes con parámetros variables.	45
4.23. (a) Imagen “covid2”. (b) Imagen “covid2” procesada con el detector de bordes con parámetros variables.	46
4.24. (a) suma de la imagen “covid2” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid2” con el detector de bordes con parámetros variables.	47
4.25. (a) Imagen “covid2”. (b) Imagen “covid2” procesada con el detector de bordes con parámetros variables.	48
4.26. (a) suma de la imagen “covid2” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid2” con el detector de bordes con parámetros variables.	49
4.27. (a) suma de la imagen “covid3” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid3” con el detector de bordes con parámetros variables.	50
4.28. (a) suma de la imagen “covid3” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid3” con el detector de bordes con parámetros variables.	50
4.29. (a) suma de la imagen “covid3” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid3” con el detector de bordes con parámetros variables.	51
4.30. (a) suma de la imagen “covid3” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid3” con el detector de bordes con parámetros variables.	51
4.31. (a) Imagen “covid4”. (b) Imagen “covid4” procesada con el detector de bordes con parámetros variables.	52
4.32. (a) suma de la imagen “covid4” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid4” con el detector de bordes con parámetros variables.	53
4.33. (a) suma de la imagen “covid4” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid4” con el detector de bordes con parámetros variables.	54
4.34. (a) suma de la imagen “covid4” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid4” con el detector de bordes con parámetros variables.	55
4.35. (a) Imagen “covid5”. (b) Imagen “covid5” procesada con el detector de bordes con parámetros variables.	56
4.36. (a) suma de la imagen “covid5” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid5” con el detector de bordes con parámetros variables.	56
4.37. (a) Imagen “covid5”. (b) Imagen “covid4” procesada con el detector de bordes con parámetros variables.	57
4.38. (a) suma de la imagen “covid5” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid5” con el detector de bordes con parámetros variables.	58
4.39. (a) Imagen “covid6”. (b) Imagen “covid6” procesada con el detector de bordes con parámetros variables.	59
4.40. (a) suma de la imagen “covid6” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid6” con el detector de bordes con parámetros variables.	59
4.41. (a) suma de la imagen “covid6” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid6” con el detector de bordes con parámetros variables.	60
4.42. (a) suma de la imagen “covid6” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid6” con el detector de bordes con parámetros variables.	60

4.43. (a) Imagen “Lung abscess and pleural empyema”. (b) Imagen “Lung abscess and pleural empyema” procesada con la curvatura principal.	63
4.44. (a) suma de la imagen “Lung abscess and pleural empyema” con el detector de bordes clásico y la curvatura principal. (b) resta de la imagen “Lung abscess and pleural empyema” con el detector de bordes clásico y la curvatura principal.	63
4.45. (a) Imagen “covid2”. (b) Imagen “covid2” procesada con la curvatura principal.	64
4.46. (a) suma de la imagen “covid2” con el detector de bordes clásico y la curvatura principal. (b) resta de la imagen “covid2” con el detector de bordes clásico y la curvatura principal.	65
4.47. (a) Imagen “covid3”. (b) Imagen “covid3” procesada con la curvatura principal.	65
4.48. (a) suma de la imagen “covid3” con el detector de bordes clásico y la curvatura principal. (b) resta de la imagen “covid3” con el detector de bordes clásico y la curvatura principal.	66
4.49. (a) Imagen “covid4”. (b) Imagen “covid4” procesada con la curvatura principal.	67
4.50. (a) suma de la imagen “covid4” con el detector de bordes clásico y la curvatura principal. (b) resta de la imagen “covid4” con el detector de bordes clásico y la curvatura principal.	68
4.51. (a) Imagen “covid5”. (b) Imagen “covid5” procesada con la curvatura principal.	69
4.52. (a) suma de la imagen “covid5” con el detector de bordes clásico y la curvatura principal. (b) resta de la imagen “covid5” con el detector de bordes clásico y la curvatura principal.	69
4.53. (a) suma de la imagen original con el detector de bordes generalizado. (b) resta de la imagen original con el detector de bordes generalizado.	70
4.54. (a) suma de la imagen “covid6” con el detector de bordes clásico y la curvatura principal. (b) resta de la imagen “covid6” con el detector de bordes clásico y la curvatura principal.	71

Capítulo 1

Introducción

Las imágenes digitales están formadas por una serie de puntos llamados píxeles, cuyo valor varía según la resolución de la imagen. Para una imagen en escala de grises, se utiliza una resolución de 8 bits, lo que significa que cada píxel puede tener un valor entre 0 y 255. Cada número dentro de este rango representa un nivel de intensidad luminosa, donde 0 corresponde al negro absoluto y el 255 al blanco absoluto, permitiendo así una amplia gama de tonos de gris en la imagen [1].

Actualmente, la manipulación de imágenes tiene infinidad de aplicaciones en diferentes campos. Entre ellas se encuentran la visión artificial, la robótica móvil, la identificación de patrones, la recuperación de imágenes, el análisis de imágenes médicas, la interpretación de información visual y la detección a distancia. Estas aplicaciones van desde la mejora y evaluación de imágenes digitales hasta la extracción de información valiosa de datos visuales, teniendo un impacto significativo en varios campos de estudio. La identificación de bordes es una técnica clave en numerosas aplicaciones de manipulación de imágenes, como el reconocimiento de objetos, la división de imágenes en regiones distintas y la percepción visual por medios computacionales. [2–5].

El objetivo principal del procesamiento de imágenes es optimizar la calidad y las propiedades de las imágenes para facilitar su interpretación. Una de las técnicas esenciales en este ámbito es la detección de bordes, la cual identifica los puntos en una imagen digital donde se produce un cambio abrupto en el brillo o la intensidad luminosa. Al identificar estos cambios, se pueden destacar las regiones de interés, como los contornos y las transiciones importantes, permitiendo la obtención de datos importantes y la interpretación visual de la imagen obtenida. Los bordes son regiones en una imagen donde ocurre un cambio notable en la intensidad de la luz, que puede detectarse analizando píxeles con una gran variación en la escala de grises. Utilizando diversos algoritmos, es posible identificar y resaltar los bordes, contornos o esquinas en la imagen procesada. Estos algoritmos se basan en las diferencias de intensidad entre los píxeles vecinos para localizar los puntos donde ocurren cambios bruscos en la imagen. La identificación de bordes es un método esencial en el análisis de imágenes y se emplea en múltiples usos, como la división de objetos, la identificación de formas y la extracción de características [2, 6].

Los métodos para la identificación de bordes se fundamentan en el uso de operadores diferenciales de orden entero. Entre los operadores más destacados están Prewitt, Sobel, Roberts, el Laplaciano de Gauss (LoG) y el algoritmo de Canny. Estos operadores están diseñados para identificar las variaciones abruptas en la luminosidad de una imagen y resaltar los bordes presentes en ella. Cada operador tiene sus propias características y ventajas, y la elección entre uno u otro varía según el tipo de imagen y de la aplicación específica. El algoritmo de Canny es especialmente valorado debido a su habilidad

para identificar bordes con alta precisión y por su eficacia en la reducción de ruido [7–9].

Los modelos de orden fraccionario ofrecen una mayor precisión en la descripción de ciertos fenómenos físicos en relación con los modelos basados en derivadas de orden completo. Estos modelos fraccionarios permiten capturar características y comportamientos complejos que no pueden ser totalmente capturados por modelos basados en derivadas de orden entero. Al introducir un componente fraccionario, se puede considerar la dependencia a largo plazo, la memoria no local y otros efectos no lineales que pueden ser críticos en muchos sistemas físicos y fenómenos naturales. El uso de modelos fraccionarios ha demostrado ser útil en campos como la dinámica de fluidos, la física de materiales, la electrónica y otras disciplinas donde la descripción precisa de los fenómenos es esencial [10].

Para las imágenes en escala de grises, los niveles de gris entre píxeles adyacentes suelen estar bastante correlacionados. Aprovechando esta característica, es posible aplicar derivadas fraccionarias para resaltar los patrones de textura en los niveles de gris contiguos. Al utilizar derivadas fraccionarias, se logra mejorar la precisión y la preservación de las propiedades de las imágenes, incluso cuando hay variaciones sutiles en las intensidades de gris. Esto resulta en una mayor fidelidad y calidad visual de la imagen procesada, ya que se destacan los detalles de textura y se conservan las características originales de la imagen. Este método es particularmente eficaz en aplicaciones donde se requiere un análisis detallado de las texturas presentes en las imágenes, como en el ámbito de la visión por computadora y el análisis de imágenes [11].

1.1. Antecedentes

En el Centro Nacional de Investigación y Desarrollo Tecnológico (TecNM/CENIDET) se han llevado a cabo investigaciones y trabajos en el campo del análisis de imágenes que involucran el cálculo fraccionario. Por ejemplo, Pereyra-Guzmán [12] en 2019 propuso un método para identificar bordes utilizando un gradiente Gaussiano conformable. Esta metodología fue aplicada en imágenes médicas mostrando mejoras en la detección de bordes incluso en presencia de ruido tipo sal-pimienta y ruido Gaussiano.

Solís-Pérez [13] en 2020 propuso un filtro de kernel Gaussiano fraccionario con kernel no singular para la obtención de un gradiente. Este gradiente fue utilizado para obtener la matriz Hessiana. Esta metodología fue aplicada para identificar vasos sanguíneos en imágenes de retinopatía [14]. También propuso un gradiente fundamentado en la derivada de Caputo-Fabrizio para identificar bordes [15].

Lavín-Delgado et al. [16] propuso una máscara de orden fraccionario identificar bordes utilizando la derivada fraccionaria conformable. Los resultados de las simulaciones mostraron que esta metodología resaltaba los bordes suaves, así como texturas. Estas propuestas demostraron ser robustas y precisas en las tareas designadas. Por ello, se busca avanzar con metodologías de procesamiento en imágenes biomédicas donde existen dificultades en las condiciones de adquisición (ruido, baja resolución, difuminación) o en las características de los objetos objetivo (objetivos múltiples, superpuestos o transparentes, así como texturas poco claras).

1.2. Estudio del estado del arte

El estado del arte para desarrollar esta tesis se divide en los siguientes temas:

- Algoritmos para identificar bordes.

- Derivadas fraccionarias.

Aunque la mayoría de las derivadas fraccionarias son lineales, no satisfacen algunas reglas matemáticas esenciales, como la regla de la cadena y el teorema del valor medio. Por lo tanto, en la literatura se han propuesto nuevas definiciones de derivadas que se centran en un enfoque local [17].

De acuerdo con varios autores, los operadores fraccionarios se pueden dividir en dos grupos: derivadas fraccionarias con núcleo singular y no singular, y derivadas locales o conformables. La primera categoría abarca los operadores que pueden captar el impacto de la memoria en los sistemas físicos, mientras que la segunda engloba aquellos que reflejan los efectos fractales del entorno.

En 1965, Roberts introdujo un método para estimar el gradiente de una imagen utilizando diferenciación discreta. Esta técnica se basa en sumar los cuadrados de las diferencias entre los píxeles adyacentes en diagonal. [18].

En 1968, Sobel `epsobel19683x3` presentó un operador que facilitaba la medición del gradiente espacial bidimensional en una imagen. Este operador se usó para determinar la magnitud absoluta del gradiente. Para ello, se emplearon dos máscaras de convolución de tamaño 3×3 , una para calcular el gradiente en la dirección x y otra para el gradiente en la dirección y . Una ventaja notable de este detector es su simplicidad de implementación en comparación con otros métodos de identificación de bordes.

En 1970, Prewitt [19] presentó un operador de gradiente que utiliza una máscara de 3×3 . Este operador mide la variación en los niveles de gris entre los píxeles adyacentes en las direcciones horizontal y vertical.

En 1970, Prewitt [19] propuso un operador de gradiente basado en una máscara de tamaño 3×3 . Este operador se utiliza para medir la diferencia entre los niveles de gris de los píxeles adyacentes en las direcciones horizontal y vertical. Una de las principales ventajas de este método es su tiempo de procesamiento reducido en comparación con el operador de Sobel.

En 1975, Roesser [20] intentó desarrollar un modelo lineal en el espacio de estado para el procesamiento lineal de imágenes en tiempo discreto. Este modelo fue expandido desde un modelo de baja dimensionalidad a uno en un espacio de estados bidimensional. La expansión incluye la ampliación de conceptos clave, como la matriz de transición de estados, el teorema de Cayley–Hamilton, así como la observabilidad y la controlabilidad.

En 1987, Canny [21] presentó un método computacional para la detección de bordes que implica calcular la magnitud y orientación del gradiente mediante aproximaciones de diferencias finitas para las derivadas parciales. En su estudio, el autor generalizó y verificó este detector de bordes al aplicar y evaluar su rendimiento en imágenes con diferentes características, midiendo su eficacia a través de la relación señal-ruido. Esto permitió evaluar la habilidad del detector para mantener la calidad de la imagen en presencia de ruido y su desempeño en diversas aplicaciones.

En 1993, Deng [22] presentó un método que utiliza un filtro Gaussiano adaptativo para identificar bordes. Los resultados obtenidos en sus investigaciones demostraron una mejora significativa en la identificación de bordes, particularmente en zonas donde estos se intersectan. Este método adaptativo facilitó una detección más precisa de los bordes, lo que mejoró la calidad del proceso en comparación con las técnicas disponibles en ese momento..

En 1998, Russo [23] introdujo un método innovador para la identificación de bordes en imágenes con ruido. Su enfoque utiliza un operador basado en lógica difusa. Los resultados experimentales mostraron que esta técnica era eficaz y superaba a otras metodologías en cuanto a sensibilidad al ruido y la detección de detalles o texturas en las imágenes.

Ese mismo año, Gudmundsson [24] sugirió la aplicación de algoritmos genéticos para identificar bordes en imágenes médicas. Este enfoque demostró ser muy eficiente en cuanto al costo computacional, facilitando la localización precisa de bordes finos y continuos en las imágenes. Los datos obtenidos en las pruebas mostraron una exitosa recuperación de la mayoría de las características clave, incluyendo los bordes finos, lo que subrayó la efectividad y el valor de este método para el análisis de imágenes médicas.

En 2002, Sharifi [25] propuso una categorización de los algoritmos más empleados para la detección de bordes. dicha categorización se organiza en cinco categorías: detectores de bordes basados en el gradiente (primera derivada o clásica), detección de cruces por cero (segunda derivada), Laplaciano del Gaussiano, detectores de bordes Gaussianos y detectores de bordes en imágenes a color. Los datos de su estudio mostraron que, en condiciones de ruido, los algoritmos de Canny, Laplaciano del Gaussiano y Sobel fueron los que ofrecieron mejor rendimiento en cuanto a precisión y robustez en la detección de bordes.

En 2003, S. H. Yun [26] realizó una demostración de imágenes ópticas de alta velocidad, alta sensibilidad y alta resolución utilizando la técnica de interferometría de dominio de frecuencia óptica con un láser de barrido de longitud de onda rápidamente sintonizado. Si bien esta técnica mostró resultados prometedores en imágenes ópticas, su implementación en aplicaciones de imágenes biomédicas aún no ha sido ampliamente probada y evaluada en detalle. Se requiere más investigación y pruebas para determinar el impacto y las limitaciones de esta técnica en el campo de la imágenes biomédicas.

En el año 2005, Nezamabadi-pour [27] desarrolló un innovador método de detección de bordes que se basa en un algoritmo inspirado en el comportamiento de hormigas reales. Este enfoque utiliza parámetros que fueron ajustados empíricamente para obtener los mejores resultados. Los experimentos realizados demostraron un excelente rendimiento del algoritmo, incluso en imágenes con una alta variación de niveles de grises. Esto evidencia la eficacia y la resistencia del método sugerido para identificar bordes con precisión en diversas condiciones de imágenes.

En 2009, Blas [28] realizó una presentación introductoria al cálculo fraccionario, explorando sus orígenes y los acontecimientos históricos que llevaron al desarrollo de esta área dentro del análisis matemático. Durante esta breve reseña histórica, el autor destacó diversos operadores fraccionarios, como la derivada fraccionaria de Riemann–Liouville y la derivada fraccionaria de Grünwald–Letnikov. Además, Blas proporcionó ejemplos de aplicaciones y resultados interesantes que han sido obtenidos utilizando estos operadores fraccionarios. Esta exposición resaltó el potencial y la importancia del cálculo fraccionario en diferentes campos de investigación.

En 2011, Gao [29] presentó una propuesta para extender la derivada de Grünwald–Letnikov en el ámbito del procesamiento de imágenes digitales. Esta extensión se utilizó en la creación de un algoritmo eficaz para la identificación de bordes. Las pruebas simuladas revelaron que este algoritmo ofrece resultados superiores en comparación con el método de Sobel, que es uno de los métodos tradicionales más utilizados. La generalización propuesta por Gao permitió capturar con mayor precisión los cambios de intensidad en las imágenes y mejorar la detección de bordes en diferentes escenarios.

En 2014, Tian [30] presentó una extensión del operador de Sobel clásico fundamentada en la derivada fraccionaria de Grünwald–Letnikov. Esta generalización fue aplicada específicamente en la identificación de características en imágenes de ultrasonido y resonancias magnéticas cerebrales. Los experimentos llevados a cabo mostraron que el operador propuesto genera efectos visuales de alta calidad y resultados

satisfactorios en la detección de características importantes en estas imágenes. Esto indica que el método propuesto por Tian es prometedor y posee potencial para ser utilizado en el análisis de imágenes.

En 2016, Yang [31] introdujo definiciones de operadores fraccionarios y diversos métodos de discretización para estos operadores. También detalló múltiples técnicas de procesamiento de imágenes fundamentadas en derivadas fraccionarias. Mediante resultados experimentales, Yang evidenció los beneficios de emplear una metodología fraccionaria en el procesamiento de imágenes. Estas ventajas incluyen una mejor captura de características sutiles, una mayor capacidad para preservar detalles importantes y una mayor robustez frente al ruido y otros artefactos presentes en las imágenes. Los hallazgos obtenidos avalan la utilidad y efectividad de los métodos de procesamiento de imágenes fundamentados en derivadas fraccionarias propuestos por Yang.

En 2017, Jyoti Dabas [32] presentó un análisis de mejora de imágenes médicas utilizando una metodología en análisis frecuencial. Este enfoque se centra en ajustar el contraste, reducción del ruido, afilado de bordes, corte de nivel de gris, interpolación y pseudo-coloración, todo ello aplicado a nivel de píxel. El enfoque se centra en el procesamiento individual de cada píxel de la imagen. Además, el alcance de esta técnica se extiende a la fusión de imágenes médicas de múltiples dominios, lo cual facilita la obtención de una representación más exhaustiva y minuciosa de la de la información médica. Los resultados obtenidos demuestran la eficacia de esta técnica en la mejora y análisis de imágenes médicas, brindando información más precisa y útil para el diagnóstico y tratamiento médico.

En 2018, Nandal [11] presentó una implementación para mejorar las propiedades de una imagen mediante el uso de máscaras diferenciales fundamentadas en el operador diferencial fraccionario de Riemann–Liouville. Este algoritmo facilita el ajuste del nivel de mejora del contraste en una imagen. Los resultados obtenidos mostraron una destacada precisión en la localización y detección de bordes más definidos en comparación con otros métodos utilizados. La implementación propuesta por Nandal proporciona una herramienta efectiva para realzar las características de las imágenes, mejorando su claridad y facilitando su análisis en diversas aplicaciones.

En 2018, Cruz-Duarte [33] presentó una contribución importante al campo del cálculo fraccionario al presentar una fórmula explícita para la derivada fraccionaria de la función Gaussiana. Dicha fórmula se basa en la definición de Caputo–Fabrizio y permite calcular de manera eficiente la derivada fraccionaria de la función Gaussiana en diversos contextos. Las expresiones obtenidas han demostrado ser de gran valor en el análisis de señales, imágenes y en la construcción de filtros y funciones de distribuciones de probabilidad. Esta contribución ha abierto nuevas oportunidades para el análisis y la modelización de fenómenos que exhiben comportamientos fraccionarios en diversas aplicaciones científicas y tecnológicas.

En 2019, Lavín-Delgado [15] llevó a cabo una notable extensión del operador de Sobel al considerar la derivada de orden fraccionario de Caputo–Fabrizio. Este enfoque innovador permitió desarrollar nuevas máscaras de orden fraccionario para identificar bordes en imágenes. El autor aplicó estas máscaras en la evaluación de imágenes médicas, particularmente en el estudio de condiciones como el cáncer de mama y las calcificaciones, entre otros. Los resultados obtenidos indicaron que la extensión del operador de Sobel empleando derivadas de orden fraccionario, sin núcleo singular, presentó un rendimiento superior en comparación con el operador clásico de Sobel. Esta contribución ha impulsado avances significativos en el análisis de imágenes médicas, abriendo nuevas posibilidades para la identificación y evaluación precisas de estructuras y características relevantes en la práctica clínica.

En 2021, Wenguan Wang [34] presentó un nuevo enfoque para la segmentación semántica basado en un paradigma de aprendizaje supervisado. Este enfoque se centra en el aprendizaje contrastante, donde

se guía el proceso de aprendizaje de las representaciones de los píxeles hacia una mayor discriminación de categorías de imágenes cruzadas. Esta estrategia mejora el rendimiento general de la segmentación semántica al proporcionar embeddings más distintivos y representativos. Además, se abordan desafíos adicionales, como el diseño de funciones de pérdida de aprendizaje, con el objetivo de optimizar aún más el proceso de segmentación y lograr resultados más precisos. Este enfoque innovador contribuye al avance de la segmentación semántica en el ámbito de la visión por computadora y tiene el potencial de mejorar las aplicaciones en diferentes áreas, como la identificación de objetos y el análisis de imágenes. En 2021, Xuan Gong [35] propuso un enfoque innovador utilizando el módulo de Gabor para la extracción de características altamente representativas. Este enfoque se basa en un método de aprendizaje de varias instancias que aborda las características deformables de Gabor. La Red de Características Deformables de Gabor integra filtros adaptativos de Gabor en convoluciones deformables, lo que permite capturar eficientemente las variaciones espaciales en objetos y extraer características discriminativas para múltiples categorías. Los experimentos realizados demuestran que esta arquitectura eficiente y compacta es capaz de proporcionar resultados robustos en presencia de ruido de tipo Gaussiano. Sin embargo, se encontró que la red es resistente al ruido de Gauss o ruido tipo Gaussiano, lo que indica que aún hay desafíos por superar en términos de su capacidad para manejar este tipo de perturbaciones en los datos. No obstante, este método constituye un progreso importante en el área de la extracción de características y ofrece perspectivas prometedoras para aplicaciones prácticas en la identificación de objetos y otras aplicaciones de visión por computadora.

1.3. Planteamiento del problema

La identificación de bordes es una etapa crucial en el análisis de imágenes, utilizada en diversas aplicaciones, que van desde la visión por computadora hasta la medicina. Los enfoques convencionales para la identificación de bordes, tales como los operadores de Sobel, Prewitt y Canny, han demostrado ser efectivos en muchos casos, pero presentan limitaciones en situaciones donde se requiere una mayor precisión y adaptabilidad a diferentes escalas y texturas de la imagen.

En este marco, se hace necesario el explorar nuevas técnicas que puedan superar estas limitaciones. Las derivadas fraccionarias han demostrado ser herramientas poderosas para modelar fenómenos complejos debido a su capacidad para capturar más información sobre la estructura de la señal. No obstante, el uso de derivadas fraccionarias en la identificación de bordes aún no ha sido lo suficientemente explorada, especialmente cuando se consideran órdenes fraccionarios complejos y variables.

Por lo tanto, el problema central que aborda esta investigación es la necesidad de crear un enfoque para la identificación de bordes que emplee operadores fraccionarios de orden constante y variable para la creación de máscaras diferenciales. Este enfoque podría proporcionar una mayor exactitud y flexibilidad en la identificación de bordes, superando las limitaciones de los métodos tradicionales. Específicamente, el estudio se centrará en:

- **Generalizar el Kernel Gaussiano:** Utilizar la derivada \mathcal{M} -truncada fraccionaria con tres órdenes diferentes para mejorar el procesamiento y la reconstrucción de imágenes.
- **Generalizar el gradiente de una imagen:** Aplicar la derivada \mathcal{M} -truncada fraccionaria con órdenes constantes y variables para la identificación de curvas y bordes en las imágenes.

- **Desarrollar algoritmos conformables y fraccionarios:** Desarrollar algoritmos especializados para identificar con precisión curvas y bordes aprovechando las características de las derivadas fraccionarias

La solución a este problema tiene implicaciones significativas para el campo del procesamiento de imágenes, proporcionando nuevas herramientas que pueden mejorar la calidad y precisión en aplicaciones críticas.

1.4. Hipótesis

El uso de derivadas fraccionarias \mathcal{M} -truncadas, tanto de orden constante como variable, en la generalización del Kernel Gaussiano y el gradiente de una imagen, mejorará significativamente la precisión y eficacia en la detección de curvas y bordes en imágenes digitales médicas.

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar una metodología de detección de bordes con base en operadores fraccionarios de orden complejo constante y variable para el diseño de máscaras diferenciales.

1.5.2. Objetivos específicos

- Generalizar el Kernel Gaussiano utilizando la derivada \mathcal{M} -truncada fraccionaria con 3 órdenes para procesar y reconstruir imágenes.
- Generalizar el gradiente de una imagen utilizando la derivada \mathcal{M} -truncada fraccionaria con 3 órdenes constantes y variables para la detección de curvas y bordes.
- Desarrollar los algoritmos conformables y fraccionarios que permitan la detección de curvas y bordes.

1.6. Estructura del documento

La estructura del documento se presenta de la siguiente forma:

- **Capítulo 1: Introducción**
Ofrece un panorama general del tema de investigación, abarcando la motivación detrás del estudio, los objetivos planteados y el planteamiento del problema que se aborda.
- **Capítulo 2: Preliminares Matemáticos**
Se proporcionan las bases matemáticas necesarias para comprender los conceptos avanzados que se discutirán en los capítulos posteriores. Se revisan conceptos clave como las derivadas fraccionarias y otros operadores matemáticos relevantes.

- **Capítulo 3: Gradiente Gaussiano \mathcal{M} -truncado Fraccionario**

Introduce la metodología del gradiente Gaussiano \mathcal{M} -truncado fraccionario. Se describen las propiedades y ventajas de utilizar este enfoque en el procesamiento de imágenes.

- **Capítulo 4: Gradiente Gaussiano \mathcal{M} -truncado Fraccionario Aplicado a Imágenes Digitales**

Presenta la aplicación práctica del gradiente Gaussiano \mathcal{M} -truncado fraccionario en el procesamiento de imágenes digitales. Se muestran ejemplos y resultados de experimentos que demuestran la eficacia de la metodología propuesta.

- **Capítulo 5: Conclusiones**

El documento concluye con un resumen de los hallazgos principales, la discusión de las implicaciones de los resultados obtenidos y posibles direcciones futuras para la investigación.

Capítulo 2

Preliminares matemáticos

2.1. Introducción

En esta sección se aborda el estudio de los fundamentos que rigen el sustento teórico del estudio de los filtros fraccionarios.

2.2. Gradiente de una imagen

El gradiente de una imagen es una medida vectorial que indica la dirección y la magnitud del cambio en la luminosidad en cada punto de la imagen. Se utiliza para resaltar las transiciones abruptas de intensidad, que generalmente corresponden a los bordes o límites entre objetos en la imagen.

El gradiente se calcula empleando métodos de derivadas parciales en el ámbito de la imagen. Por lo general, se emplean operadores Sobel, Prewitt o Roberts, que aplican máscaras o filtros a la imagen para determinar las derivadas en las direcciones horizontales y verticales.

En cada ubicación de la imagen, el gradiente se representa como un vector con componentes en las direcciones horizontal y vertical. La medida del vector gradiente refleja la intensidad del cambio de intensidad en ese punto y la orientación del vector señala la dirección en la que ocurre el mayor cambio. El gradiente de una imagen es crucial en numerosas aplicaciones de procesamiento de imágenes, como la identificación de bordes, el segmentado de objetos, la corrección de imágenes, entre otras técnicas de análisis. Facilita la identificación y el realce de las áreas relevantes en la imagen y ofrece información detallada para el posterior procesamiento y análisis de la imagen. Entonces, podemos definir el gradiente como sigue

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{d}{dx}f(x, y) \\ \frac{d}{dy}f(x, y) \end{bmatrix}, \quad (2.1)$$

donde, el vector G apunta en la dirección de la mayor variación en el punto por unidad de distancia, con una magnitud y dirección determinadas por

$$|G| = \sqrt{G_x^2 + G_y^2}, \quad (2.2)$$

$$\Phi(x, y) = \tan^{-1} \left(\frac{G_y}{G_x} \right). \quad (2.3)$$

Generalmente, se estima la magnitud del gradiente tomando en cuenta los valores absolutos

$$|G| \approx |G_x| + |G_y|. \quad (2.4)$$

Cualquiera de las ecuaciones (2.2), (2.3) y (2.4) puede utilizarse para obtener la magnitud del gradiente, ya que todas se relacionan con un umbral específico. Si el valor de la magnitud supera dicho umbral, se considera que el punto es un borde. Las derivadas de la ecuación (2.1) se basan en diferencias centrales, es decir,

$$G_x = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}, \quad (2.5)$$

$$G_y = \frac{f(y + \Delta y) - f(y - \Delta y)}{2\Delta y}, \quad (2.6)$$

donde, tratándose de una imagen digital, se asume que $\Delta x = \Delta y = 1$. En consecuencia, se obtiene

$$G_x = f(x + 1, y) - f(x - 1, y), \quad (2.7)$$

$$G_y = f(x, y + 1) - f(x, y - 1). \quad (2.8)$$

Esta es la manera más básica de calcular el gradiente en un punto. Para binarizar una imagen, se considera que

$$g(x, y) = \begin{cases} 1, & \text{si } G[f(x, y)] > T, \\ 0, & \text{si } G[f(x, y)] \leq T, \end{cases} \quad (2.9)$$

en la que T es un valor de umbral no negativo, solo se consideran importantes los píxeles de borde que superen el valor de T .

Tomando en cuenta una vecindad de píxeles adyacentes de 3×3 , las expresiones (2.7) y (2.8) se pueden representar en forma matricial como sigue

$$G_x = \begin{bmatrix} a_0 & a_1 & a_2 \\ Ca_7 & a_8 & Ca_3 \\ a_6 & a_5 & a_4 \end{bmatrix}, \quad (2.10)$$

$$G_y = \begin{bmatrix} a_0 & Ca_1 & a_2 \\ a_7 & a_8 & a_3 \\ a_6 & Ca_5 & a_4 \end{bmatrix}, \quad (2.11)$$

donde a_0 hasta a_8 son píxeles y C cambia según la máscara empleada o puede ser una constante escalar.

$$G_x = (f(x + 1, y - 1) - f(x - 1, y - 1)) + C(f(x + 1, y) - f(x - 1, y)) \\ + (f(x + 1, y + 1) - f(x - 1, y + 1)), \quad (2.12)$$

$$G_y = (f(x - 1, y + 1) - f(x - 1, y - 1)) + C(f(x, y + 1) - f(x, y - 1)) + (f(x + 1, y + 1) - f(x + 1, y - 1)). \quad (2.13)$$

2.3. Convolución

La convolución en imágenes es una operación matemática que combina dos funciones: una imagen de entrada y un kernel (también conocido como máscara o filtro) para producir una nueva imagen resultante. El proceso de convolución se realiza mediante la superposición y multiplicación de los elementos de la imagen de entrada y el kernel, seguido de la suma de estos productos en cada ubicación de la imagen.

En el ámbito de las imágenes, la convolución se emplea generalmente para aplicar filtros o máscaras a una imagen con el fin de resaltar características específicas, como bordes, texturas o suavizar la imagen. El kernel es una matriz pequeña de valores que se desliza sobre la imagen de entrada, multiplicando y sumando los píxeles correspondientes en cada posición para generar el valor resultante en la imagen de salida.

La convolución en imágenes es una operación lineal y local, Esto implica que el valor de salida en cada posición se determina únicamente por los píxeles de la imagen de entrada que se multiplican por el kernel en esa posición. La operación de convolución es esencial en el procesamiento de imágenes y se aplica en diversas áreas, como el filtrado, la detección de bordes, el suavizado y el realce de características y muchas más.

En procesamiento de imágenes, la convolución se suele denotar con el símbolo \otimes .

2.4. Detectores de Bordes

Un detector de bordes es una herramienta clave en el procesamiento de imágenes y en la visión por computadora

Una herramienta clave para el procesamiento de imágenes es el detector de bordes

Una herramienta clave para el procesamiento de imágenes es el detector de bordes [36], cuyo objetivo principal es identificar los bordes dentro de una imagen. Los bordes son las líneas o curvas donde ocurre un cambio brusco en la intensidad del color o la luminancia de la imagen. Detectar estos bordes es crucial para tareas como la identificación de objetos, la segmentación de imágenes, la reconstrucción tridimensional y la visión computarizada

2.4.1. Funcionamiento de un Detector de Bordes

- **Preprocesamiento:**

- **Suavizado:** Antes de aplicar el detector de bordes, la imagen suele ser suavizada (por ejemplo, mediante un filtro Gaussiano) para reducir el ruido que podría generar detecciones falsas de bordes.

■ **Cálculo de Gradientes:**

- **Gradientes:** Los detectores de bordes generalmente determinan el gradiente de la imagen, que cuantifica la variación en la intensidad los píxeles y es un buen indicador de la presencia de un borde.
- **Magnitud y Dirección:** Se determina la magnitud del gradiente para evaluar la intensidad del borde y su dirección para identificar la orientación del borde.

■ **Aplicación del Detector de Bordes:**

- **Operadores de Gradiente:** Se utilizan operadores específicos para detectar bordes, tales como el operador de Sobel, Prewitt, o Roberts.
- **Detección No Máxima:** Para obtener bordes más precisos, se puede aplicar la supresión no máxima, que elimina todos los píxeles que no sean máximos locales en la dirección del gradiente.

■ **Umbralización:**

- **Umbrales:** Se aplican uno o más umbrales para clasificar los píxeles como bordes o no bordes. En algunos métodos, como el detector de Canny, se utilizan dos umbrales para la detección de bordes fuertes y débiles.

2.4.2. Ejemplos de Detectores de Bordes

Hay diversos operadores que se emplean comúnmente en el procesamiento de imágenes para identificar bordes. Algunos de los más comunes y reconocidos son los operadores de Roberts, Prewitt y Sobel. Estos operadores son altamente efectivos y han sido objeto de extenso análisis en la literatura.

Operador de Roberts

es un operador utilizado para la detección de bordes en imágenes que utiliza una máscara de convolución para resaltar cambios abruptos en la intensidad luminosa. Fue propuesto por Lawrence Roberts en la década de 1960 como una alternativa simple y eficaz en la identificación de bordes.

El operador de Roberts se fundamenta en la evaluación de las diferencias de luminosidad entre píxeles vecinos en una imagen. Para ello, utiliza dos máscaras de tamaño 2×2

Máscara utilizada para G_x

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{2.14}$$

máscara utilizada para G_y

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{2.15}$$

Estas máscaras se aplican de forma deslizante sobre la imagen, calculando la diferencia de luminosidad entre los píxeles vecinos en dirección diagonal. En cada punto de la imagen, se obtiene un valor que refleja la magnitud absoluta del gradiente espacial estimado.

El operador de Roberts es especialmente útil en casos donde se desea una detección de bordes rápida y sencilla, aunque puede presentar una menor sensibilidad a bordes diagonales en comparación con otros operadores más complejos. Se emplea en distintas aplicaciones de procesamiento de imágenes, incluyendo la segmentación, el reconocimiento de objetos y el análisis de texturas.

Operador de Prewitt

Es un operador de detección de bordes en imágenes que se utiliza para resaltar cambios abruptos en la intensidad luminosa. Fue propuesto por Judith Prewitt en la década de 1970 y se fundamenta en la evaluación de gradientes espaciales en la imagen.

El detector de bordes de Prewitt emplea dos máscaras de convolución, una destinada a identificar bordes horizontales y otra para bordes verticales. Ambas máscaras tienen un tamaño de 3×3 y se aplican de forma deslizante sobre la imagen.

A partir de las ecuaciones (2.10) y (2.11) con $C = 1$, se obtienen las siguientes máscaras de convolución:

Máscara utilizada para G_x

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad (2.16)$$

máscara utilizada para G_y

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad (2.17)$$

y de manera escalar

$$\begin{aligned} G_x = & (f(x+1, y-1) - f(x-1, y-1)) + (f(x+1, y) - f(x-1, y)) \\ & + (f(x+1, y+1) - f(x-1, y+1)), \end{aligned} \quad (2.18)$$

$$\begin{aligned} G_y = & (f(x-1, y+1) - f(x-1, y-1)) + (f(x, y+1) - f(x, y-1)) \\ & + (f(x+1, y+1) - f(x+1, y-1)). \end{aligned} \quad (2.19)$$

Al aplicar estas máscaras en la imagen, se calculan las diferencias de luminosidad entre píxeles vecinos en las direcciones horizontal y vertical. Luego, se combinan las magnitudes de los gradientes obtenidos para obtener una imagen de salida que destaca los bordes existentes en la imagen original.

Operador de Sobel

Es una herramienta empleada para identificar bordes en imágenes. Fue propuesto por Irwin Sobel en 1968 y se basa en el cálculo de gradientes espaciales para resaltar cambios abruptos en la intensidad luminosa de la imagen.

El operador de Sobel emplea un par de máscaras de convolución: una para detectar bordes horizontales y otra para bordes verticales. Ambas máscaras son matrices de 3×3 que se aplican de forma deslizante sobre la imagen.

A partir de las ecuaciones (2.10) y (2.11) con $C = 2$, se derivan las siguientes máscaras de convolución

Máscara utilizada para G_x

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad (2.20)$$

máscara utilizada para G_y

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad (2.21)$$

y de manera escalar

$$\begin{aligned} G_x = & (f(x+1, y-1) - f(x-1, y-1)) + 2 * (f(x+1, y) - f(x-1, y)) \\ & +(f(x+1, y+1) - f(x-1, y+1)), \end{aligned} \quad (2.22)$$

$$\begin{aligned} G_y = & (f(x-1, y+1) - f(x-1, y-1)) + 2 * (f(x, y+1) - f(x, y-1)) \\ & +(f(x+1, y+1) - f(x+1, y-1)). \end{aligned} \quad (2.23)$$

Al aplicar estas máscaras en la imagen, se calculan las diferencias de luminosidad entre píxeles vecinos en las direcciones horizontal y vertical. Luego, se combinan las magnitudes de los gradientes obtenidos para obtener una imagen de salida que enfatiza los bordes en la imagen original.

El operador de Prewitt y el operador de Sobel

El operador de Prewitt y el operador de Sobel son técnicas comúnmente utilizadas en el procesamiento de imágenes para actividades como el análisis de bordes, la segmentación de objetos y la detección de contornos. Su uso generalizado se debe a su simplicidad y efectividad, lo que los hace muy populares en este campo.

2.4.3. Aplicaciones de los Detectores de Bordes

- **Reconocimiento de Objetos:** Identificación de formas y contornos para reconocer objetos en una imagen.

- **Segmentación de Imágenes:** Separación de una imagen en regiones distintas basadas en los bordes detectados.
- **Visión Artificial:** Aplicación en sistemas de visión artificial para funciones como la navegación autónoma y el control de calidad en procesos de manufactura.
- **Análisis de Imágenes Médicas:** Identificación de estructuras anatómicas y detección de anomalías en imágenes médicas.

2.5. Filtro de Kernel Gaussiano

Es un filtro empleado en el procesamiento de imágenes para suavizar o difuminar la imagen. Este filtro se fundamenta en la función de distribución gaussiana, comúnmente conocida como campana de Gauss. La fórmula general para calcular los valores del kernel Gaussiano es

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (2.24)$$

donde μ representa la media, σ es la desviación estándar y x corresponde a la entrada.

El tamaño del kernel gaussiano determina el nivel de suavizado aplicado a la imagen. Generalmente, se utiliza un kernel de tamaño impar. Por ejemplo, 3×3 , 5×5 , 7×7 , etc., para mantener la simetría y el punto central.

Para usar el filtro de kernel gaussiano en una imagen, se mueve el kernel sobre cada píxel de la imagen y se calcula el promedio ponderado de los valores de los píxeles adyacentes, utilizando los valores del kernel gaussiano como pesos.

Es importante mencionar que el tamaño y la desviación estándar del kernel gaussiano pueden modificarse según los requisitos específicos de cada aplicación. La desviación estándar (σ) del kernel gaussiano influye considerablemente en el efecto del filtro sobre la imagen. Este valor regula el nivel de suavizado y desenfoque aplicado. Un valor de σ muy elevado puede eliminar detalles importantes, mientras que un valor demasiado bajo puede no ser eficaz para eliminar el ruido.

2.6. Cálculo fraccionario

El cálculo fraccionario tiene sus raíces en el año 1695 e implica la extensión de las definiciones de derivada e integral ordinarias a órdenes fraccionarios. Esta extensión del cálculo tradicional permite considerar y modelar de manera más precisa diversos fenómenos y problemas reales, al incorporar efectos de memoria y filtrado en las ecuaciones diferenciales.

La principal ventaja de utilizar el cálculo fraccionario radica en su capacidad para capturar y representar adecuadamente ciertos comportamientos y propiedades que no son capturados por las derivadas e integrales de orden entero. Al incluir los órdenes fraccionarios, se pueden abordar fenómenos complejos que exhiben memoria a largo plazo, dependencia no local y otros efectos no lineales. Esto resulta particularmente útil en la modelización de sistemas y problemas reales, donde la precisión y la representación fiel de los fenómenos son de suma importancia.

Al aplicar el cálculo fraccionario, se obtienen ecuaciones diferenciales fraccionarias que brindan una descripción más completa y precisa de los fenómenos estudiados. Estas ecuaciones incorporan la memoria y los efectos de filtrado en el sistema, permitiendo una mejor aproximación y comprensión de los problemas reales. Así, el cálculo fraccionario ha emergido como una herramienta importante en múltiples áreas, incluyendo la física y la ingeniería, las ciencias aplicadas y el procesamiento de señales, entre otros, donde se requiere una modelización más precisa de los fenómenos complejos y dinámicos [37].

A continuación, se presentan algunas definiciones fundamentales del cálculo fraccionario:

Derivada fraccionaria: La derivada fraccionaria amplía el concepto de derivada de orden entero a órdenes fraccionarios. Se representa como $D^n f(t)$, donde D^n es el operador de derivada fraccionaria y $f(t)$ es la función sobre la que se realiza la derivada. La derivada fraccionaria captura la tasa de cambio de una función en un orden fraccionario, lo que permite analizar fenómenos con comportamientos complejos y no lineales.

Integral fraccionaria: La integral fraccionaria extiende el concepto de integral de orden entero a órdenes fraccionarios. Se denota como $I^n f(t)$, donde I^n es el operador de integral fraccionaria. La integral fraccionaria permite calcular el área bajo una curva en un orden fraccionario, lo que resulta beneficioso para modelar y analizar sistemas con memoria a largo plazo y dependencia no local.

Ecuación diferencial fraccionaria: Una ecuación diferencial fraccionaria es una ecuación que incluye derivadas fraccionarias e integrales fraccionarias. Estas ecuaciones generalizan las ecuaciones diferenciales ordinarias y permiten describir fenómenos y sistemas con características complejas y no lineales. Las ecuaciones fraccionarias son de gran importancia en áreas como la física, la ingeniería y la ciencia de datos, ya que brindan herramientas para modelar y resolver problemas más realistas y desafiantes.

Estas definiciones básicas del cálculo fraccionario proporcionan una base sólida para entender y utilizar los conceptos y métodos relacionados con este ámbito. El cálculo fraccionario ha demostrado ser una herramienta poderosa para abordar desafíos sofisticados y capturar fenómenos más precisos en diversas áreas de la ciencia y la tecnología.

2.6.1. La función Gamma (Γ)

La función gamma (Γ) es una función matemática que extiende el concepto de factorial a números reales y complejos [38]. Es una función que encuentra aplicaciones en múltiples áreas de las matemáticas, la física y la estadística. Está definida para $x > 0$ por la siguiente integral definida

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt \tag{2.25}$$

Esta integral es convergente para todos los valores positivos de x , lo que permite extender la noción de factorial, que normalmente se define solo para números enteros no negativos, se extiende a números reales y complejos.

Además de su definición a través de la integral, la función gamma también cumple con la siguiente relación de recurrencia:

$$\Gamma(x + 1) = x \cdot \Gamma(x) \tag{2.26}$$

Esta propiedad ilustra la relación entre la función gamma y el factorial, ya que $\Gamma(n) = (n - 1)!$ para números enteros positivos n .

La función gamma tiene numerosas aplicaciones en teoría de números, teoría de funciones especiales, física matemática y otras ramas de las matemáticas aplicadas. También es utilizada en la formulación de diversas distribuciones de probabilidad, como la distribución gamma.

Resumiendo, la función gamma es una herramienta matemática esencial que extiende el concepto de factorial a valores no enteros y complejos, y tiene múltiples aplicaciones en la modelización matemática y en la teoría de funciones especiales.

2.6.2. Definición de la derivada fraccionaria de Riemann-Liouville

La derivada de Riemann-Liouville es una ampliación de la derivada convencional que se utiliza en el contexto de cálculo fraccional. Mientras que la derivada estándar se aplica a funciones definidas en números enteros, esta derivada se aplica a funciones definidas en intervalos no enteros, incluyendo fracciones.

La derivada de Riemann-Liouville de orden α de una función $f(t)$ se expresa mediante la fórmula integral siguiente:

$$D^\alpha f(t) = \frac{1}{\Gamma(n - \alpha)} \frac{d^n}{dt^n} \int_a^t (t - \tau)^{n-\alpha-1} f(\tau) d\tau \tag{2.27}$$

donde n es el entero más cercano mayor o igual a α , Γ es la función gamma y a es el límite inferior del intervalo de integración. La integral es una integral de Riemann definida.

En esta fórmula, α puede ser cualquier número real, lo que posibilita extender el concepto de derivada a valores no enteros. Cuando α es un número entero positivo, la derivada de Riemann-Liouville se convierte en la derivada ordinaria de orden α . Este concepto es particularmente valioso en el estudio de ecuaciones diferenciales fraccionarias y en el análisis de sistemas dinámicos con órdenes no enteros. Su aplicación se ha vuelto significativa en varios campos de la física y la ingeniería, así como en la modelización matemática de fenómenos complejos.[37].

2.7. Derivada conformable

Desde una perspectiva práctica, las definiciones de derivadas e integrales fraccionarias no tienen un significado físico y geométrico tan claro como las derivadas e integrales de orden entero [39]. Además, es relevante mencionar que las derivadas e integrales fraccionarias no siguen las reglas del cálculo tradicional, como la regla de la cadena, la regla del producto y la regla del cociente, entre otras. Fue Khalil quien introdujo el concepto de derivada conformable, una noción que, si bien no representa una derivada fraccionaria en sí misma, puede considerarse como una extensión de las derivadas ordinarias. La derivada conformable proporciona una forma alternativa de generalizar las derivadas tradicionales, permitiendo abordar problemas en los que se requiere considerar efectos de memoria o dependencias no locales.

A diferencia de las derivadas fraccionarias, que involucran órdenes fraccionarios y capturan comportamientos más complejos, La derivada conformable se fundamenta en el concepto de preservar la estructura

de las derivadas ordinarias, pero adaptándola para tratar fenómenos con propiedades adicionales. Esto brinda una mayor flexibilidad y una representación más completa de ciertos problemas en los que la memoria y las dependencias no locales son relevantes.

Esta derivada conformable se distingue por adherirse a las propiedades de las derivadas de orden entero y, adicionalmente, actúa como un operador matemático local. Un operador matemático se considera local si su efecto en un punto depende exclusivamente de los valores de la función y sus derivadas en una vecindad infinitesimal de ese punto. Esto significa que para evaluar el operador en un punto dado, solo se necesita información en una vecindad inmediata del punto, y no se requiere información global de la función en todo su dominio. Debido a estas características, la derivada conformable ha despertado un notable interés en la investigación de sus propiedades y aplicaciones en ciencia, ingeniería y ciencia de datos, donde se requiere una descripción más precisa y flexible de los sistemas y fenómenos estudiados. A través de esta extensión de las derivadas ordinarias, es posible abordar problemas complejos que involucran interacciones a largo plazo y efectos no locales, ampliando así las herramientas disponibles para el análisis y la modelización de sistemas dinámicos y fenómenos complejos. [17].

2.8. Derivada \mathcal{M} -truncada fraccionaria

La derivada \mathcal{M} -truncada de f de orden α , denotada como ${}_i^{\mathcal{M}}\mathcal{D}^{\alpha,\beta}\{f(t)\}$, se define a continuación

$${}_i^{\mathcal{M}}\mathcal{D}^{\alpha,\beta}\{f(t)\} = \lim_{\epsilon \rightarrow 0} \frac{f(t_i E_\beta(\epsilon t^{-\alpha})) - f(t)}{\epsilon}, \quad (2.28)$$

donde la función truncada de Mittag-Leffler con un parámetro se define de la siguiente manera.

$${}_i E_\beta(z) = \sum_{k=0}^i \frac{z^k}{\Gamma(\beta k + 1)}. \quad (2.29)$$

Algunas características de esta derivada se describen en el teorema siguiente. [40].

Teorema 1. *Sea $0 < \alpha \leq 1, \beta > 0, a, b \in \mathbb{R}$ y f, g α -diferenciable en el punto $t > 0$. Entonces*

1. ${}_i^{\mathcal{M}}\mathcal{D}^{\alpha,\beta}(af + bg)(t) = a{}_i^{\mathcal{M}}\mathcal{D}^{\alpha,\beta}f(t) + b{}_i^{\mathcal{M}}\mathcal{D}^{\alpha,\beta}g(t)$.
2. ${}_i^{\mathcal{M}}\mathcal{D}^{\alpha,\beta}(f \cdot g)(t) = f(t){}_i^{\mathcal{M}}\mathcal{D}^{\alpha,\beta}g(t) + g(t){}_i^{\mathcal{M}}\mathcal{D}^{\alpha,\beta}f(t)$.
3. ${}_i^{\mathcal{M}}\mathcal{D}^{\alpha,\beta}\left(\frac{f}{g}\right)(t) = \frac{g(t){}_i^{\mathcal{M}}\mathcal{D}^{\alpha,\beta}f(t) - f(t){}_i^{\mathcal{M}}\mathcal{D}^{\alpha,\beta}g(t)}{[g(t)]^2}$.
4. ${}_i^{\mathcal{M}}\mathcal{D}^{\alpha,\beta}(c) = 0$, donde $f(t) = c$ es una constante.
5. (Regla de la cadena). Si f es diferenciable, entonces ${}_i^{\mathcal{M}}\mathcal{D}^{\alpha,\beta}\{f(t)\} = \frac{t^{1-\alpha}}{\Gamma(\beta+1)} \frac{df(t)}{dt}$.

A continuación se presenta el siguiente operador el cual incluye la ley de potencia, el decaimiento exponencial y la memoria Mittag-Leffler completa en convolución con la derivada \mathcal{M} -truncada [41].

$${}_0^{\text{MRL}}\mathcal{D}_t^{\alpha,\beta,\gamma}\{y(t)\} = \frac{1}{\Gamma(n-\gamma)} {}_i^{\mathcal{M}}\mathcal{D}^{\alpha,\beta} \int_0^t \frac{f(\tau)}{(t-\tau)^{\gamma-n+1}} d\tau, \quad n-1 < \alpha, \gamma \leq n, \quad \beta > 0. \quad (2.30)$$

2.9. Detector de bordes generalizado

A continuación, se propone un detector de bordes que utiliza gradientes fraccionarios en las direcciones x e y . Estos gradientes son modificados por los ángulos θ y ϕ , que permiten realzar las características de los bordes. Para cada componente, se calcula el valor absoluto, y finalmente, se suman todas las componentes para obtener el detector de bordes final, como se muestra a continuación

$$g_{xy} = \left| \cos(\theta) \cdot G_x^{\alpha\beta\gamma} \right| + \left| \sin(\theta) \cdot G_y^{\alpha\beta\gamma} \right| + \left| \cos(\phi) \cdot G_x^{\xi\eta\lambda} \right| + \left| \sin(\phi) \cdot G_y^{\xi\eta\lambda} \right|. \quad (2.31)$$

En esta ecuación, $G_x^{\alpha\beta\gamma}$, $G_y^{\alpha\beta\gamma}$, $G_x^{\xi\eta\lambda}$ y $G_y^{\xi\eta\lambda}$ representan los gradientes fraccionarios en las direcciones x e y . Estos gradientes son multiplicados por los factores angulares $\cos(\theta)$ y $\sin(\theta)$, donde permiten realzar características de los bordes en una dirección específica. Después de multiplicar los gradientes por los factores angulares correspondientes, se obtiene el valor absoluto de cada componente. Finalmente, todas las componentes se suman para obtener el detector de bordes final, g_{xy} , que combina las características de los bordes en las direcciones determinadas por θ y ϕ . Este enfoque aprovecha la flexibilidad de los gradientes fraccionarios y la capacidad de ajustar los ángulos para adaptarse a diferentes tipos de bordes y patrones presentes en las imágenes digitales.

Cuando

$$\begin{aligned} \theta &= 0, \\ \phi &= 90^\circ, \\ \alpha &= \xi, \\ \beta &= \eta, \\ \gamma &= \lambda, \end{aligned} \quad (2.32)$$

se tiene el detector de bordes tradicional

$$g_{xy} = \left| G_x^{\alpha\beta\gamma} \right| + \left| G_y^{\alpha\beta\gamma} \right|. \quad (2.33)$$

2.10. Matriz Hessiana

La matriz Hessiana es una matriz cuadrada formada por las segundas derivadas parciales, que proporciona información sobre la curvatura principal de una imagen. La curvatura principal de una imagen mide cómo cambia la dirección de la superficie en un punto específico. En el contexto del procesamiento de imágenes, especialmente en la detección de bordes y características, se examina cómo varía la intensidad de los píxeles en diversas direcciones alrededor de un punto específico. Al calcular los valores propios, es posible identificar y resaltar de manera efectiva los bordes y las características más relevantes en una imagen [42]. La incorporación de la matriz hessiana en el proceso de detección de bordes permite aprovechar información adicional presente en la imagen, lo que resulta beneficioso en imágenes

médicas. Al realzar los bordes, se facilita el análisis y la interpretación por parte de los profesionales de la salud, lo que puede realizar diagnósticos más certeros. A continuación se presenta el método sugerido para la detección de bordes en imágenes digitales mediante el uso de la matriz hessiana. La matriz Hessiana convencional se define mediante la siguiente fórmula:

$$H = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix}, \quad (2.34)$$

donde, I es la imagen en escala de grises.

$\frac{\partial^2 I}{\partial x^2}$ es la segunda derivada parcial de la imagen con respecto a x

$\frac{\partial^2 I}{\partial x \partial y}$ es la segunda derivada parcial de la imagen con respecto a x y con respecto a y .

$\frac{\partial^2 I}{\partial y^2}$ es la segunda derivada parcial de la imagen con respecto a y .

Las derivadas parciales se calculan utilizando las siguientes fórmulas.

$$\frac{\partial^{\alpha\beta\gamma} I}{\partial x^{\alpha\beta\gamma}} \approx (\mathcal{M}^{-RL} \mathcal{D}_t^{\alpha\beta\gamma} g(x) \otimes [1 \ 0 \ -1]) \otimes I, \quad (2.35)$$

$$\frac{\partial^{2\alpha\beta\gamma} I}{\partial x^{2\alpha\beta\gamma}} \approx (\mathcal{M}^{-RL} \mathcal{D}_t^{\alpha\beta\gamma} g(x) \otimes [1 \ 0 \ -1]) \otimes \frac{\partial^{\alpha\beta\gamma} I}{\partial x^{\alpha\beta\gamma}}. \quad (2.36)$$

$$\frac{\partial^{\alpha\beta\gamma} I}{\partial y^{\alpha\beta\gamma}} \approx \left(\mathcal{M}^{-RL} \mathcal{D}_t^{\alpha\beta\gamma} g(x) \otimes \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \right) \otimes I, \quad (2.37)$$

$$\frac{\partial^{2\alpha\beta\gamma} I}{\partial y^{2\alpha\beta\gamma}} \approx \left(\mathcal{M}^{-RL} \mathcal{D}_t^{\alpha\beta\gamma} g(x) \otimes \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \right) \otimes \frac{\partial^{\alpha\beta\gamma} I}{\partial y^{\alpha\beta\gamma}}. \quad (2.38)$$

$$\frac{\partial^{2\alpha\beta\gamma} I}{\partial x^{\alpha\beta\gamma} \partial y^{\alpha\beta\gamma}} \approx (\mathcal{M}^{-RL} \mathcal{D}_t^{\alpha\beta\gamma} g(x) \otimes [1 \ 0 \ -1]) \otimes \frac{\partial^{\alpha\beta\gamma} I}{\partial y^{\alpha\beta\gamma}}, \quad (2.39)$$

$$\frac{\partial^{2\alpha\beta\gamma} I}{\partial x^{\alpha\beta\gamma} \partial y^{\alpha\beta\gamma}} \approx \frac{\partial^{2\alpha\beta\gamma} I}{\partial y^{\alpha\beta\gamma} \partial x^{\alpha\beta\gamma}}, \quad (2.40)$$

donde, \otimes indica el operador de convolución.

Tomando en cuenta el gradiente fraccionario \mathcal{M} -truncado

Considerando el gradiente fraccionario \mathcal{M} -truncado en el sentido de Riemann-Liouville [2.30], se construye la matriz Hessiana en términos fraccionarios

$$\mathcal{M}\text{-RL } H^{\alpha\beta\gamma} = \begin{bmatrix} \frac{\partial^2 \alpha\beta\gamma I}{\partial x^{2\alpha\beta\gamma}} & \frac{\partial^2 \alpha\beta\gamma I}{\partial x^{\alpha\beta\gamma} \partial y^{\alpha\beta\gamma}} \\ \frac{\partial^2 \alpha\beta\gamma I}{\partial y^{\alpha\beta\gamma} \partial x^{\alpha\beta\gamma}} & \frac{\partial^2 \alpha\beta\gamma I}{\partial y^{2\alpha\beta\gamma}} \end{bmatrix}. \quad (2.41)$$

La matriz Hessiana se emplea para calcular la curvatura principal, la cual define los bordes y sus resultados finales [43]. En esta matriz, λ_1 y λ_2 representan los valores máximo y mínimo, respectivamente, es decir, los valores propios (eigenvalores) de la matriz Hessiana [42]. Dado que λ_1 presenta menos contraste, se selecciona λ_2 para los fines de este estudio, ya que este valor propio ofrece una mayor información sobre los bordes y la curvatura principal.

La curvatura principal se expresa mediante las siguientes fórmulas:

$$P(x^{\alpha\beta\gamma}) = \max(\lambda_1, 0), \quad \forall \alpha, \beta, \gamma \in (0, 1], \quad (2.42)$$

y

$$P(x^{\alpha\beta\gamma}) = \min(\lambda_2, 0), \quad \forall \alpha, \beta, \gamma \in (0, 1]. \quad (2.43)$$

2.11. Índices de desempeño en el procesamiento de imágenes

En la literatura, se han desarrollado varias métricas para evaluar diferentes características de una imagen procesada en relación con la imagen original. En el contexto de una imagen, la entropía se refiere a una medida de la imprevisibilidad o irregularidad en los niveles de intensidad de los píxeles. En otras palabras, la entropía cuantifica cuánta información se necesita para describir la imagen en términos de variaciones en sus niveles de intensidad.

Entropía

La entropía (H), basada en la teoría de la información, se emplea en el procesamiento de imágenes y visión por computadora para diversas aplicaciones, incluyendo segmentación de imágenes, detección de bordes, compresión de imágenes y evaluación de calidad. Una imagen con alta entropía es considerada más compleja y rica en información, mientras que una imagen con baja entropía es más uniforme y predecible [44].

La fórmula para calcular la entropía de una imagen en escala de grises es

$$H = - \sum_k p_k \log_2(p_k), \quad (2.44)$$

donde p_k es la probabilidad de que un píxel tenga un valor de intensidad k en la imagen.

La evaluación del nivel de ruido en la detección de bordes frecuentemente se lleva a cabo empleando métricas estándar como la relación señal-ruido (PSNR) y el error cuadrático medio (MSE). Aunque

estas métricas no son específicas para evaluar la precisión de la detección de bordes, son útiles para evaluar la robustez de los algoritmos, ya que proporcionan una medida cuantitativa del nivel de ruido presente en las imágenes resultantes.

PSNR (Peak Signal-to-Noise Ratio)

El PSNR (Relación Señal-Ruido en el Pico) es una métrica empleada para evaluar la calidad de una imagen en relación con una imagen de referencia. Su cálculo se realiza mediante la siguiente fórmula:

$$\text{PSNR} = 10 \log \frac{\text{máx}(I)^2}{MSE}, \quad (2.45)$$

donde $\text{máx}(\cdot)$ representa el valor máximo que un píxel puede tener en la imagen. Por ejemplo, en una imagen en escala de grises de 8 bits, este valor es 255.

El MSE (Mean Squared Error), o error cuadrático medio, se calcula sumando los cuadrados de las diferencias entre los valores de los píxeles de la imagen de referencia y los de la imagen evaluada, y luego dividiendo por el total de píxeles.

$$\text{MSE} = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n [I_1(x, y) - I_2(xy)]^2, \quad (2.46)$$

donde,

- m y n son las dimensiones de las imágenes (en píxeles).
- $I_1(i, j)$ representa el valor del píxel en la posición (i, j) de la imagen original.
- $I_2(i, j)$ es el valor del píxel en la posición (i, j) de la imagen de referencia.

El PSNR se mide en decibelios (dB) y ofrece una evaluación cuantitativa de la calidad de la imagen. Un valor de PSNR más alto indica una mayor similitud entre las imágenes y una mejor calidad de la imagen evaluada en relación con la imagen de referencia. [36].

SSIM (Índice de Similitud Estructural)

El SSIM (Índice de Similitud Estructural) es una métrica que evalúa la calidad de una imagen al comparar la similitud estructural entre una imagen original y una imagen que ha sido procesada o comprimida. Esta métrica se emplea frecuentemente en el procesamiento y compresión de imágenes para determinar el grado de degradación de la calidad de la imagen después de aplicar algún procesamiento o compresión.

El SSIM se fundamenta en cómo percibe la visión humana y considera diversos aspectos de la calidad de una imagen, tales como luminancia, contraste y estructura. Compara la similitud entre regiones locales de las dos imágenes y ofrece un valor numérico que oscila entre 0 y 1. Un valor de 1 significa que las imágenes son idénticas en términos de estructura y calidad, mientras que valores inferiores reflejan mayores diferencias.

El SSIM se calcula considerando los siguientes componentes

- Luminancia (L): Mide la concordancia en la intensidad de los píxeles entre las dos imágenes.

- Contraste (C): Analiza la similitud en las variaciones de intensidad entre regiones locales de las imágenes.
- Estructura (S): Compara la similitud en la organización de las texturas de las dos imágenes.

Además, estas componentes proporcionan una medida general de cuánto se asemeja la imagen procesada a la imagen original en términos de percepción visual. Cuanto más cercano a 1 sea el valor del SSIM, mayor será la similitud estructural entre las imágenes. Además, es una herramienta clave para evaluar la calidad de las imágenes en diferentes campos [45]. Por lo tanto, podemos expresar SSIM como sigue

$$\text{SSIM}(x, y) = (L(x, y))^\alpha \cdot (C(x, y))^\beta \cdot (S(x, y))^\gamma. \quad (2.47)$$

Aquí, x y y son las dos imágenes que se están comparando, mientras que α , β y γ son parámetros que ajustan la relevancia de los componentes L , C y S , respectivamente. La selección de estos parámetros varía según la aplicación y cómo se desea valorar la luminancia, el contraste y la estructura en la evaluación de la calidad de la imagen.

BRISQUE (Blur/Sharpness Identification through Clustering and Kernel Estimation)

El término "BRISQUE" se refiere a un algoritmo y una métrica de calidad de imagen llamada "Blur/Sharpness Identification through Clustering and Kernel Estimation" (Identificación de Desenfoque/Nitidez mediante Agrupación y Estimación del Núcleo). Esta métrica se utiliza para evaluar la calidad de imágenes digitales y determinar si una imagen está borrosa o nítida.

El BRISQUE se fundamenta en métodos de aprendizaje automático y emplea un modelo estadístico para evaluar la calidad de la imagen. Esto incluye la extracción de características de la imagen, como estadísticas relacionadas con la textura y el contraste, y la aplicación de un modelo previamente entrenado para estimar la calidad de la imagen en términos de nitidez o desenfoque.

El BRISQUE proporciona un valor numérico que refleja la calidad de la imagen en términos de su claridad. Un valor bajo de BRISQUE indica que la imagen es nítida, mientras que un valor alto sugiere que la imagen podría estar borrosa o de menor calidad.

La fórmula para calcular el BRISQUE se fundamenta en un modelo estadístico que emplea diversas características extraídas de la imagen. Utiliza técnicas de aprendizaje automático y un modelo de regresión para evaluar la calidad de la imagen en términos de nitidez o desenfoque [46]. Así, el BRISQUE se representa como

$$\text{BRISQUE} = \mathbf{w}^T \mathbf{F}. \quad (2.48)$$

- \mathbf{w} representa un vector de pesos obtenido a partir de un modelo de regresión que ha sido entrenado usando un conjunto de datos de imágenes junto con sus evaluaciones de calidad subjetiva.
- \mathbf{F} es un vector de características extraídas de la imagen que se evalúa, basado en estadísticas naturales de la imagen (Natural Image Quality Evaluator) la cual es una métrica sin referencia utilizada para evaluar la calidad de imágenes. A diferencia de otras métricas que necesitan una imagen de referencia para realizar comparaciones, NIQE evalúa la calidad basándose únicamente en la imagen que se está examinando. NIQE se fundamenta en el análisis de las estadísticas naturales de las imágenes, utilizando un modelo de características derivadas de un conjunto de

imágenes de alta calidad. Estas características incluyen estadísticas locales de luminancia y contrastes, entre otras, que se extraen a partir de las sub-bandas de la imagen utilizando un marco de análisis multi-escala y multi-orientación.

- \mathbf{w}^T indica la transposición del vector de pesos \mathbf{w} , lo cual permite realizar la multiplicación matricial adecuada para obtener la puntuación final de calidad de la imagen.

El producto escalar $\mathbf{w}^T \mathbf{F}$ da como resultado la puntuación BRISQUE, que cuantifica la calidad perceptual de la imagen.

2.12. Función objetivo.

Una función objetivo, también denominada función de costo o función de pérdida, es una función matemática que se busca optimizar. Dependiendo del contexto, la optimización puede involucrar la maximización o minimización de esta función.

2.12.1. Definición y Propósito

Definición: La función objetivo es una expresión matemática que describe el objetivo que se pretende lograr. Puede depender de varias variables y parámetros, y su forma específica está determinada por el problema que se está resolviendo.

Propósito: El objetivo principal es determinar los valores de las variables que optimizan, ya sea minimizando o maximizando, la función objetivo. En el ámbito del aprendizaje automático, por ejemplo, se busca minimizar una función de pérdida que cuantifica el error en las predicciones.

En el ámbito de la Optimización por Enjambre de Partículas (PSO), una técnica de optimización inspirada en la inteligencia colectiva, la función objetivo se emplea para que las partículas evalúen la calidad de sus posiciones actuales en el espacio de búsqueda. Por ejemplo, si estamos minimizando una función de costo en PSO, cada partícula ajusta su posición y velocidad basándose en el valor de la función objetivo en su ubicación actual. [47].

Capítulo 3

Gradiente Gaussiano \mathcal{M} -truncado fraccionario

El propósito de este estudio es optimizar la calidad de las imágenes médicas y para esto se proponen técnicas de procesamiento fraccionarias.

Las imágenes son afectadas al momento de hacer la adquisición de la imagen, lo que se quiere es eliminar el ruido, así como los bordes gruesos, se han propuesto máscaras diferenciales de primero y segundo orden, pero los de primer orden producen bordes gruesos y los de segundo orden son sensibles al ruido. Se ha demostrado que los operadores fraccionarios han mejorado estos inconvenientes.

3.1. Generalización del kernel Gaussiano vía \mathcal{M} -truncada fraccionaria

Para obtener un kernel gaussiano a partir de una función de distribución continua, se puede seguir un proceso de interpolación en el que se muestrean los valores de esta función en ubicaciones discretas. Luego, es común normalizar los valores para asegurar que la suma de los elementos del kernel sea uno, lo que facilita su aplicación en operaciones de convolución.

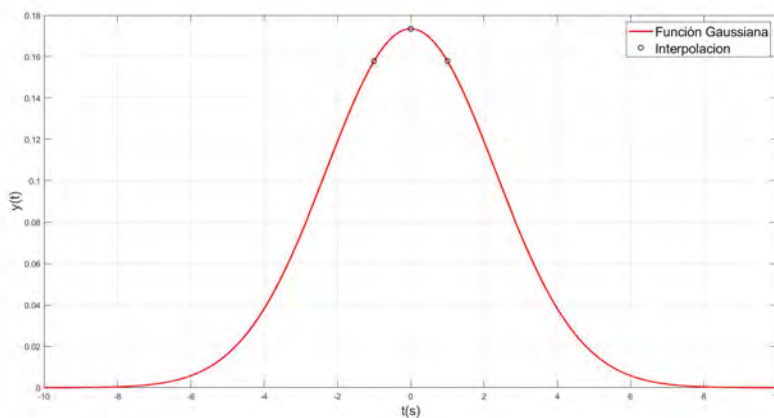


Fig. 3.1. Función de distribución gaussiana y el kernel gaussiano.

El principio de separabilidad en el contexto de kernels en dos dimensiones se refiere a la posibilidad de expresar un kernel bidimensional como el producto de dos kernels unidimensionales. Este principio es particularmente útil en situaciones donde la función que se desea modelar o transformar puede ser separada en sus dimensiones.

Para generalizar el Kernel gaussiano utilizando la derivada \mathcal{M} -truncada fraccionaria primero se obtiene la derivada del kernel gaussiano cuya expresión se muestra en (2.24) como se muestra a continuación

$$\dot{g}(x) = -\frac{(x-\mu)^2}{2\sqrt{2\pi}\sigma^3} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (3.1)$$

y se puede generalizar utilizando la derivada \mathcal{M} -truncada fraccionaria como sigue

$${}_0^{MRL} \mathcal{D}_t^{\alpha, \beta, \gamma} \{g(x)\} = -\frac{t^{1-\alpha}}{\Gamma(\beta+1)} \frac{(x-\mu)^2}{2\sqrt{2\pi}\sigma^3} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (3.2)$$

Para aproximar la solución de la ecuación (3.2), aplicamos la integral de Lotka-Volterra a ambos lados de la ecuación, obteniendo así

$$g(x) = f_0 + \frac{1}{\Gamma(\gamma)} \int_0^x \frac{\lambda^{1-\alpha} (x-\lambda)^{\gamma-1}}{\Gamma(\beta+1)} f(x, \lambda) d\lambda. \quad (3.3)$$

En un punto específico $x = x_{n+1}, n = 0, 1, 2, \dots$, la ecuación anterior se reescribe como

$$g_{n+1} = f_0 + \frac{1}{\Gamma(\gamma)} \int_0^{x_{n+1}} \frac{\lambda^{1-\alpha} (x_{n+1}-\lambda)^{\gamma-1}}{\Gamma(\beta+1)} f(x, \lambda) d\lambda. \quad (3.4)$$

Las función $g(x, \lambda)$, se puede aproximar usando la interpolación Lagrangiana por partes de la siguiente manera

$$g_{n+1} = f_0 + \frac{1}{\Gamma(\gamma)} \sum_{j=0}^n \int_{x_j}^{x_{j+1}} \frac{\lambda^{1-\alpha} (t_{n+1}-\lambda)^{\gamma-1}}{\Gamma(\beta+1)} f(x, \lambda) d\lambda. \quad (3.5)$$

Finalmente, en $[x_j, x_{j+1}]$ se obtiene una estimación numérica para la solución de la ecuación presentada en (3.2)

$$g_{n+1} = f_0 + \frac{(\Delta t)^\gamma}{\Gamma(\gamma+2)} \sum_{j=0}^n \left[\frac{x_j^{1-\alpha} f(x_j)}{\Gamma(\beta+1)} \left((n+1-j)^\gamma \times (n-j+2+\gamma) - (n-j)^\gamma (n-j+2+2\gamma) \right) - \frac{x_{j-1}^{1-\alpha} f(x_{j-1})}{\Gamma(\beta+1)} \left((n-j+1)^{\gamma+1} - (n-j)^\gamma (n-j+1+\gamma) \right) \right]. \quad (3.6)$$

A continuación, se muestran las soluciones de la ecuación diferencial fraccionaria utilizando los tres parámetros $(\alpha, \beta \text{ y } \gamma)$ iguales a uno y con los tres parámetros diferentes.

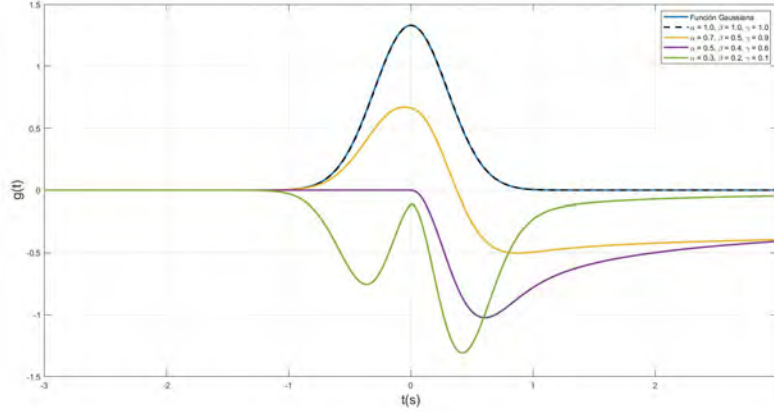


Fig. 3.2. respuestas a la ecuación diferencial fraccionaria con $\mu = 0$ y $\sigma = 0.3$

En la Figura 3.2 se observa que cuando α , β y $\gamma = 1$, obtenemos la función de distribución gaussiana clásica.

3.2. Gradiente Gaussiano \mathcal{M} -truncado fraccionario de orden constante

Para obtener el kernel gaussiano generalizado mediante la derivada \mathcal{M} -truncada fraccionaria de tercer orden, se realiza la interpolación de la solución de la ecuación diferencial fraccionaria. (3.2).

El gradiente gaussiano que involucra la derivada fraccionaria \mathcal{M} -truncada se obtiene a partir de las siguientes expresiones.

A lo largo de x esta dado por

$$\mathcal{M}\text{-RL}\nabla I_x = g^{\alpha,\beta,\gamma} \otimes h_x \otimes I = G_x^{\alpha\beta\gamma}, \quad (3.7)$$

mientras que a lo largo de y

$$\mathcal{M}\text{-RL}\nabla I_y = g^{\alpha,\beta,\gamma} \otimes h_y \otimes I = G_y^{\alpha\beta\gamma}, \quad \alpha, \beta, \gamma \in (0, 1), \quad (3.8)$$

donde I es una imagen en tonos grises y además

$$h_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}, \quad (3.9)$$

$$h_y = h_x^T = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}. \quad (3.10)$$

Las ecuaciones (3.11) y (3.10) representan los filtros derivativos a lo largo de x e y , respectivamente.

$$\mathcal{M}\text{-RL}\nabla I_x y = g_{xy} \quad (3.11)$$

Por lo tanto, tomando en consideración las ecuaciones (3.7) y (3.8), representamos el detector de bordes como se muestra a continuación

$$\mathcal{M}^{-RL}\nabla I_{xy} = \left| \mathcal{M}^{-RL}\nabla I_x \right| + \left| \mathcal{M}^{-RL}\nabla I_y \right|. \quad (3.12)$$

$$g_{xy} = \left| G_x^{\alpha,\beta,\gamma} \right| + \left| G_y^{\alpha,\beta,\gamma} \right|. \quad (3.13)$$

y al final sumamos o restamos el resultado obtenido del detector de bordes con la imagen en tonos grises.

Todo el proceso anterior se representa en el siguiente diagrama de bloques, Figura 3.3.

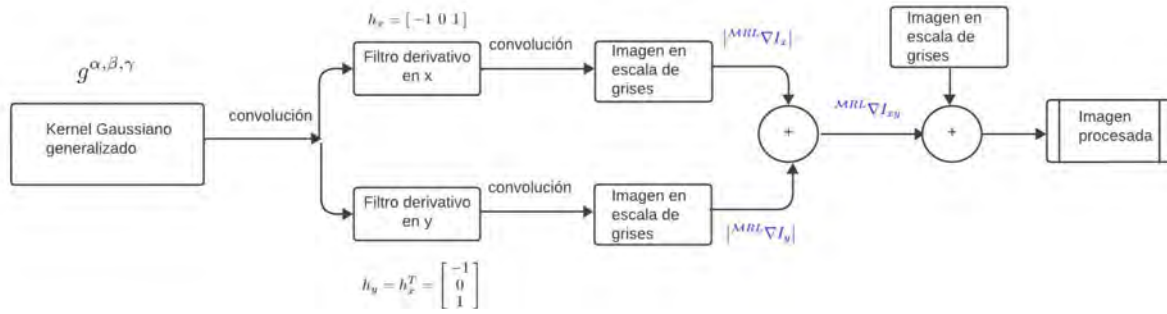


Fig. 3.3. Metodología para el procesamiento de imágenes usando la derivada- \mathcal{M} truncada fraccionaria.

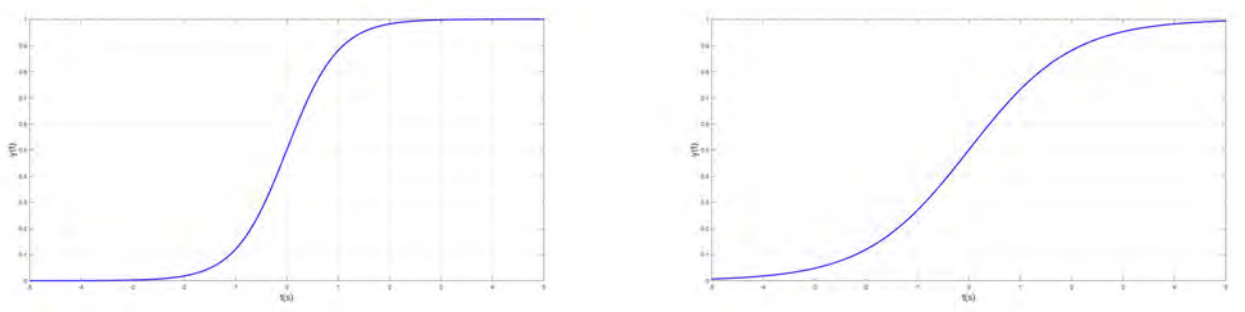
3.3. Gradiente Gaussiano \mathcal{M} -truncado fraccionario de orden variable

El término "función suave" se emplea de manera informal para referirse a una función matemática que es continua y tiene derivadas en todos los puntos de su dominio y cuyas derivadas son continuas. La suavidad de una función implica que no tiene discontinuidades bruscas, picos, saltos abruptos o cambios repentinos en su comportamiento.

En términos más técnicos, una función se considera suave si es infinitamente diferenciable en todo su dominio. Esto significa que no solo tiene derivadas de primer orden (es decir, es diferenciable), Además, tiene derivadas de órdenes superiores en cada punto. La suavidad de una función es una propiedad deseada en muchos contextos matemáticos y científicos, ya que funciones suaves suelen ser más fáciles de trabajar y analizar.

En el contexto del análisis numérico y la aproximación de funciones, también puede referirse a funciones que pueden ser aproximadas de manera precisa por funciones polinómicas o funciones continuas. En resumen, el término función suave generalmente se refiere a funciones que exhiben un comportamiento continuo y diferenciable en su dominio [48].

A continuación, en la Figura 3.4 se muestran dos funciones suaves que oscilan entre 0 y 1.



(a) $y(t) = \tanh(t)$

(b) $y(t) = \frac{1}{1+e^{-t}}$

Fig. 3.4. Funciones suaves. (a) tangente hiperbolica, (b) Sigmoidal.

El gradiente gaussiano que involucra la derivada fraccionaria \mathcal{M} -truncada de orden variable se obtiene a partir de las siguientes expresiones.

A lo largo de x esta dado por

$$\mathcal{M}\text{-}RL\nabla I_x = g^{\alpha(t),\beta(t),\gamma(t)} \otimes h_x \otimes I = G_x^{\alpha(t)\beta(t)\gamma(t)} \quad (3.14)$$

mientras que a lo largo de y

$$\mathcal{M}\text{-}RL\nabla I_y = g^{\alpha(t),\beta(t),\gamma(t)} \otimes h_y \otimes I = G_y^{\alpha(t)\beta(t)\gamma(t)} \quad (3.15)$$

donde h_x , h_y estan dadas por las expresiones (3.7) y (3.8), $\alpha(t)$, $\beta(t)$ y $\gamma(t) \in (0, 1)$, son funciones suaves e I es una imagen en tonos grises.

Con lo cual el detector de bordes \mathcal{M} -truncado fraccionario de orden variable esta dado por la siguiente expresi3n:

$$g_{xy} = \left| G_x^{\alpha(t)\beta(t)\gamma(t)} \right| + \left| G_y^{\alpha(t)\beta(t)\gamma(t)} \right|. \quad (3.16)$$

Capítulo 4

Gradiente Gaussiano \mathcal{M} -truncado fraccionario aplicado a imágenes digitales

4.1. Gradiente Gaussiano \mathcal{M} -truncado fraccionario de orden constante aplicado a imágenes médicas

Para utilizar el detector de bordes mostrado en (2.31), es necesario optimizar los siguientes parámetros: α , β , γ , ξ , η y λ , que representan los diferentes órdenes; θ y ϕ , que representan los ángulos direccionales; y σ , que es la desviación estándar. La optimización de estos parámetros se realizó utilizando el algoritmo de Optimización por Enjambre de Partículas (PSO), con el fin de determinar los valores óptimos.

4.1.1. Función objetivo

La función objetivo empleada se define como una función que acepta dos argumentos: un vector de parámetros \mathbf{x} (que incluye α , β , γ , ξ , η , λ , θ , ϕ y σ) y una imagen la cual es la imagen que se va a procesar. Esta función calcula el error cuadrático medio (MSE) del modelo de regresión lineal, empleando los parámetros \mathbf{x} y la imagen. El objetivo es minimizar este error cuadrático medio, ajustando los parámetros del modelo para obtener la mejor aproximación posible a los datos de la imagen.

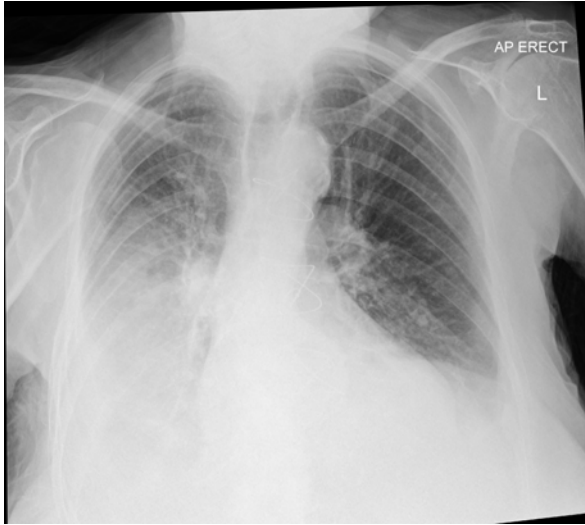
Aquí se presentan los resultados obtenidos al aplicar el detector de bordes generalizado (2.31) con los valores optimizados de los parámetros α , β , γ , ξ , η , λ , θ , ϕ y σ . Estos resultados se combinan tanto sumando como restando la imagen original con el resultado del detector de bordes generalizado, como se muestra en el diagrama de bloques de la Figura 3.3.

4.1.2. Resultados

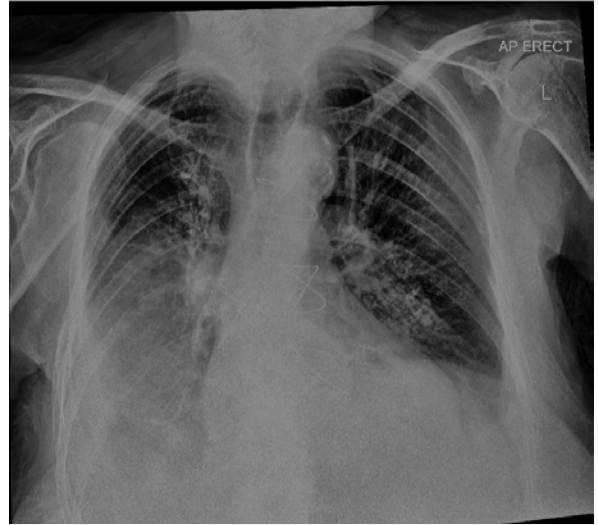
La Figura 4.1b presenta el resultado de procesar la imagen “Lung abscess and pleural empyema” con el detector de bordes generalizado, obteniéndose un error cuadrático medio de $MSE = 0.0014$. Los

CAPÍTULO 4. GRADIENTE GAUSSIANO \mathcal{M} -TRUNCADO FRACCIONARIO APLICADO A IMÁGENES DIGITALES

parámetros optimizados utilizados fueron: $\alpha = 0.610161$, $\beta = 0.752903$, $\gamma = 0.625035$, $\xi = 0.744537$, $\eta = 0.6$, $\lambda = 0.621689$, $\theta = 0.983921$, $\phi = 3.141849$ y $\sigma = 0.515605$.



(a) Imagen "Lung abscess and pleural empyema"



(b) Imagen procesada

Fig. 4.1. (a) Imagen "Lung abscess and pleural empyema". (b) Imagen procesada con el detector de bordes generalizado.

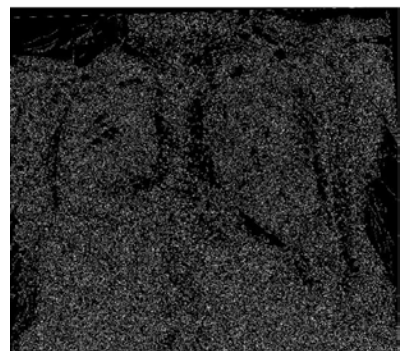
A continuacion en la Figura 4.2 se muestran los resultados al procesar la imagen "Lung abscess and pleural empyema" con el detector de bordes de Prewitt, el detector de bordes de Sobel y el detector de bordes de Canny.



(a) Operador Prewitt



(b) Operador Sobel



(c) Operador Canny

Fig. 4.2. Detectores de bordes clásicos a)Prewitt, b)Sobel, c)Canny

Finalmente se realiza el procedimiento mostrado en la Figura 3.3. En la Figura 4.3 se presentan los resultados de sumar y restar la imagen "Lung abscess and pleural empyema" con el resultado de aplicar el detector de bordes generalizado a dicha imagen,

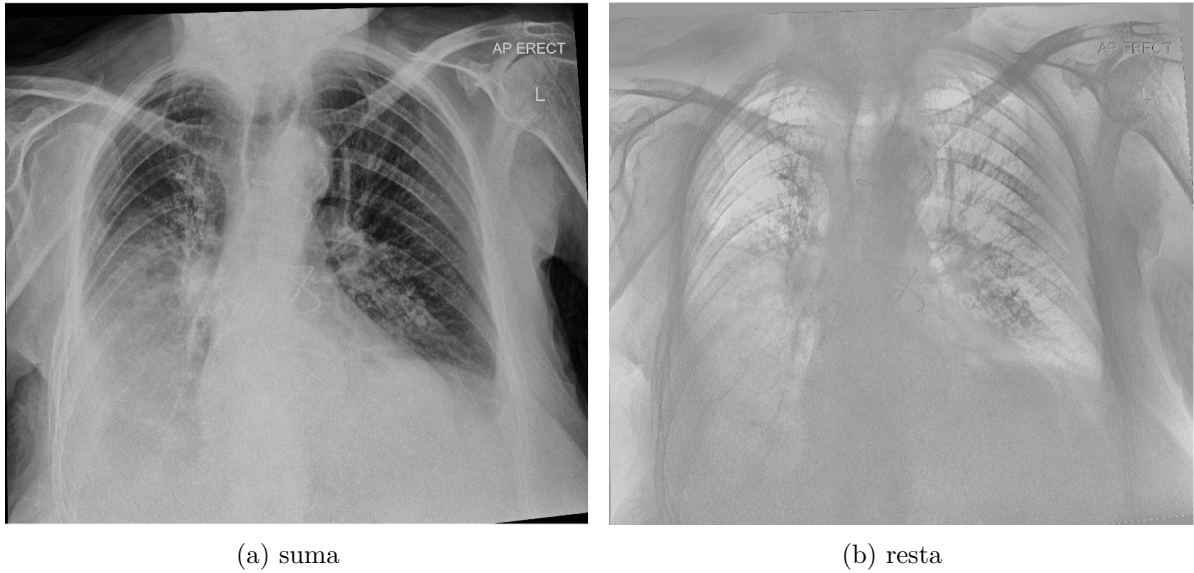


Fig. 4.3. (a) suma de la imagen “Lung abscess and pleural empyema” con el detector de bordes generalizado. (b) resta de la imagen “Lung abscess and pleural empyema” con el detector de bordes generalizado.

A continuación, la tabla 4.1 presenta los valores obtenidos al calcular las métricas para la imagen “Lung abscess and pleural empyema” obtenida con el detector de bordes generalizado dado en (2.31), el operador de Prewitt, el operador de Sobel y el algoritmo de Canny. La entropía de la imagen “Lung abscess and pleural empyema” es de 7.3459

Tabla 4.1. Resultados de las métricas de la imagen “Lung abscess and pleural empyema”

Operador	Entropía	PSNR	SSIM	BRISQUE
Bordes generalizado	7.6243	28.5218	0.7261	37.7641
Prewitt	0.0995	3.5407	0.0094	46.0652
Sobel	0.1021	3.5428	0.0094	46.3362
Canny	0.5891	4.0229	0.0110	43.4754

La Figura 4.4b presenta el resultado de procesar la imagen “covid2” con el detector de bordes generalizado, se obtuvo un error cuadrático medio de $MSE = 0.004$. Los parámetros optimizados utilizados fueron: $\alpha = 0.6$, $\beta = 0.9999$, $\gamma = 0.6$, $\xi = 0.9999$, $\eta = 0.702673$, $\lambda = 0.9999$, $\theta = 0.783621$, $\phi = 3.0123$ y $\sigma = 0.3$.

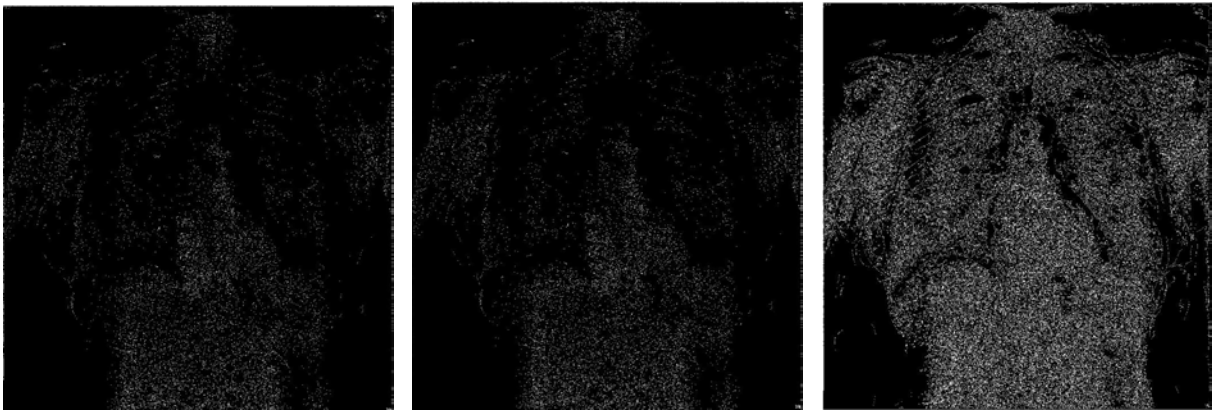


(a) Imagen "covid".

(b) Imagen procesada

Fig. 4.4. (a) Imagen "covid2". (b) Imagen procesada con el detector de bordes generalizado.

A continuación en la Figura 4.5 se muestran los resultados al procesar la imagen "covid2" con el detector de bordes de Prewitt, el detector de bordes de Sobel y el detector de bordes de Canny.



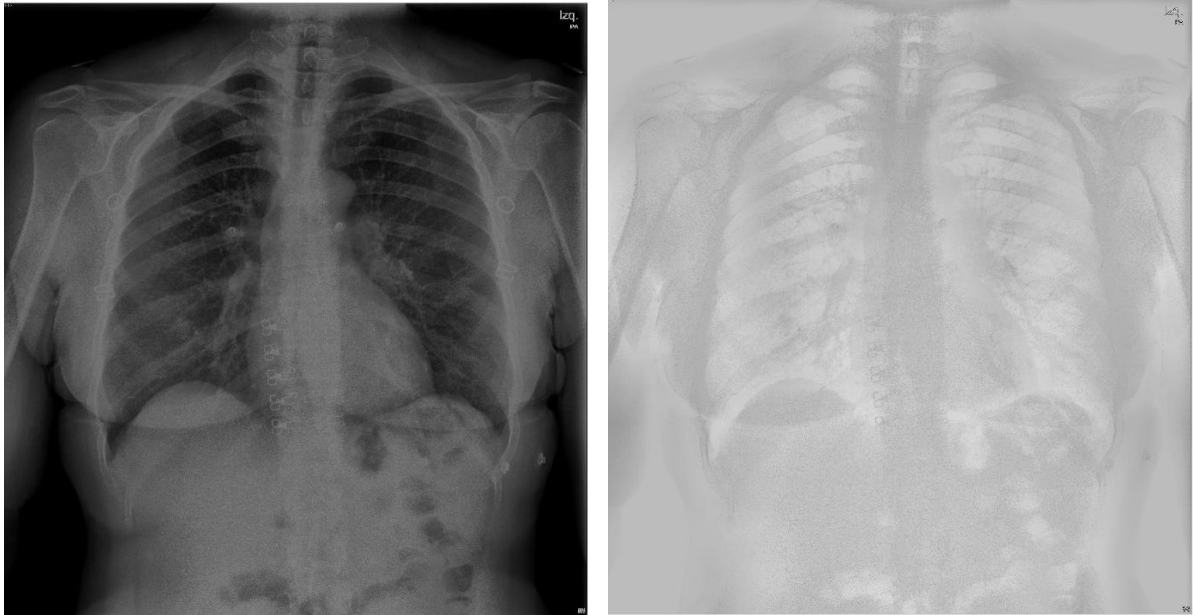
(a) Operador Prewitt

(b) Operador Sobel

(c) Operador Canny

Fig. 4.5. Detectores de bordes clásicos a)Prewitt, b)Sobel, c)Canny

Finalmente se realiza el procedimiento mostrado en el diagrama mostrado en la Figura 3.3. En la Figura 4.6 se presentan los resultados de sumar y restar la imagen "covid2" con el resultado de aplicar el detector de bordes generalizado a dicha imagen,



(a) suma imagen “covid2” con el detector de bordes generalizado. (b) resta imagen “covid2” con el detector de bordes generalizado.

Fig. 4.6. Resultados al sumar y restar la imagen “covid2” con el detector de bordes generalizado

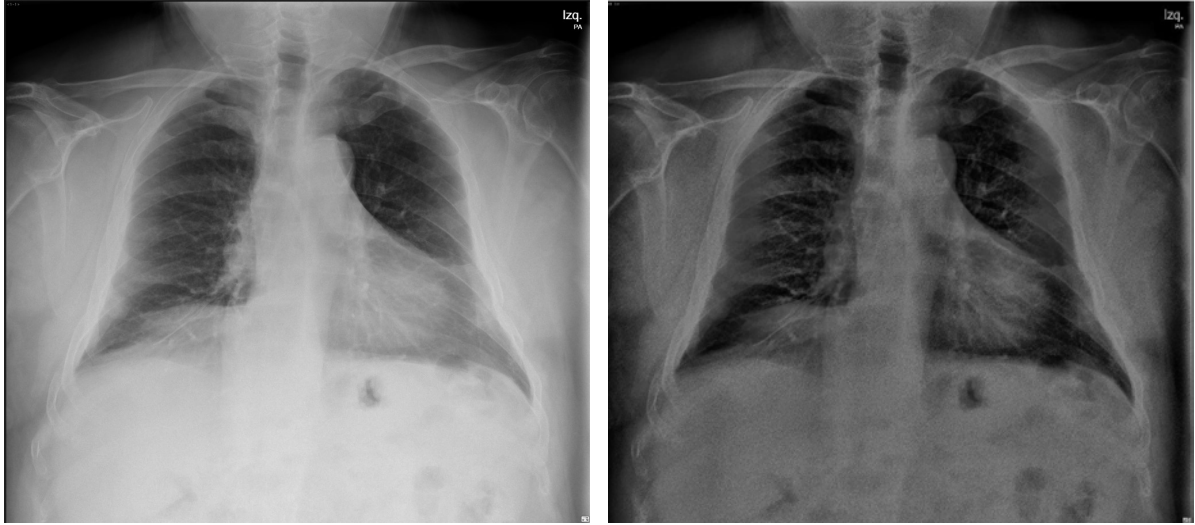
A continuación en la tabla 4.2 se muestran los valores obtenidos al calcular las métricas para la imagen “covid2” obtenida con el detector de bordes generalizado dado en la expresión (2.31), el operador de Prewitt, el operador de Sobel y el algoritmo de Canny.

La entropía de la imagen “covid2” es de 7.6242

Tabla 4.2. Resultados de las métricas de la imagen “covid2”

Operador	Entropía	PSNR	SSIM	BRISQUE
Bordes generalizado	7.6248	28.9416	0.8077	38.4863
Prewitt	0.0995	5.2079	0.0312	48.0881
Sobel	0.1515	5.2165	0.0312	48.5045
Canny	0.5903	5.5270	0.0323	43.5752

La Figura 4.7b muestra el resultado de procesar la imagen “covid3” con el detector de bordes generalizado, se obtuvo un error cuadrático medio de $MSE = 0.0099$. Los parámetros optimizados utilizados fueron: $\alpha = 0.6$, $\beta = 0.975621$, $\gamma = 0.6$, $\xi = 0.9999$, $\eta = 0.6$, $\lambda = 0.6$, $\theta = 3.882284$, $\phi = 2.313351$ y $\sigma = 1.026802$.



(a) Imagen “covid3”.

(b) Imagen procesada

Fig. 4.7. (a) Imagen “covid3”. (b) Imagen procesada con el detector de bordes generalizado.

A continuación en la Figura 4.8 se muestran los resultados al procesar la imagen “covid3” con el detector de bordes de Prewitt, el detector de bordes de Sobel y el detector de bordes de Canny.



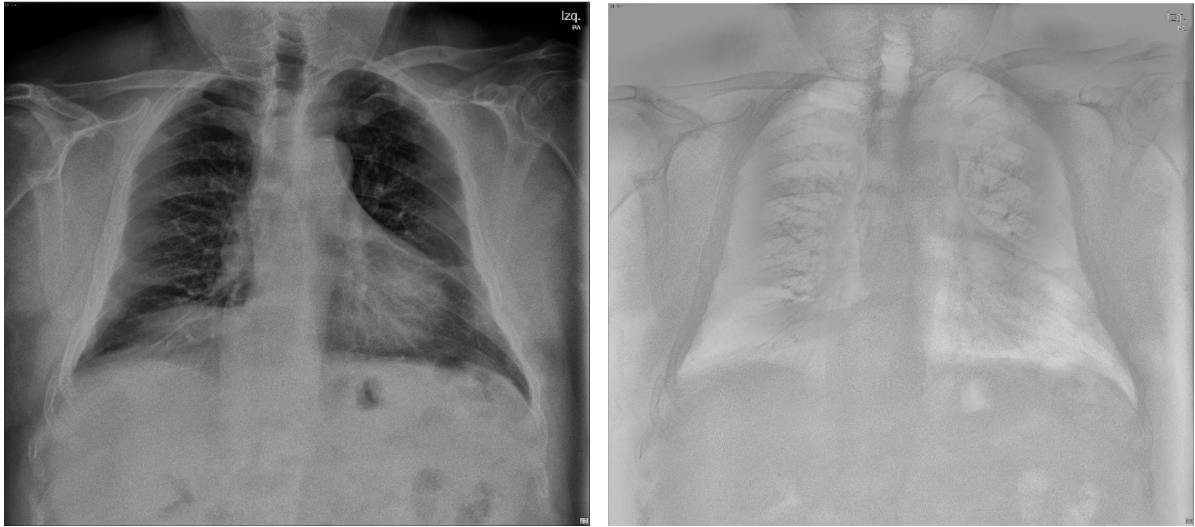
(a) Operador Prewitt

(b) Operador Sobel

(c) Operador Canny

Fig. 4.8. Detectores de bordes clásicos a)Prewitt, b)Sobel, c)Canny

Finalmente se realiza el procedimiento mostrado en el diagrama mostrado en la Figura 3.3. En la Figura 4.9 se presentan los resultados de sumar y restar la imagen “covid3” con el resultado de aplicar el detector de bordes generalizado a dicha imagen,



(a) suma imagen “covid3” con el detector de bordes generalizado. (b) resta imagen “covid3” con el detector de bordes generalizado.

Fig. 4.9. Resultados al sumar y restar la imagen “covid3” con el detector de bordes generalizado

A continuación en la tabla 4.3 se muestran los valores obtenidos al calcular las métricas para la imagen “covid3” obtenida con el detector de bordes generalizado dado en la expresión (2.31), el operador de Prewitt, el operador de Sobel y el algoritmo de Canny.

La entropía de la imagen “covid3” es de 7.6964

Tabla 4.3. Resultados de las métricas de la imagen “covid3”

Operador	Entropía	PSNR	SSIM	BRISQUE
Bordes generalizado	7.5098	20.0479	0.6654	42.0796
Prewitt	0.1195	4.5746	0.0096	48.1872
Sobel	0.1270	4.5809	0.0097	48.7444
Canny	0.6974	5.1052	0.0113	43.4582

La Figura 4.10b muestra el resultado de procesar la imagen “covid4” con el detector de bordes generalizado, se obtuvo un error cuadrático medio de $MSE = 0.0102$. Los parámetros optimizados utilizados fueron: $\alpha = 0.980683$, $\beta = 0.81013$, $\gamma = 0.9999$, $\xi = 0.846122$, $\eta = 0.6$, $\lambda = 0.725368$, $\theta = 5.838555$, $\phi = 3.827927$ y $\sigma = 0.339124$.

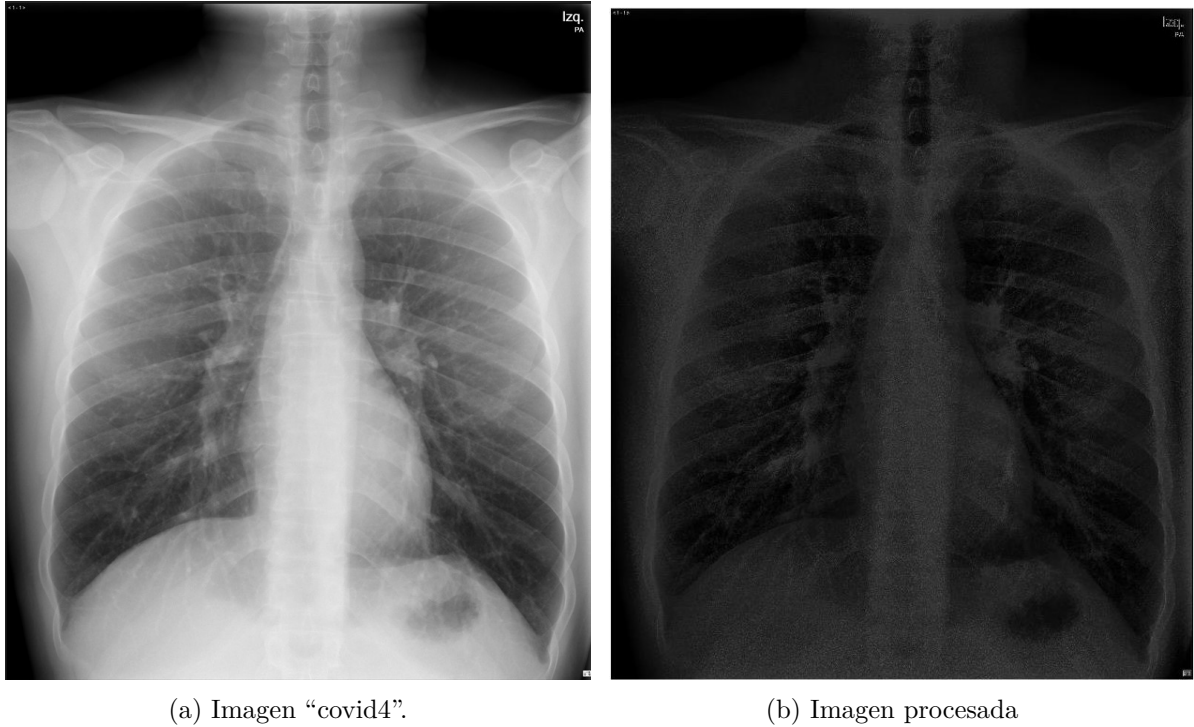


Fig. 4.10. (a) Imagen "covid4". (b) Imagen procesada con el detector de bordes generalizado.

A continuacion en la Figura 4.11 se muestran los resultados al procesar la imagen "covid4" con el detector de bordes de Prewitt, el detector de bordes de Sobel y el detector de bordes de Canny.

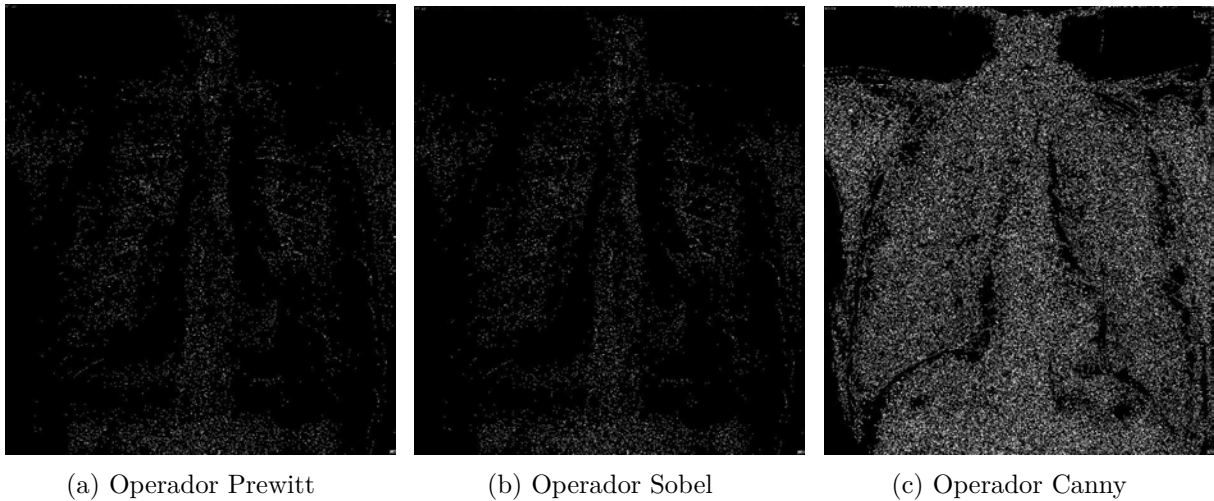
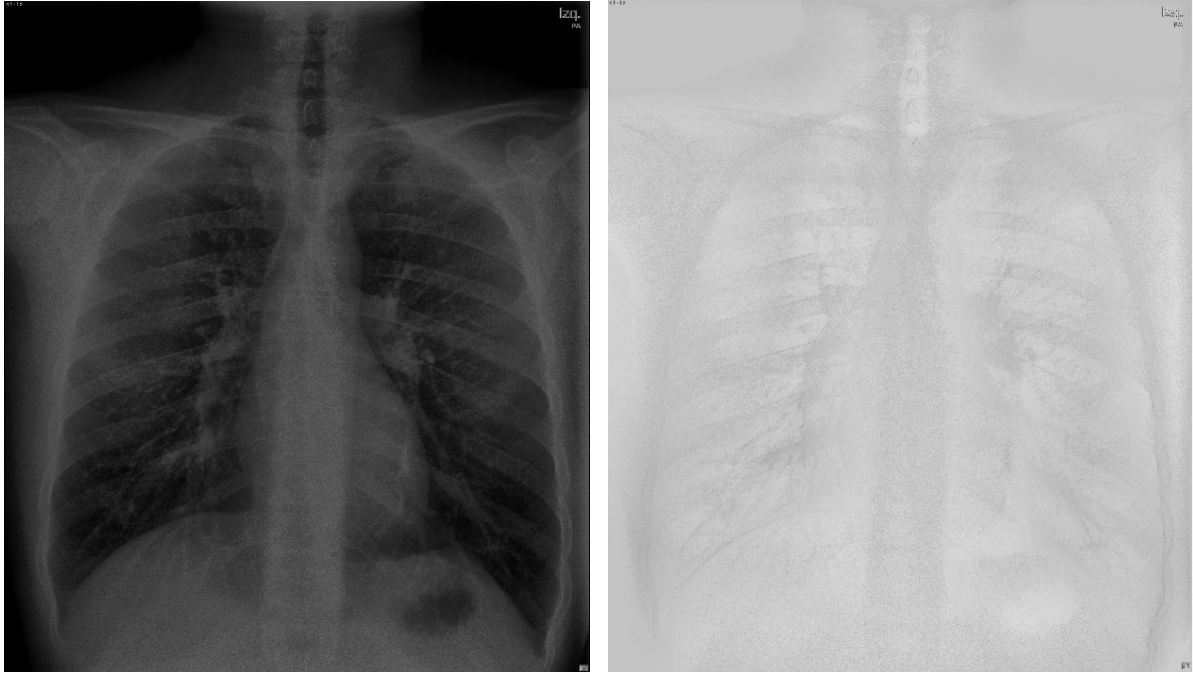


Fig. 4.11. Detectores de bordes clásicos a)Prewitt, b)Sobel, c)Canny

Finalmente se realiza el procedimiento mostrado en en el diagrama mostrado en la Figura 3.3. En la Figura 4.12 se presentan los resultados de sumar y restar la imagen "covid4" con el resultado de aplicar el detector de bordes generalizado a dicha imagen,



(a) suma imagen “covid4” con el detector de bordes generalizado. (b) resta imagen “covid4” con el detector de bordes generalizado.

Fig. 4.12. Resultados al sumar y restar la imagen “covid4” con el detector de bordes generalizado

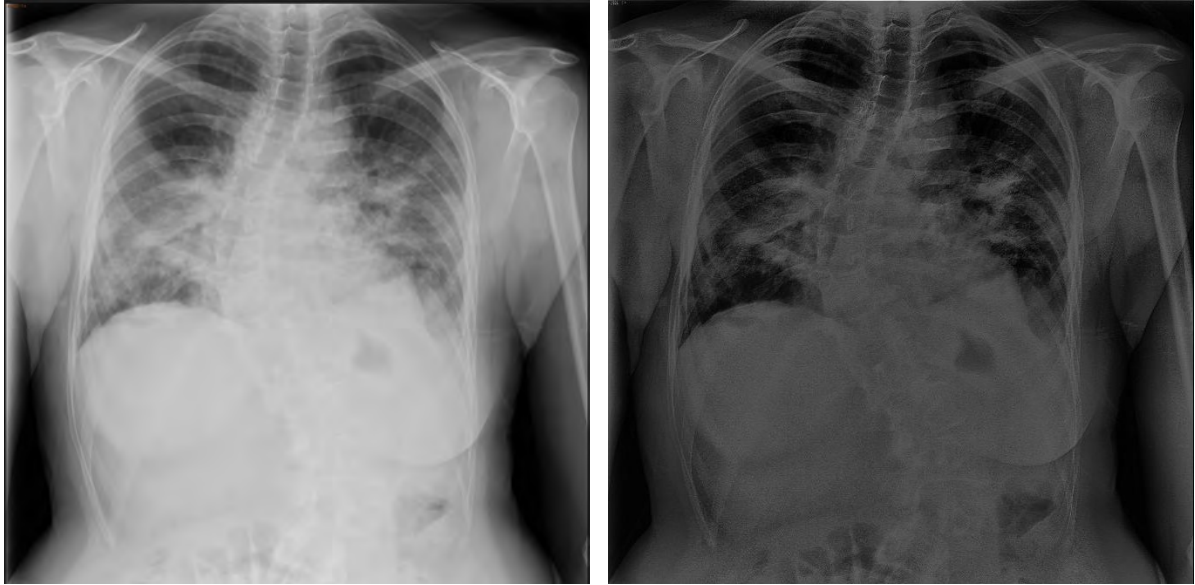
A continuación en la tabla 4.4 se muestran los valores obtenidos al calcular las métricas para la imagen “covid4” obtenida con el detector de bordes generalizado dado en la expresión (2.31), el operador de Prewitt, el operador de Sobel y el algoritmo de Canny.

La entropía de la imagen “covid4” es de 7.7547

Tabla 4.4. Resultados de las métricas de la imagen “covid4”

Operador	Entropía	PSNR	SSIM	BRISQUE
Bordes generalizado	7.6839	19.9080	0.4030	43.4095
Prewitt	0.1121	5.1363	0.0230	47.0059
Sobel	0.1203	5.1438	0.0231	47.5195
Canny	0.6445	5.4051	0.0243	43.4583

La Figura 4.13b muestra el resultado de procesar la imagen “covid5” con el detector de bordes generalizado, se obtuvo un error cuadrático medio de $MSE = 0.0054$. Los parámetros optimizados utilizados fueron: $\alpha = 0.906169$, $\beta = 0.947864$, $\gamma = 0.971699$, $\xi = 0.842331$, $\eta = 0.6$, $\lambda = 0.959676$, $\theta = 4.705024$, $\phi = 5.753433$ y $\sigma = 0.3$.



(a) Imagen "covid5".

(b) Imagen procesada

Fig. 4.13. (a) Imagen "covid5". (b) Imagen procesada con el detector de bordes generalizado.

A continuacion en la Figura 4.14 se muestran los resultados al procesar la imagen "covid5" con el detector de bordes de Prewitt, el detector de bordes de Sobel y el detector de bordes de Canny.



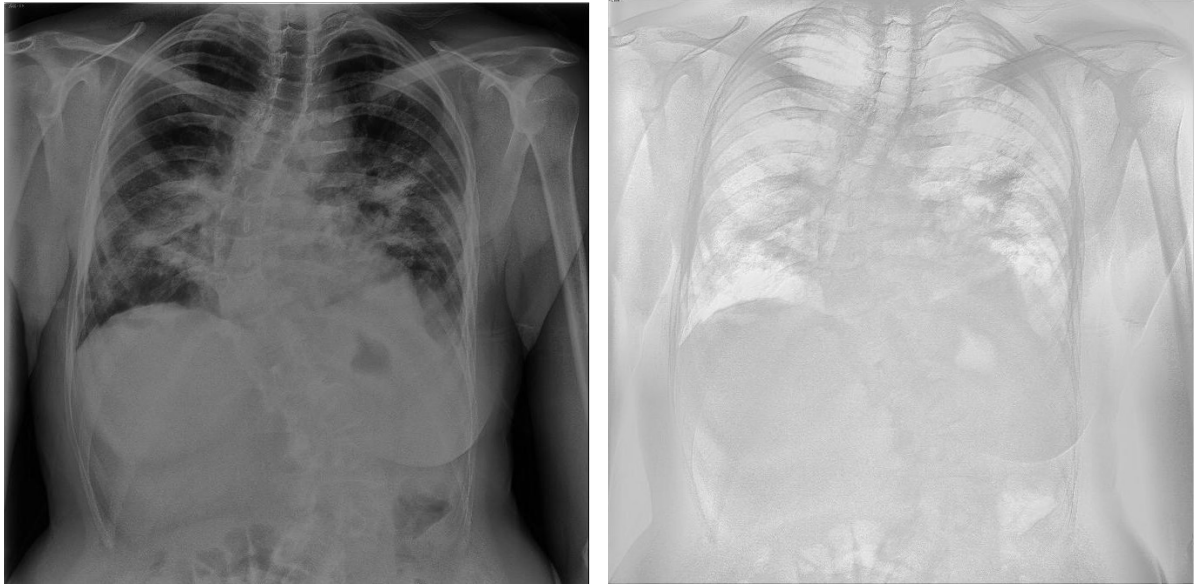
(a) Operador Prewitt

(b) Operador Sobel

(c) Operador Canny

Fig. 4.14. Detectores de bordes clásicos a)Prewitt, b)Sobel, c)Canny

Finalmente se realiza el procedimiento mostrado en en el diagrama mostrado en la Figura 3.3. En la Figura 4.15 se presentan los resultados de sumar y restar la imagen "covid5" con el resultado de aplicar el detector de bordes generalizado a dicha imagen,



(a) suma imagen “covid5” con el detector de bordes generalizado. (b) resta imagen “covid5” con el detector de bordes generalizado.

Fig. 4.15. Resultados al sumar y restar la imagen “covid5” con el detector de bordes generalizado

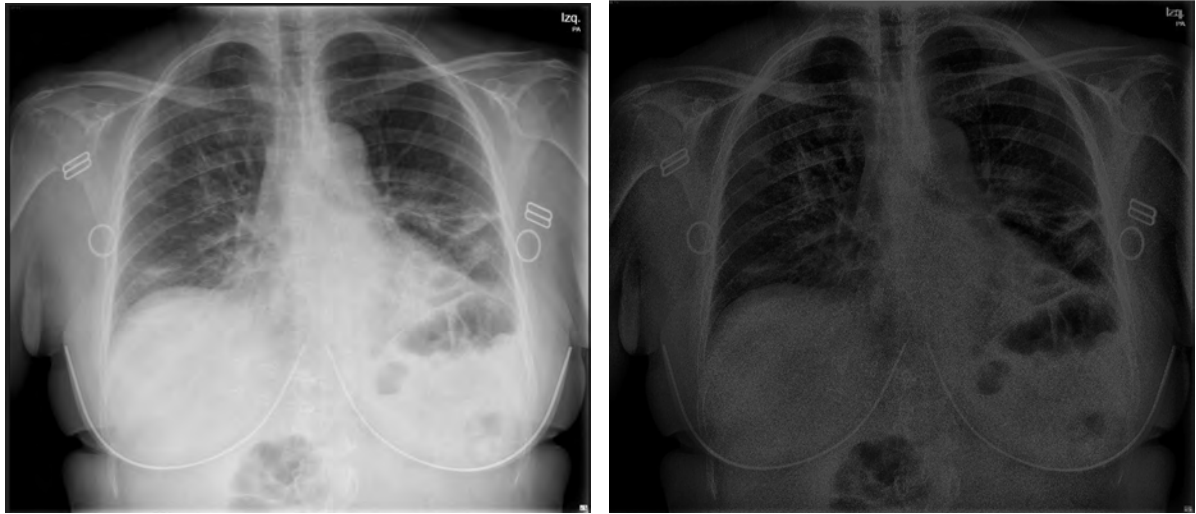
A continuación en la tabla 4.5 se muestran los valores obtenidos al calcular las métricas para la imagen “covid5” obtenida con el detector de bordes generalizado dado en la expresión (2.31), el operador de Prewitt, el operador de Sobel y el algoritmo de Canny.

La entropía de la imagen “covid5” es de 7.5823

Tabla 4.5. Resultados de las métricas de la imagen “covid5”

Operador	Entropía	PSNR	SSIM	BRISQUE
Bordes generalizado	7.6708	22.6821	0.5312	41.6035
Prewitt	0.0578	4.2391	0.0032	44.5088
Sobel	0.0592	4.2396	0.0032	44.5364
Canny	0.5699	4.5335	0.0047	43.5773

La Figura 4.16b muestra el resultado de procesar la imagen “covid6” con el detector de bordes generalizado, se obtuvo un error cuadrático medio de $MSE = 0.0152$. Los parámetros optimizados utilizados fueron: $\alpha = 0.983467$, $\beta = 0.794846$, $\gamma = 0.663303$, $\xi = 0.600153$, $\eta = 0.702917$, $\lambda = 0.725987$, $\theta = 3.141088$, $\phi = 1.306748$ y $\sigma = 0.465154$.



(a) Imagen "covid6".

(b) Imagen procesada

Fig. 4.16. (a) Imagen "covid6". (b) Imagen procesada con el detector de bordes generalizado.

A continuacion en la Figura 4.17 se muestran los resultados al procesar la imagen 4.16a con el detector de bordes de Prewitt, el detector de bordes de Sobel y el detector de bordes de Canny.



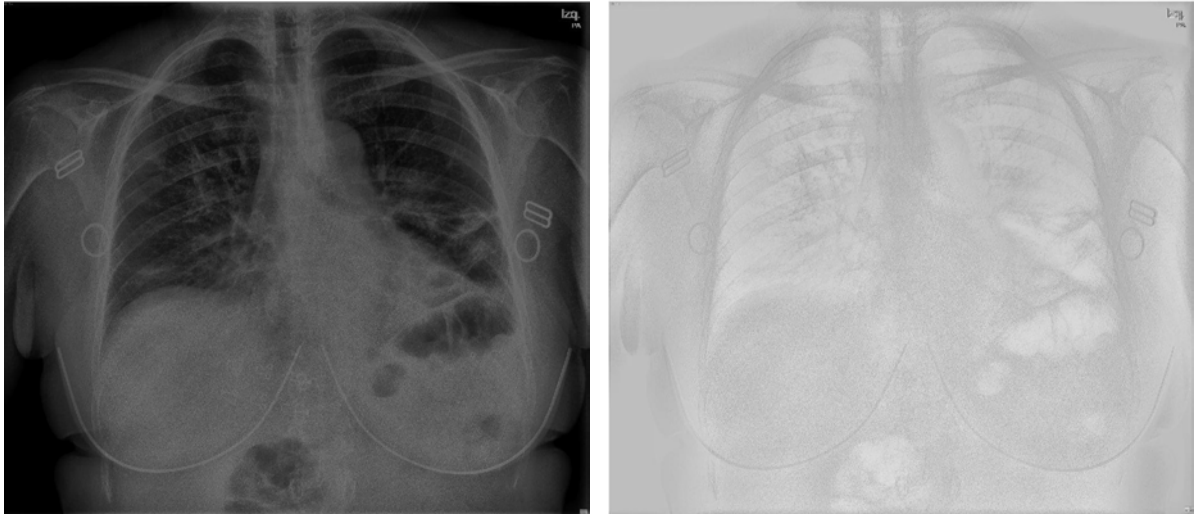
(a) Operador Prewitt

(b) Operador Sobel

(c) Operador Canny

Fig. 4.17. Detectores de bordes clásicos a)Prewitt, b)Sobel, c)Canny

Finalmente se realiza el procedimiento mostrado en en el diagrama mostrado en la Figura 3.3. En la Figura 4.18 se presentan los resultados de sumar y restar la imagen "covid6" con el resultado de aplicar el detector de bordes generalizado a dicha imagen,



(a) suma imagen “covid6” con el detector de bordes generalizado. (b) resta imagen “covid6” con el detector de bordes generalizado.

Fig. 4.18. Resultados al sumar y restar la imagen “covid6” con el detector de bordes generalizado

A continuación en la tabla 4.6 se muestran los valores obtenidos al calcular las métricas para la imagen “covid6” obtenida con el detector de bordes generalizado dado en la expresión (2.31), el operador de Prewitt, el operador de Sobel y el algoritmo de Canny.

La entropía de la imagen “covid6” es de 7.7183

Tabla 4.6. Resultados de las métricas de la imagen “covid6”

Operador	Entropía	PSNR	SSIM	BRISQUE
Bordes generalizado	7.7252	18.1854	0.3484	43.4199
Prewitt	0.1514	4.9355	0.0206	48.7944
Sobel	0.1610	4.9458	0.0207	49.2233
Canny	0.6500	5.3075	0.0219	43.4584

4.1.3. Análisis de resultados

De los resultados anteriores se observa que, al procesar imágenes médicas con el detector de bordes generalizado, se obtienen mejores resultados en las métricas en comparación con los detectores de bordes clásicos. En general, la entropía de todas las imágenes fue mayor a 7.5, lo que indica una buena cantidad de detalles y variaciones en los niveles de gris, cercana al valor máximo. El PSNR para todas las imágenes fue superior a 17, lo cual sugiere una calidad de imagen moderada. Por otro lado, el SSIM mostró valores promedio alrededor de 0.3, indicando que las imágenes presentaron distorsión. El BRISQUE, cuya interpretación es inversa (valores más bajos indican mejor calidad), fue en general más bajo con el detector de bordes generalizado en comparación con los resultados obtenidos utilizando los detectores de bordes clásicos.

4.2. Gradiente Gaussiano \mathcal{M} -truncado fraccionario de orden variable aplicado a imágenes médicas

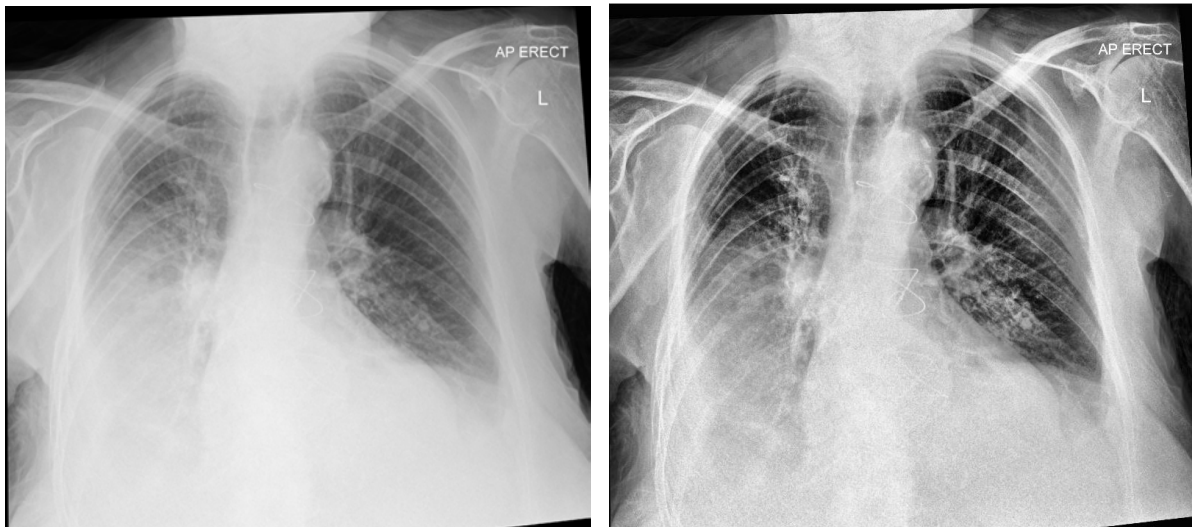
Para utilizar el detector de bordes mostrado en la expresión (3.16), es necesario solo optimizar σ que es la desviación estándar ya que α , β y γ son funciones suaves. La optimización de este parámetros se llevó a cabo mediante el algoritmo de PSO con el objetivo de encontrar el valor optimo de σ .

4.2.1. Función objetivo

La función objetivo usada se define como una función que toma dos argumentos: un vector de parámetros \mathbf{x} (σ) y una imagen la cual es la imagen que se va a procesar. Esta función evalúa el error cuadrático medio (MSE) del modelo de regresión lineal, utilizando el parámetro \mathbf{x} (σ) y la imagen. El objetivo es minimizar este error cuadrático medio, ajustando σ obtener la mejor aproximación posible a los datos de la imagen.

4.2.2. Resultados

La Figura (4.19b) muestra el resultado de procesar la imagen “Lung abscess and pleural empyema” con el detector de bordes con parámetros variables, obteniéndose un error cuadrático medio de $MSE = 0.000066$. Los parámetros utilizados fueron: $\alpha = \beta = \gamma = \tanh(t)$ y el valor optimizado de $\sigma = 1.124970$.



(a) Imagen "Lung abscess and pleural empyema"

(b) Imagen procesada

Fig. 4.19. (a) Imagen “Lung abscess and pleural empyema”. (b) Imagen “Lung abscess and pleural empyema” procesada con el detector de bordes con parámetros variables.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.20 se presentan los resultados de sumar y restar la imagen “Lung abscess and pleural empyema” con el resultado de aplicar el detector de bordes con parámetros variables a dicha imagen.

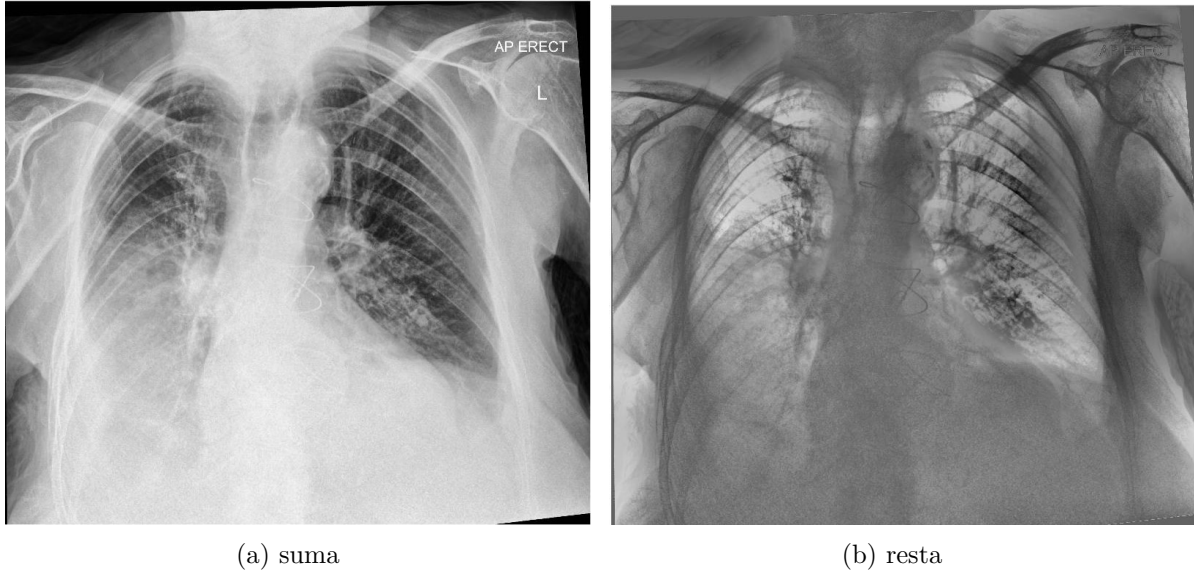


Fig. 4.20. (a) suma de la imagen “Lung abscess and pleural empyema” con la imagen procesada con el detector con parámetros variables. (b) resta de la imagen “Lung abscess and pleural empyema” con la imagen procesada con el detector de bordes con parámetros variables.

La Figura (4.21b) muestra el resultado de procesar la imagen “Lung abscess and pleural empyema” con el detector de bordes con parámetros variables, obteniéndose un error cuadrático medio de $MSE = 0.000032$. Los parámetros utilizados fueron: $\alpha = \beta = \gamma = \frac{1}{1+e^{-t}}$ y el valor optimizado de $\sigma = 1.254747$.

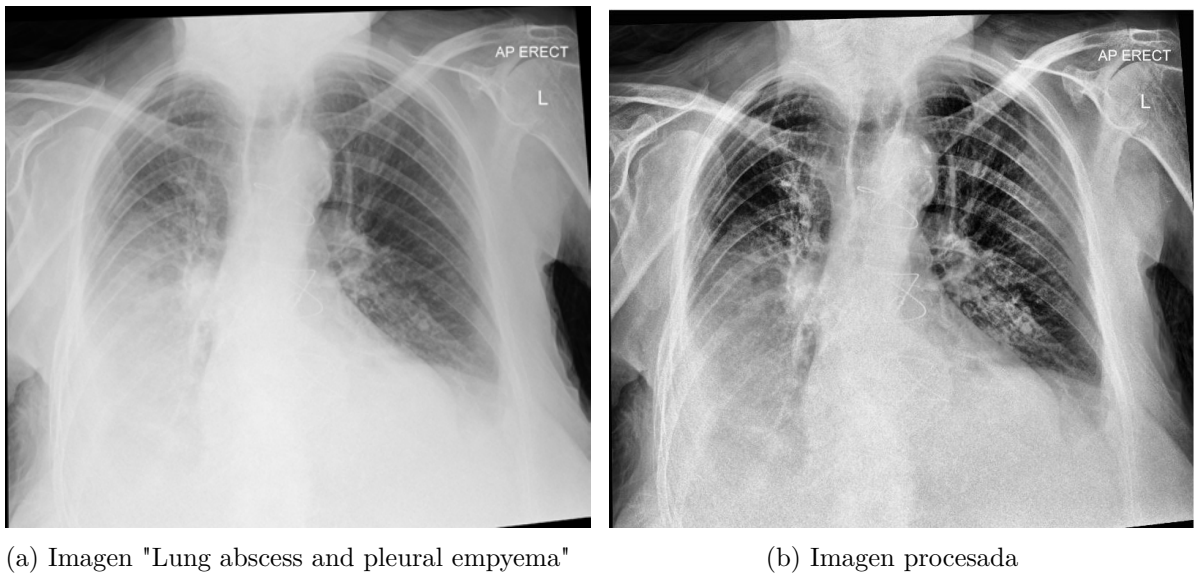


Fig. 4.21. (a) Imagen “Lung abscess and pleural empyema”. (b) Imagen “Lung abscess and pleural empyema” procesada con el detector de bordes con parámetros variables.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.22 se presentan los resultados de sumar y restar la imagen “Lung abscess and pleural empyema” con el

resultado de aplicar el detector de bordes con parámetros variables a dicha imagen.

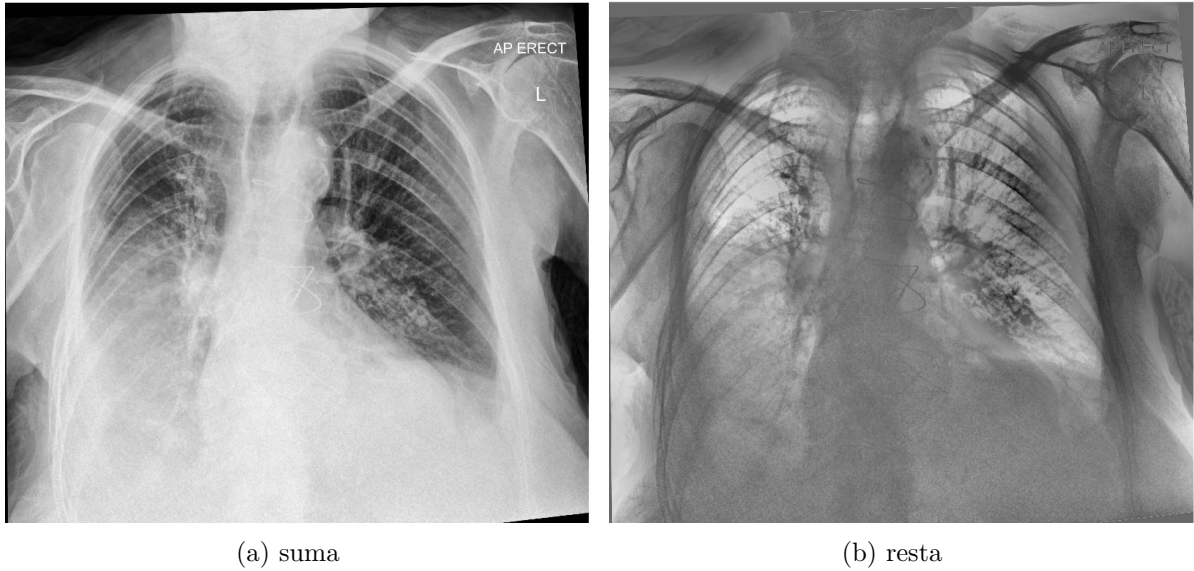


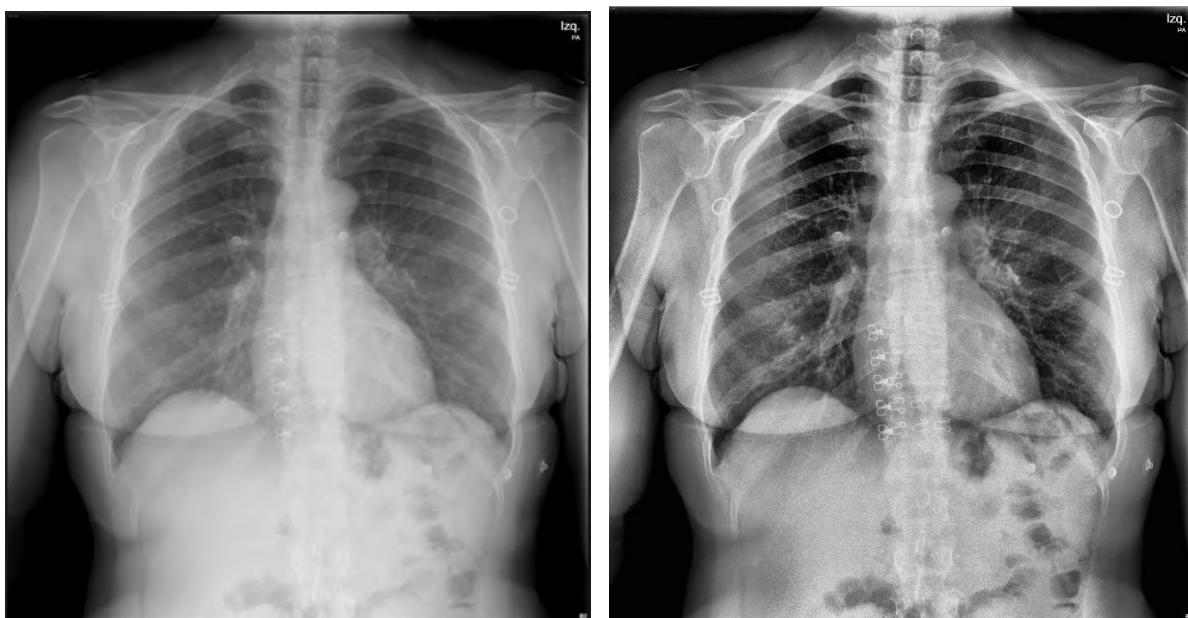
Fig. 4.22. (a) suma de la imagen “Lung abscess and pleural empyema” con el detector de bordes con parámetros variables. (b) resta de la imagen “Lung abscess and pleural empyema” con el detector de bordes con parámetros variables.

En la Tabla 4.7, se muestran los valores obtenidos al calcular las métricas para la imagen médica “Lung abscess and pleural empyema” procesada con el detector de bordes con parámetros variables usando las funciones suaves tangente hiperbólica y sigmoideal.

Tabla 4.7. Resultados de las métricas de la imagen “Lung abscess and pleural empyema” procesada con funciones suaves

Operador	Entropía	PSNR	SSIM	BRISQUE
Función tangente hiperbólica	7.5481	39.1227	0.9640	44.0855
Función sigmoideal	7.5562	41.1036	0.9706	44.0151

La Figura (4.23b) muestra el resultado de procesar la imagen “covid2” con el detector de bordes con parámetros variables, obteniéndose un error cuadrático medio de $MSE = 0.000184$. Los parámetros utilizados fueron: $\alpha = \beta = \gamma = \tanh(t)$ y el valor optimizado de $\sigma = 1.128368$.



(a) Imagen covid2"

(b) Imagen procesada

Fig. 4.23. (a) Imagen "covid2". (b) Imagen "covid2" procesada con el detector de bordes con parámetros variables.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.24 se presentan los resultados de sumar y restar la imagen "covid2" con el resultado de aplicar el detector de bordes con parámetros variables a dicha imagen.

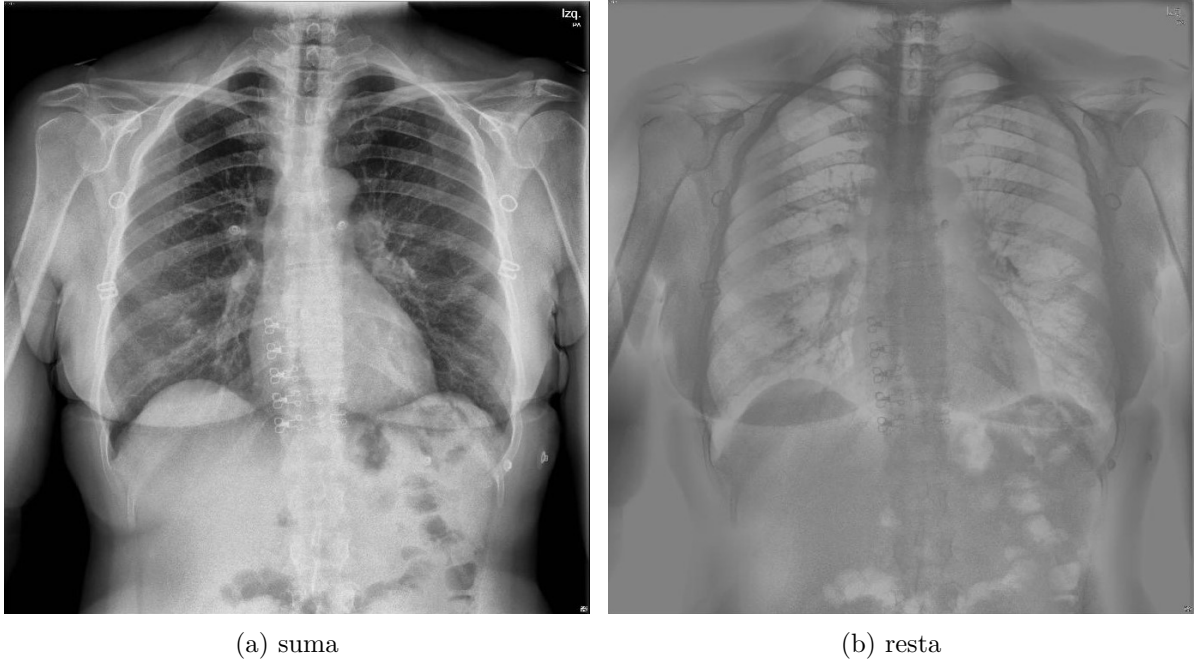
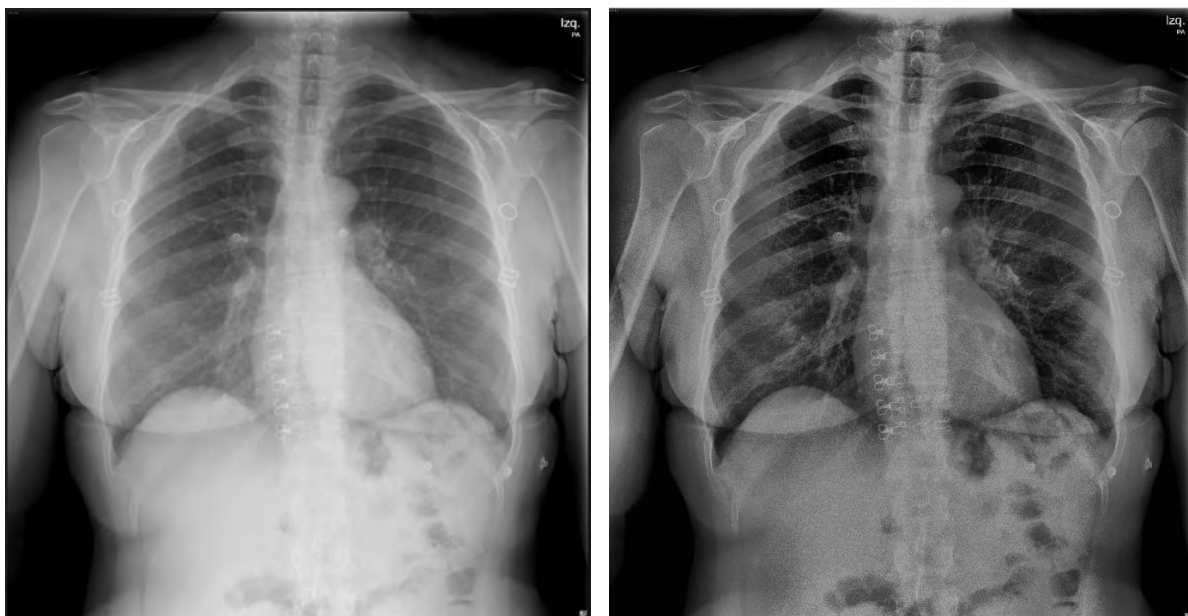


Fig. 4.24. (a) suma de la imagen “covid2” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid2” con el detector de bordes con parámetros variables.

La Figura (4.25b) muestra el resultado de procesar la imagen “covid2” con el detector de bordes con parámetros variables, obteniéndose un error cuadrático medio de $MSE = 0.000424$. Los parámetros utilizados fueron: $\alpha = \beta = \gamma = \frac{1}{1+e^{-t}}$ y el valor optimizado de $\sigma = 0.374089$.



(a) Imagen covid2"

(b) Imagen procesada

Fig. 4.25. (a) Imagen "covid2". (b) Imagen "covid2" procesada con el detector de bordes con parámetros variables.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.26 se presentan los resultados de sumar y restar la imagen "covid2" con el resultado de aplicar el detector de bordes con parámetros variables a dicha imagen.

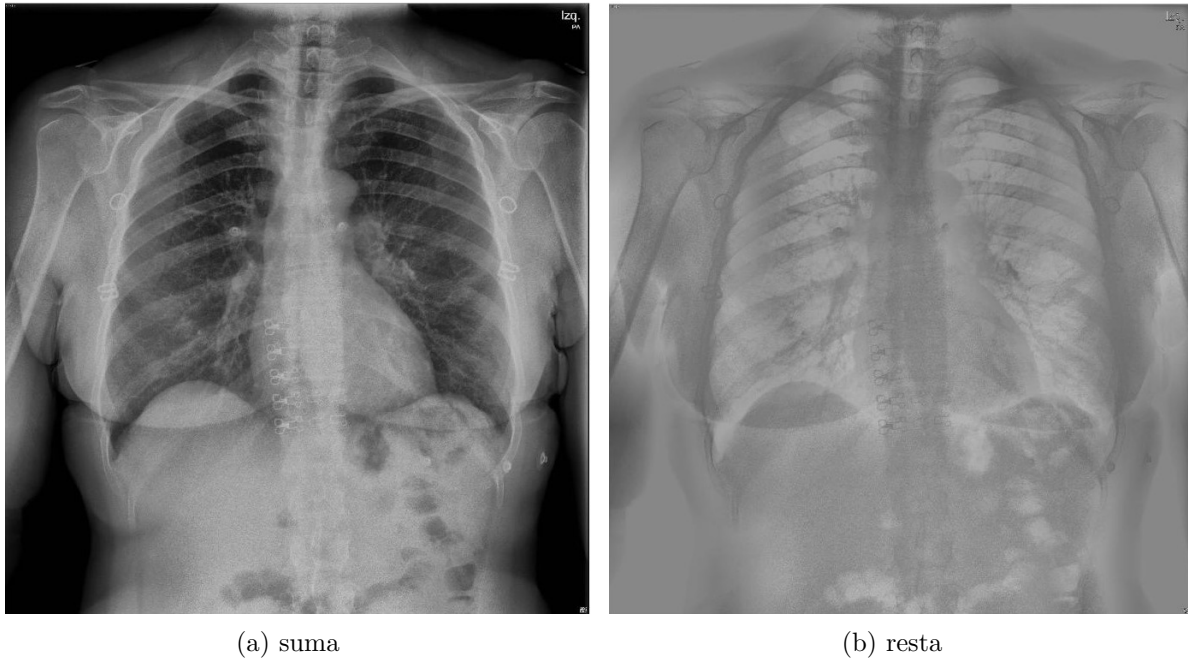


Fig. 4.26. (a) suma de la imagen “covid2” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid2” con el detector de bordes con parámetros variables.

En la Tabla 4.8, se muestran los valores obtenidos al calcular las métricas para la imagen médica “covid2” procesada con el detector de bordes con parámetros variables usando las funciones suaves tangente hiperbólica y sigmooidal.

Tabla 4.8. Resultados de las métricas de la imagen “covid2” procesada con funciones suaves

Operador	Entropía	PSNR	SSIM	BRISQUE
Función tangente hiperbólica	7.6071	30.6381	0.7969	26.3234
Función sigmooidal	7.6248	28.9416	0.8077	38.4863

La Figura (4.27b) muestra el resultado de procesar la imagen “covid3” con el detector de bordes con parámetros variables, obteniéndose un error cuadrático medio de $MSE = 0.000219$. Los parámetros utilizados fueron: $\alpha = \beta = \gamma = \tanh(t)$ y el valor optimizado de $\sigma = 1.138389$.

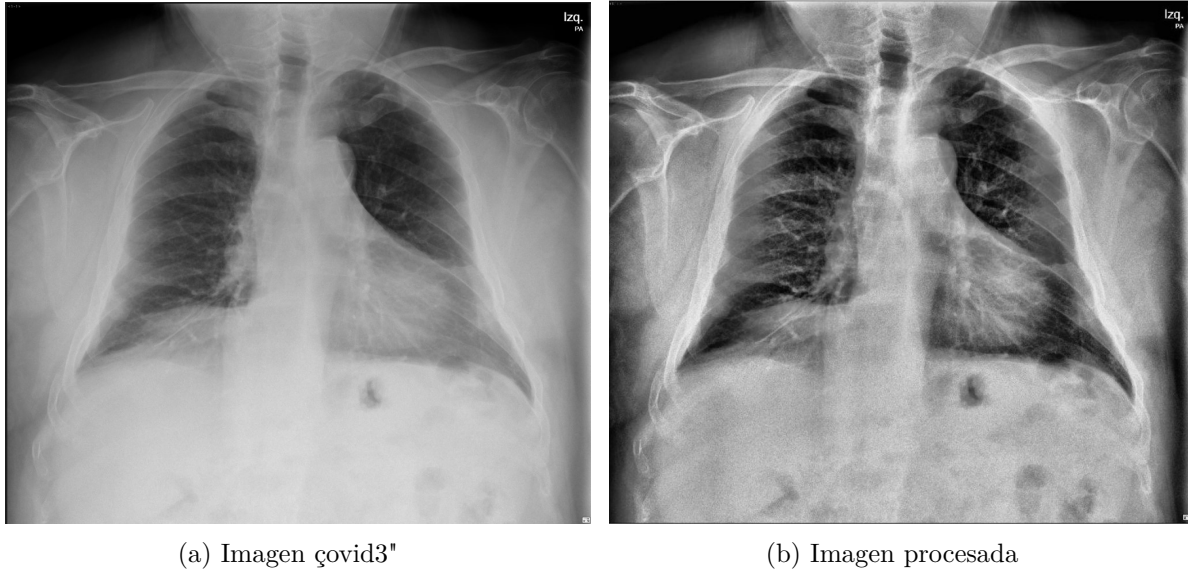


Fig. 4.27. (a) suma de la imagen "covid3" con el detector de bordes con parámetros variables. (b) resta de la imagen "covid3" con el detector de bordes con parámetros variables.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.28 se presentan los resultados de sumar y restar la imagen "covid3" con el resultado de aplicar el detector de bordes con parámetros variables a dicha imagen.

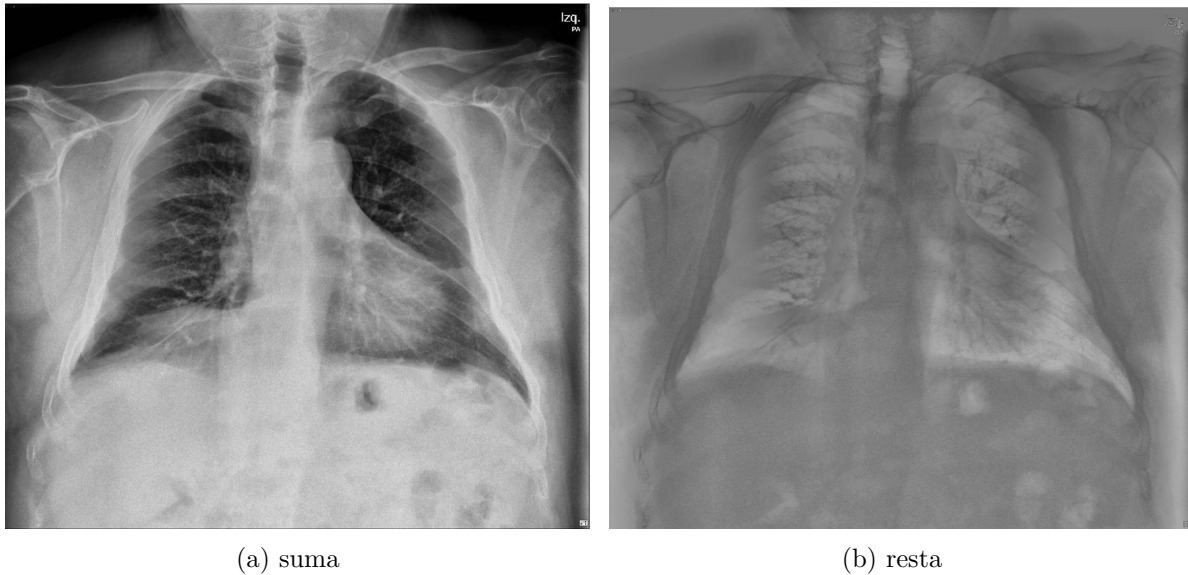


Fig. 4.28. (a) suma de la imagen "covid3" con el detector de bordes con parámetros variables. (b) resta de la imagen "covid3" con el detector de bordes con parámetros variables.

La Figura (4.29b) muestra el resultado de procesar la imagen "covid3" con el detector de bordes con parámetros variables, obteniéndose un error cuadrático medio de $MSE = 0.000205$. Los parámetros utilizados fueron: $\alpha = \beta = \gamma = \frac{1}{1+e^{-t}}$ y el valor optimizado de $\sigma = 1.254771$.

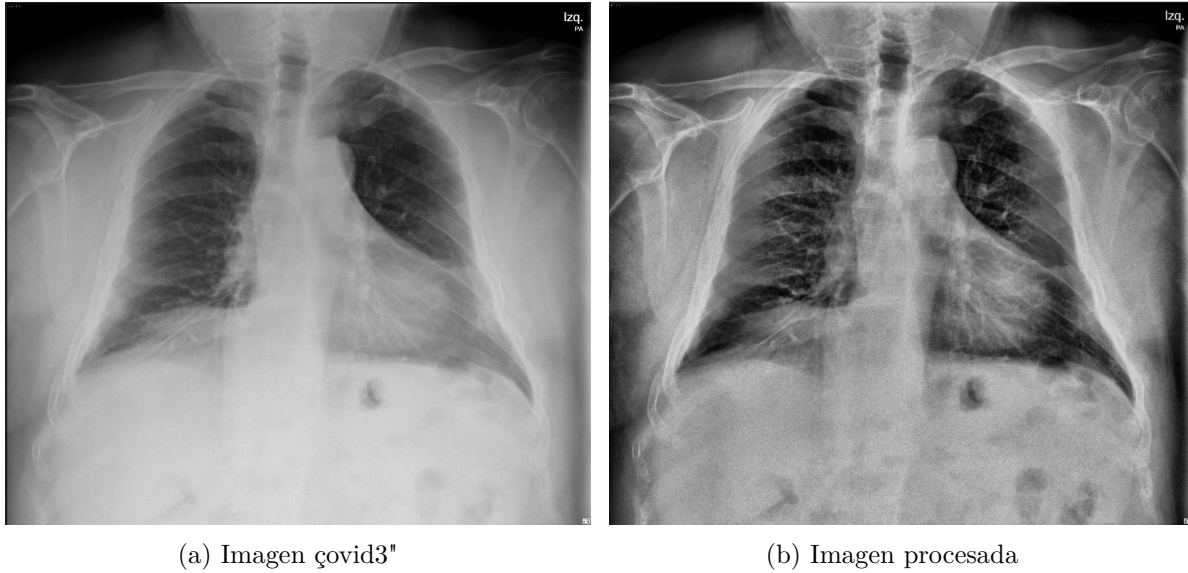


Fig. 4.29. (a) suma de la imagen "covid3" con el detector de bordes con parámetros variables. (b) resta de la imagen "covid3" con el detector de bordes con parámetros variables.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.28 se presentan los resultados de sumar y restar la imagen "covid3" con el resultado de aplicar el detector de bordes con parámetros variables a dicha imagen.

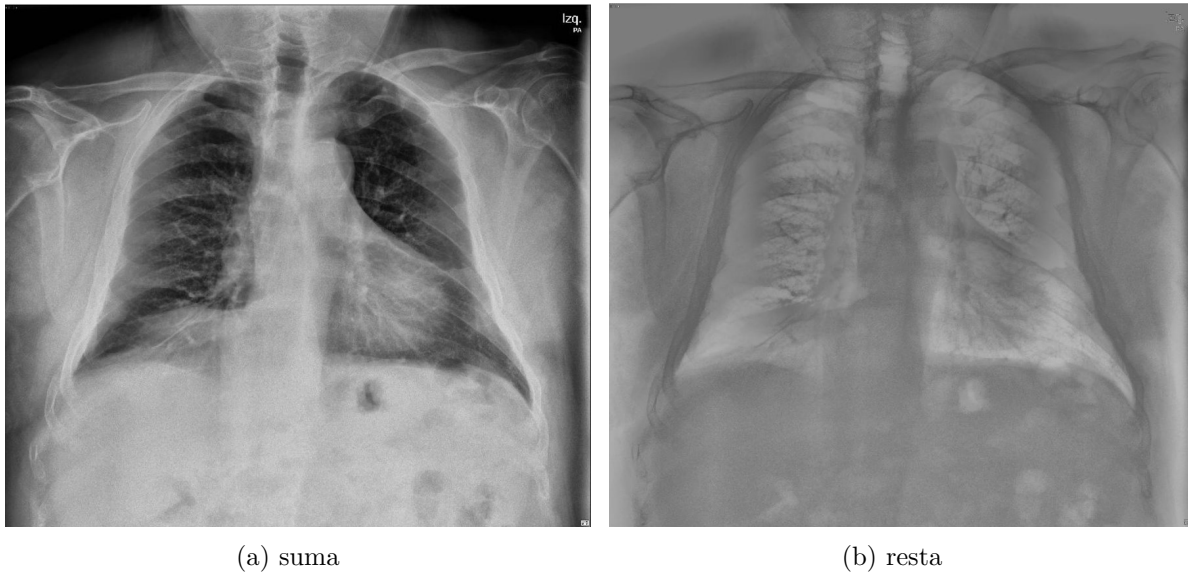


Fig. 4.30. (a) suma de la imagen "covid3" con el detector de bordes con parámetros variables. (b) resta de la imagen "covid3" con el detector de bordes con parámetros variables.

En la Tabla 4.9, se muestran los valores obtenidos al calcular las métricas para la imagen médica "covid3" procesada con el detector de bordes con parámetros variables usando las funciones suaves tangente hiperbólica y sigmoïdal.

Tabla 4.9. Resultados de las métricas de la imagen “covid3” procesada con funciones suaves

Operador	Entropía	PSNR	SSIM	BRISQUE
Función tangente hiperbolica	7.6399	29.3872	0.7381	23.2248
Función sigmoïdal	7.6388	29.8795	0.7689	23.0223

La Figura (4.31b) muestra el resultado de procesar la imagen “covid4” con el detector de bordes con parámetros variables, obteniéndose un error cuadrático medio de $MSE = 0.000189$. Los parámetros utilizados fueron: $\alpha = \beta = \gamma = \tanh(t)$ y el valor optimizado de $\sigma = 1.128408$.

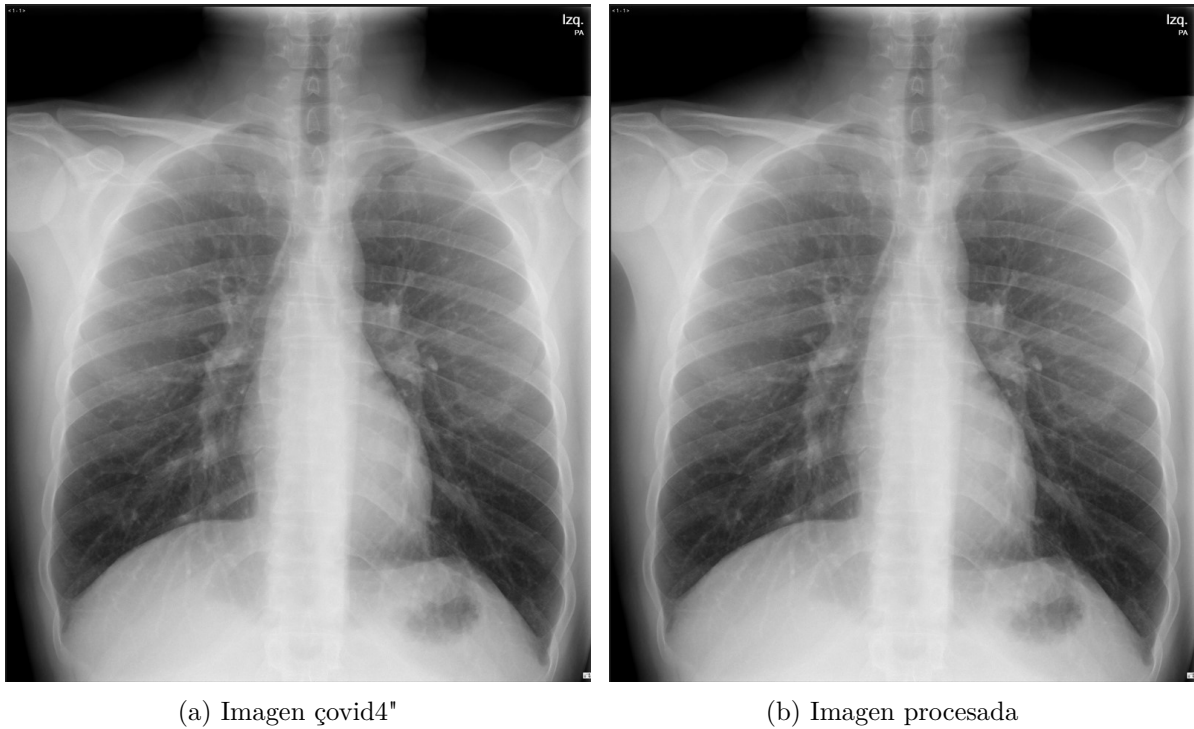


Fig. 4.31. (a) Imagen “covid4”. (b) Imagen “covid4” procesada con el detector de bordes con parámetros variables.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.32 se presentan los resultados de sumar y restar la imagen “covid4” con el resultado de aplicar el detector de bordes con parámetros variables a dicha imagen.

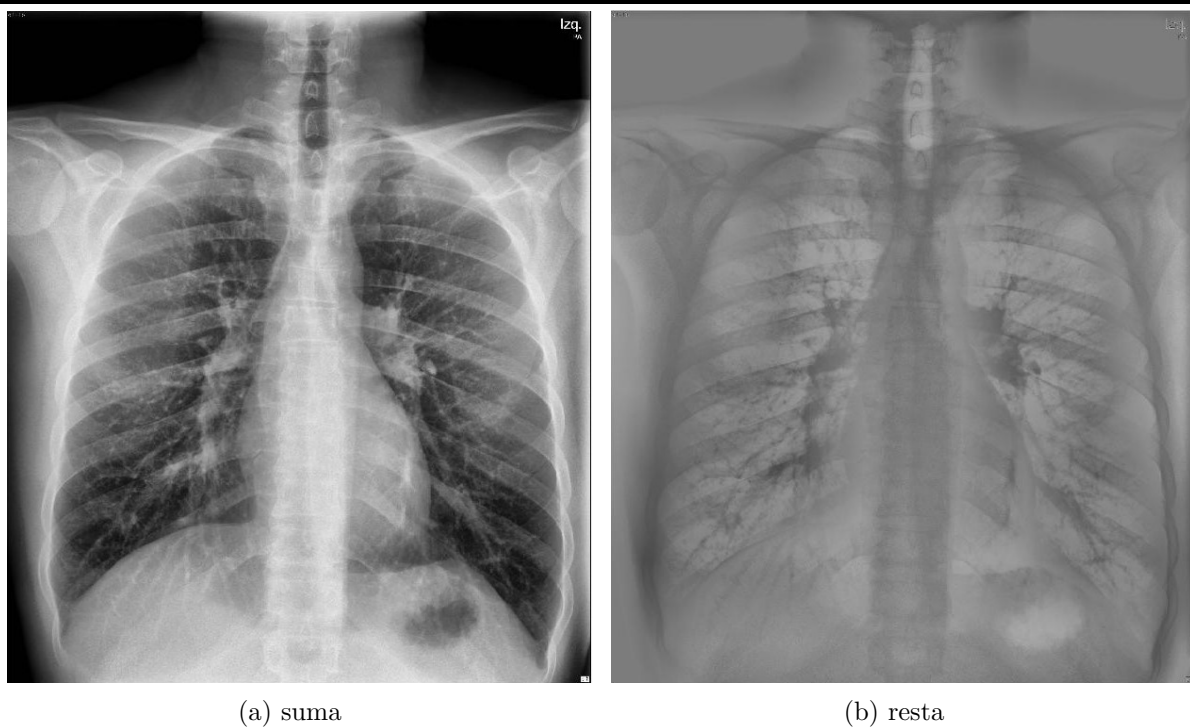
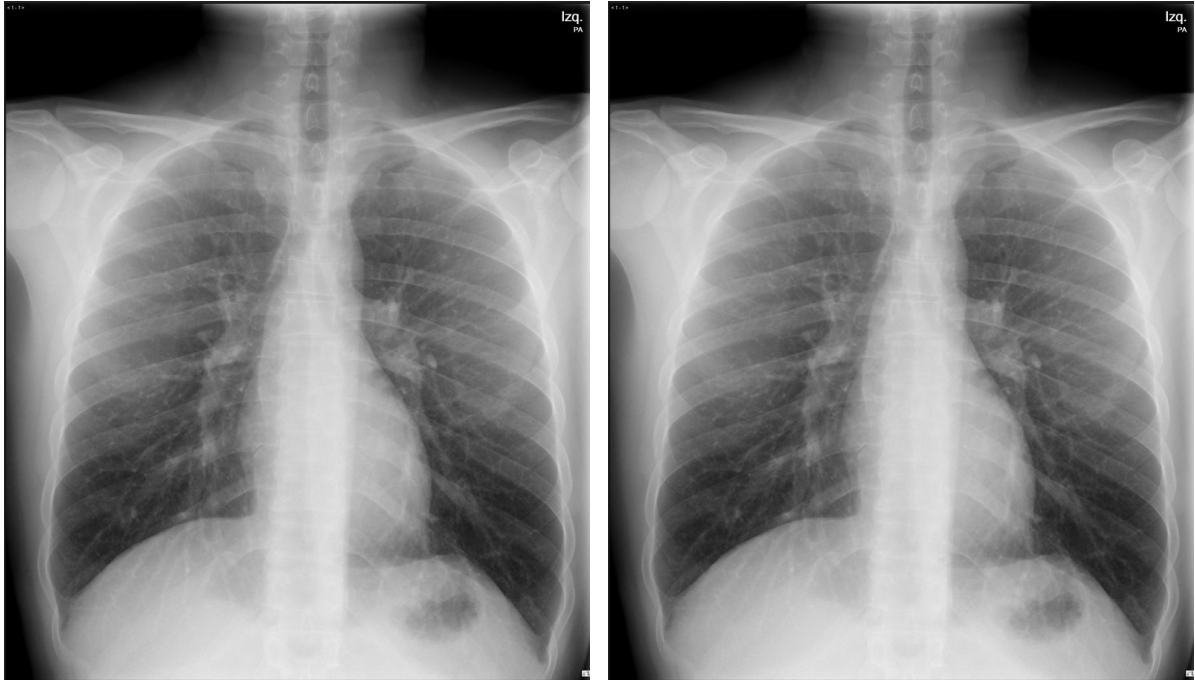


Fig. 4.32. (a) suma de la imagen “covid4” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid4” con el detector de bordes con parámetros variables.

La Figura (4.33b) muestra el resultado de procesar la imagen “covid4” con el detector de bordes con parámetros variables, obteniéndose un error cuadrático medio de $MSE = 0.000219$. Los parámetros utilizados fueron: $\alpha = \beta = \gamma = \frac{1}{1+e^{-t}}$ y el valor optimizado de $\sigma = 1.128389$.



(a) Imagen covid4"

(b) Imagen procesada

Fig. 4.33. (a) suma de la imagen "covid4" con el detector de bordes con parámetros variables. (b) resta de la imagen "covid4" con el detector de bordes con parámetros variables.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.34 se presentan los resultados de sumar y restar la imagen "covid4" con el resultado de aplicar el detector de bordes con parámetros variables a dicha imagen.

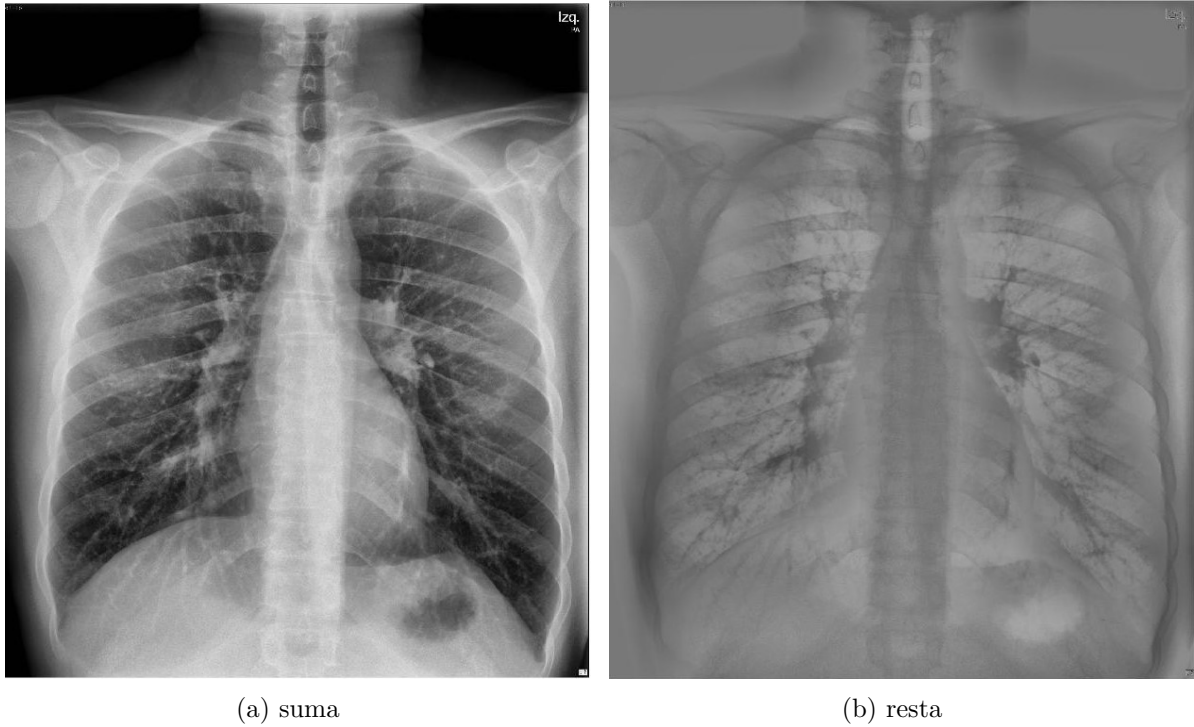


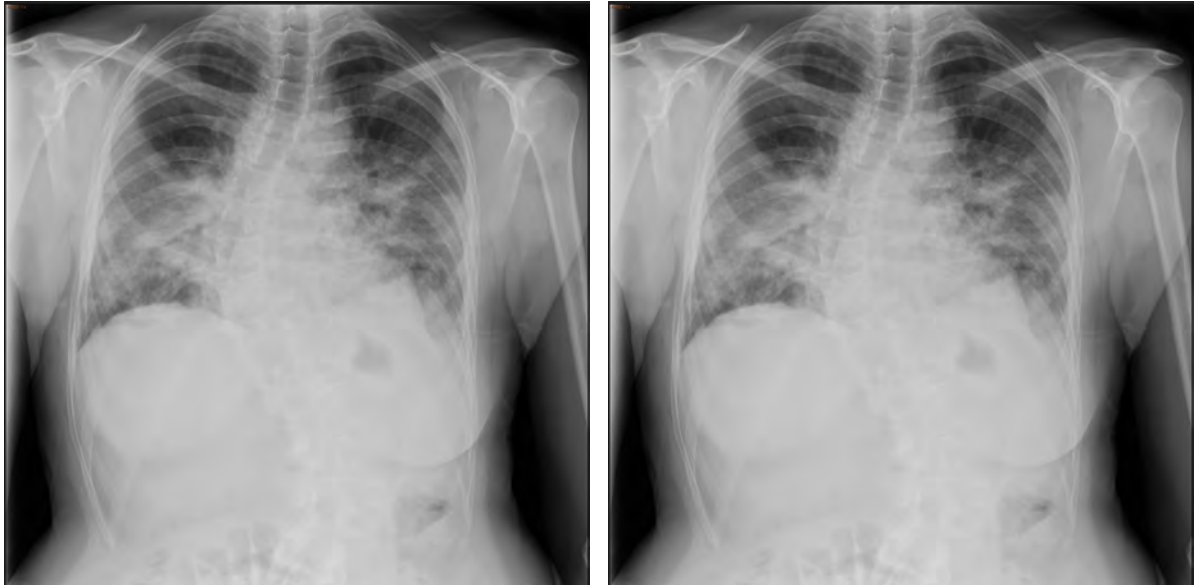
Fig. 4.34. (a) suma de la imagen “covid4” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid4” con el detector de bordes con parámetros variables.

En la Tabla 4.10, se muestran los valores obtenidos al calcular las métricas para la imagen médica “covid4” procesada con el detector de bordes con parámetros variables usando las funciones suaves tangente hiperbólica y sigmooidal.

Tabla 4.10. Resultados de las métricas de la imagen “covid4” procesada con funciones suaves

Operador	Entropía	PSNR	SSIM	BRISQUE
Función tangente hiperbólica	7.7430	31.5068	0.7973	30.4194
Función sigmooidal	7.6399	29.3872	0.7381	23.2248

La Figura (4.35b) muestra el resultado de procesar la imagen “covid5” con el detector de bordes con parámetros variables, obteniéndose un error cuadrático medio de $MSE = 0.000124$. Los parámetros utilizados fueron: $\alpha = \beta = \gamma = \tanh(t)$ y el valor optimizado de $\sigma = 1.128346$.

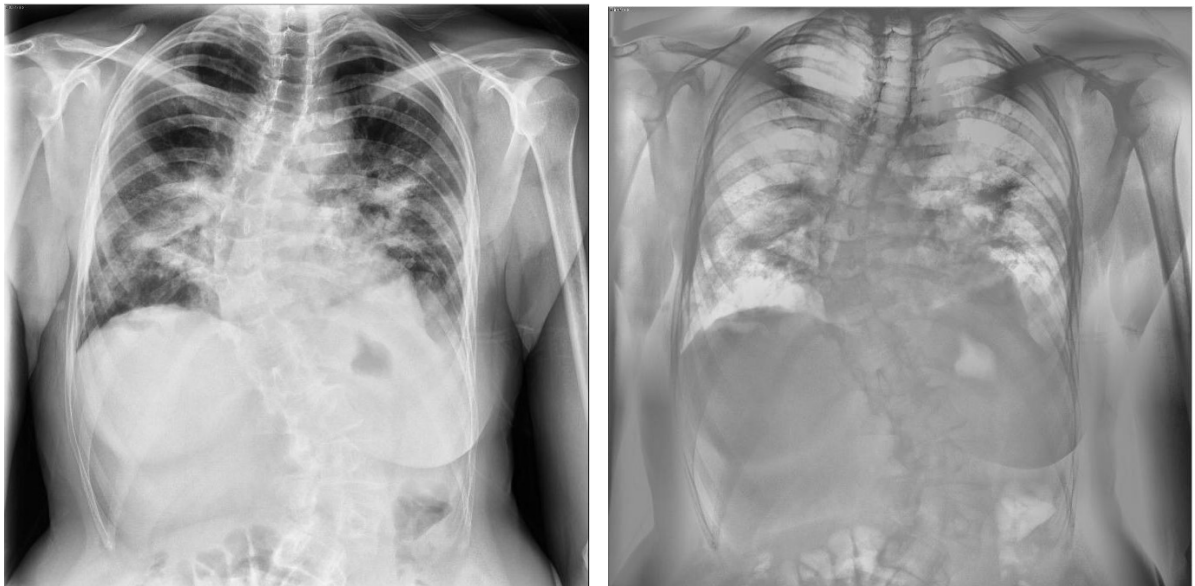


(a) Imagen covid5"

(b) Imagen procesada

Fig. 4.35. (a) Imagen "covid5". (b) Imagen "covid5" procesada con el detector de bordes con parámetros variables.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.36 se presentan los resultados de sumar y restar la imagen "covid5" con el resultado de aplicar el detector de bordes con parámetros variables a dicha imagen.



(a) suma

(b) resta

Fig. 4.36. (a) suma de la imagen "covid5" con el detector de bordes con parámetros variables. (b) resta de la imagen "covid5" con el detector de bordes con parámetros variables.

CAPÍTULO 4. GRADIENTE GAUSSIANO M -TRUNCADO FRACCIONARIO APLICADO A
IMÁGENES DIGITALES

La Figura (4.37b) muestra el resultado de procesar la imagen “covid5” con el detector de bordes con parámetros variables, obteniéndose un error cuadrático medio de $MSE = 0.000138$. Los parámetros utilizados fueron: $\alpha = \beta = \gamma = \frac{1}{1+e^{-t}}$ y el valor optimizado de $\sigma = 1.254726$.

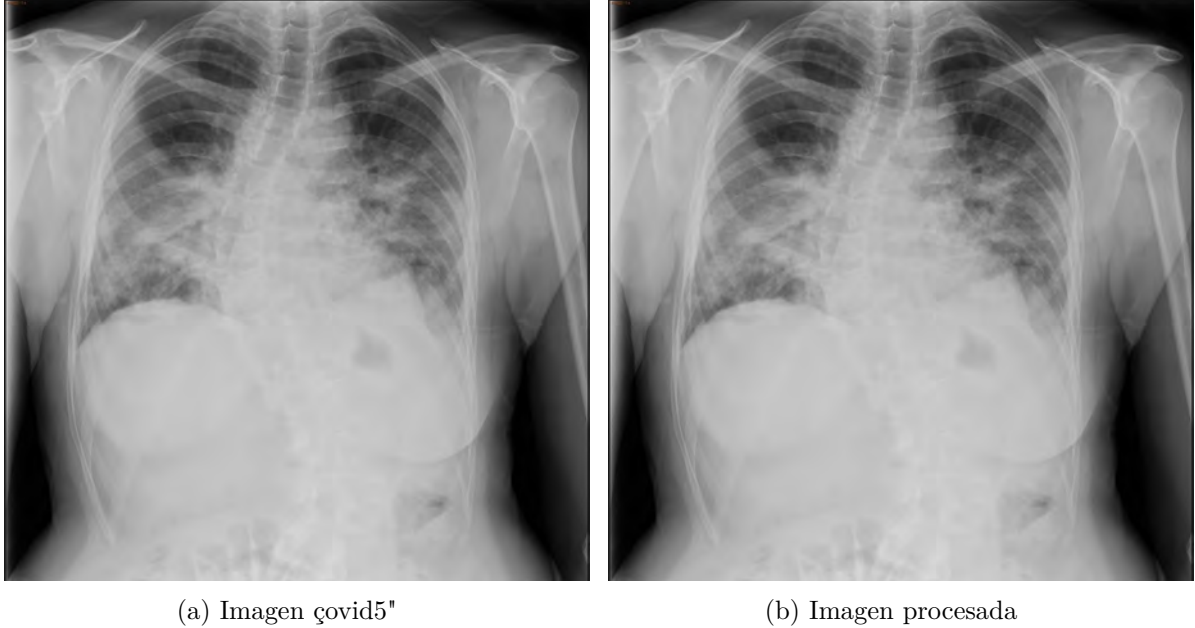
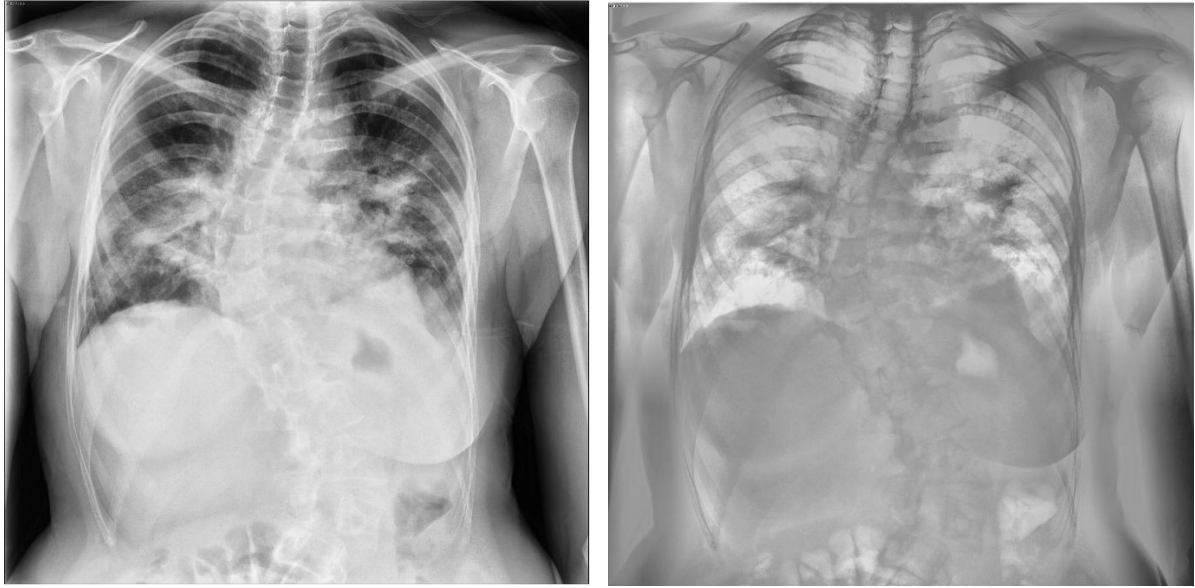


Fig. 4.37. (a) Imagen “covid5”. (b) Imagen “covid4” procesada con el detector de bordes con parámetros variables.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.38 se presentan los resultados de sumar y restar la imagen “covid5” con el resultado de aplicar el detector de bordes con parámetros variables a dicha imagen.



(a) suma

(b) resta

Fig. 4.38. (a) suma de la imagen “covid5” con el detector de bordes con parámetros variables. (b) resta de la imagen “covid5” con el detector de bordes con parámetros variables.

En la Tabla 4.11, se muestran los valores obtenidos al calcular las métricas para la imagen médica “covid5” procesada con el detector de bordes con parámetros variables usando las funciones suaves tangente hiperbólica y sigmooidal.

Tabla 4.11. Resultados de las métricas de la imagen “covid5” procesada con funciones suaves

Operador	Entropía	PSNR	SSIM	BRISQUE
Función tangente hiperbólica	7.5643	35.3178	0.9250	28.5703
Función sigmooidal	7.5640	35.2179	0.9349	28.7894

La Figura (4.39b) muestra el resultado de procesar la imagen “covid6” con el detector de bordes con parámetros variables, obteniéndose un error cuadrático medio de $MSE = 0.000317$. Los parámetros utilizados fueron: $\alpha = \beta = \gamma = \tanh(t)$ y el valor optimizado de $\sigma = 1.128370$.

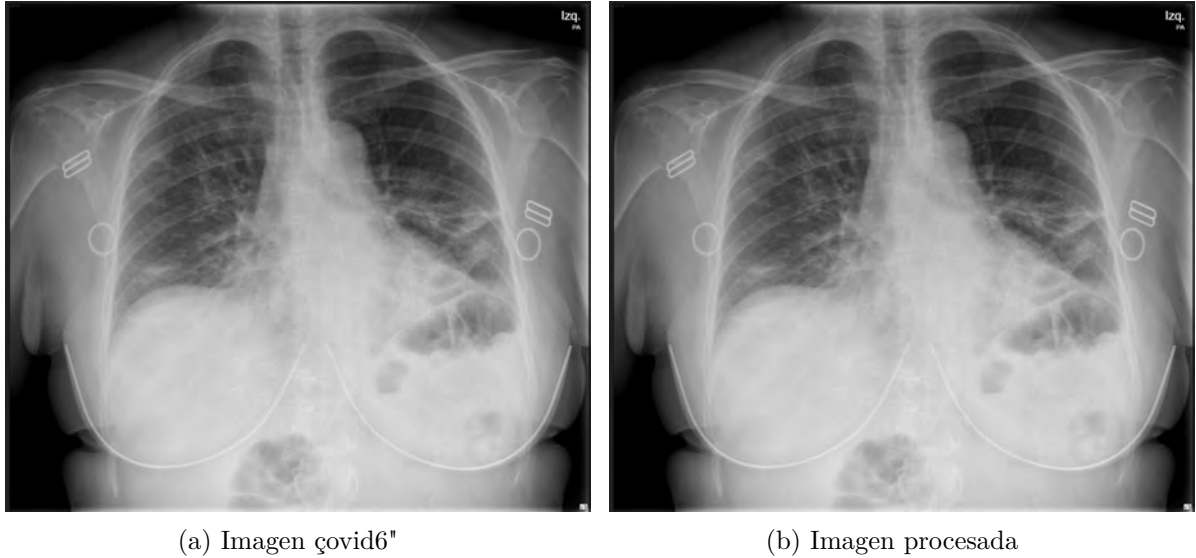


Fig. 4.39. (a) Imagen "covid6". (b) Imagen "covid6" procesada con el detector de bordes con parámetros variables.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.40 se presentan los resultados de sumar y restar la imagen "covid6" con el resultado de aplicar el detector de bordes con parámetros variables a dicha imagen.

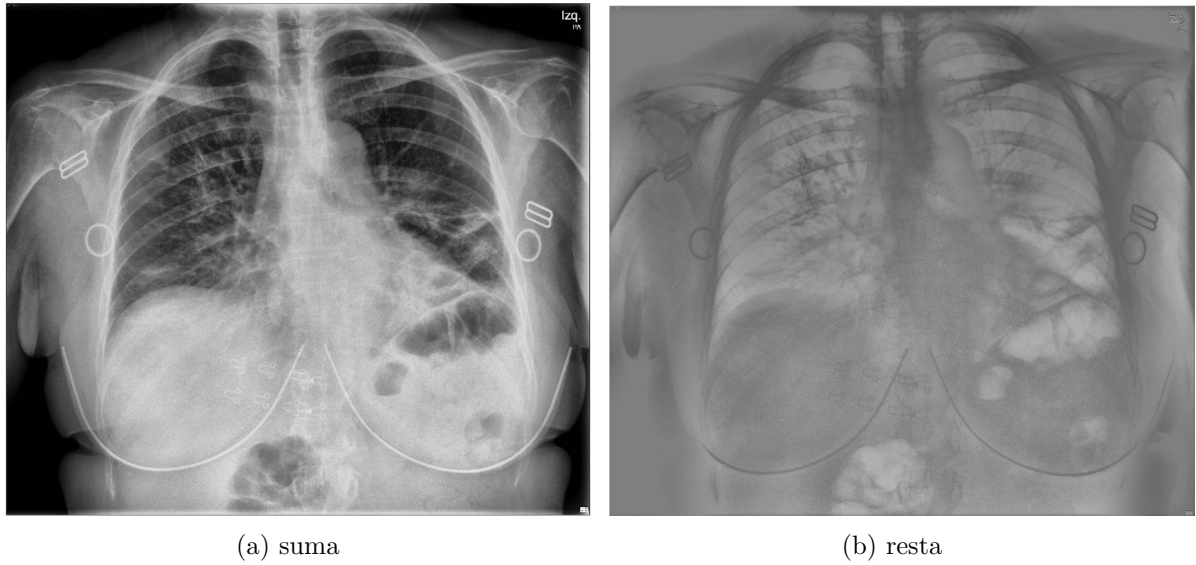


Fig. 4.40. (a) suma de la imagen "covid6" con el detector de bordes con parámetros variables. (b) resta de la imagen "covid6" con el detector de bordes con parámetros variables.

La Figura (4.41b) muestra el resultado de procesar la imagen "covid6" con el detector de bordes con parámetros variables, obteniéndose un error cuadrático medio de $MSE = 0.000290$. Los parámetros utilizados fueron: $\alpha = \beta = \gamma = \frac{1}{1+e^{-t}}$ y el valor optimizado de $\sigma = 1.254764$.

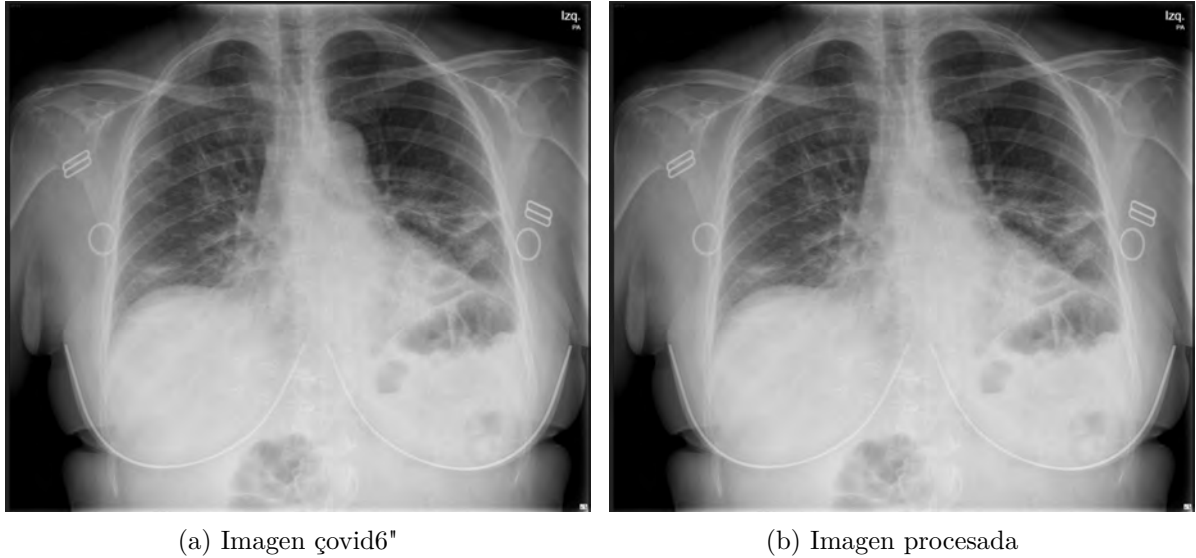


Fig. 4.41. (a) suma de la imagen "covid6" con el detector de bordes con parámetros variables. (b) resta de la imagen "covid6" con el detector de bordes con parámetros variables.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.42 se presentan los resultados de sumar y restar la imagen "covid6" con el resultado de aplicar el detector de bordes con parámetros variables a dicha imagen.

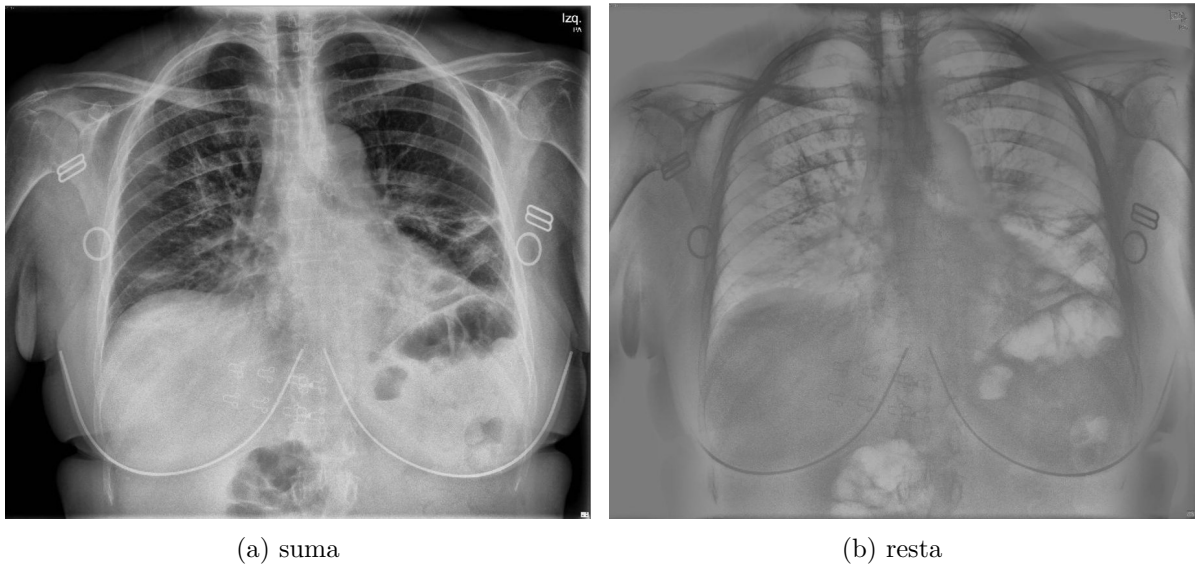


Fig. 4.42. (a) suma de la imagen "covid6" con el detector de bordes con parámetros variables. (b) resta de la imagen "covid6" con el detector de bordes con parámetros variables.

En la Tabla 4.12, se muestran los valores obtenidos al calcular las métricas para la imagen médica "covid6" procesada con el detector de bordes con parámetros variables usando las funciones suaves tangente hiperbolica y sigmoideal.

Tabla 4.12. Resultados de las métricas de la imagen “covid6” procesada con funciones suaves

Operador	Entropía	PSNR	SSIM	BRISQUE
Función tangente hiperbolica	7.6776	26.9868	0.6621	29.7055
Función sigmoideal	7.6762	27.4468	0.6985	28.8371

4.2.3. Análisis de resultados

Tabla 4.13. Comparación de resultados

Figura	Método	Entropía	PSNR	SSIM	BRISQUE
covid1	Bordes generalizado	7.6243	28.5218	0.7261	37.7641
	Función tangente hiperbolica	7.5481	39.1227	0.9640	44.0855
	Función sigmoideal	7.5562	41.1036	0.9706	44.0151
Covid2	Bordes generalizado	7.6248	28.9416	0.8077	38.4863
	Función tangente hiperbolica	7.6071	30.6381	0.7969	26.3234
	Función sigmoideal	7.6246	29.9416	0.8079	38.4869
Covid3	Bordes generalizado	7.5098	20.0479	0.6654	42.0796
	Función tangente hiperbolica	7.6399	29.3872	0.7381	23.2248
	Función sigmoideal	7.6388	29.8795	0.7689	23.0223
Covid4	Bordes generalizado	7.6839	19.9080	0.4030	43.4095
	Función tangente hiperbolica	7.7430	31.5068	0.7973	30.4194
	Función sigmoideal	7.6399	29.3872	0.7381	23.2248
Covid5	Bordes generalizado	7.6708	22.6821	0.5312	41.6035
	Función tangente hiperbolica	7.5643	35.3178	0.9250	28.5703
	Función sigmoideal	7.5640	35.2179	0.9349	28.7894
Covid6	Bordes generalizado	7.7252	18.1854	0.3484	43.4199
	Función tangente hiperbolica	7.6776	26.9868	0.6621	29.7055
	Función sigmoideal	7.6762	27.4468	0.6985	28.8371

Los resultados anteriores indican que, al aplicar el detector de bordes con órdenes variables a imágenes médicas, se logran mejores métricas en comparación con los detectores de bordes tradicionales. En general, la entropía de todas las imágenes fue mayor a 7.5, lo que indica una buena cantidad de detalles y variaciones en los niveles de gris, cercana al valor máximo. El PSNR para todas las imágenes fue superior a 25, lo cual sugiere una calidad de imagen moderada. Por otro lado, el SSIM mostró valores superiores a 0.7, inclusive en algunas imágenes llegó a 0.9 lo cual indica una compresión moderada, indicando que las imágenes no presentaron distorsión. El BRISQUE, cuya interpretación es inversa (valores más bajos indican mejor calidad), fue en general más bajo con el detector de bordes generalizado en comparación con los resultados obtenidos utilizando los detectores de bordes clásicos.

4.3. Matriz Hessiana

La matriz Hessiana se utiliza para calcular la curvatura principal que define los bordes y sus características. En este contexto, λ_1 y λ_2 representan los valores máximos y mínimos, respectivamente, conocidos también como los eigenvalores de la matriz Hessiana. Debido a la falta de contraste en λ_2 , se elige λ_1 para los objetivos de esta investigación.

4.3.1. Función objetivo

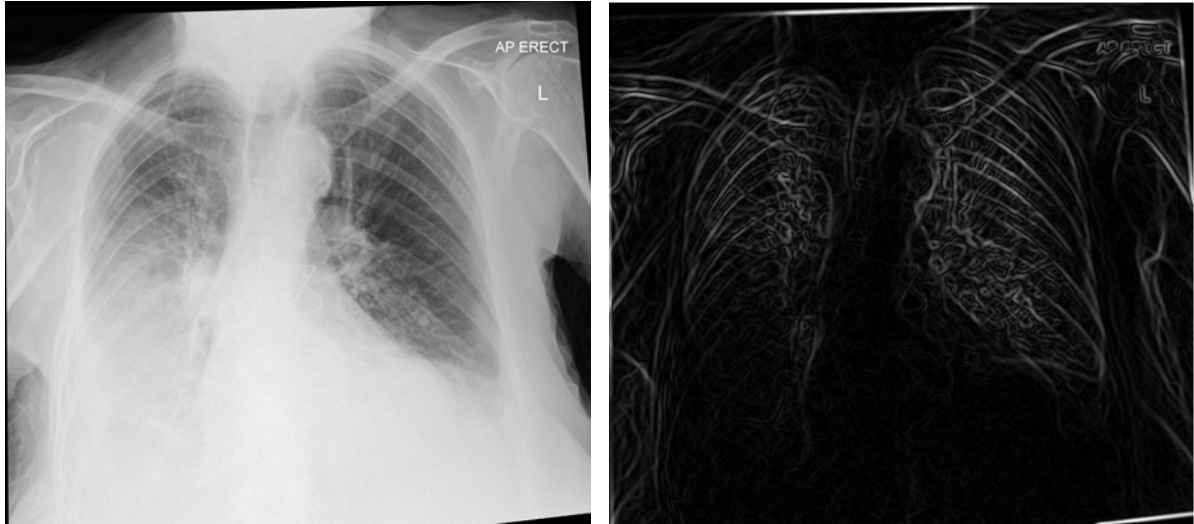
La función objetivo empleada se define como una función que recibe dos argumentos: un vector de parámetros \mathbf{x} (que incluye α , β , γ y σ) y una imagen que será procesada. Esta función calcula el error cuadrático medio (MSE) del modelo de regresión lineal, utilizando los parámetros \mathbf{x} y la imagen. El propósito es minimizar el MSE ajustando los parámetros del modelo para lograr la mejor aproximación posible a los datos de la imagen.

Una vez obtenidos los parámetros optimizados, se procesa la imagen con el detector de bordes (2.33) y se aplica la curvatura principal. Estos resultados se obtienen combinando la imagen original con el resultado de aplicar la curvatura principal a la imagen en cuestión, mediante sumas y restas, como se muestra en el diagrama de bloques de la Figura 3.3.

4.3.2. Resultados

En las siguientes imágenes, se presentan los resultados obtenidos al aplicar la curvatura principal a las imágenes médicas..

La Figura 4.43b presenta los resultados de procesar la imagen “Lung abscess and pleural empyema” utilizando el detector de bordes clásico definido en la expresión (2.33), obteniendo un error cuadrático medio de $MSE = 0.000338$. Los parámetros optimizados utilizados fueron: $\alpha = 0.4$, $\beta = 0.396084$, $\gamma = 0.9999$ y $\sigma = 0.504228$. Luego, se aplicó la curvatura principal.



(a) Imagen "Lung abscess and pleural empyema"

(b) Imagen procesada

Fig. 4.43. (a) Imagen "Lung abscess and pleural empyema". (b) Imagen "Lung abscess and pleural empyema" procesada con la curvatura principal.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.44 se presentan los resultados de sumar y restar la imagen "Lung abscess and pleural empyema" con el resultado de aplicar el detector de bordes clásico y la curvatura principal a dicha imagen.



(a) suma

(b) resta

Fig. 4.44. (a) suma de la imagen "Lung abscess and pleural empyema" con el detector de bordes clásico y la curvatura principal. (b) resta de la imagen "Lung abscess and pleural empyema" con el detector de bordes clásico y la curvatura principal.

La Figura 4.45b presenta los resultados de procesar la imagen "covid2" utilizando el detector de bordes clásico definido en la expresión (2.33), obteniendo un error cuadrático medio de $MSE = 0.000335$. Los

parámetros optimizados utilizados fueron: $\alpha = 0.391222$, $\beta = 0.228885$, $\gamma = 0.9999$ y $\sigma = 0.504623$. Luego, se aplicó la curvatura principal.



(a) Imagen "covid2"

(b) Imagen procesada

Fig. 4.45. (a) Imagen "covid2". (b) Imagen "covid2" procesada con la curvatura principal.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.46 se presentan los resultados de sumar y restar la imagen "covid2" con el resultado de aplicar el detector de bordes clásico y la curvatura principal a dicha imagen.

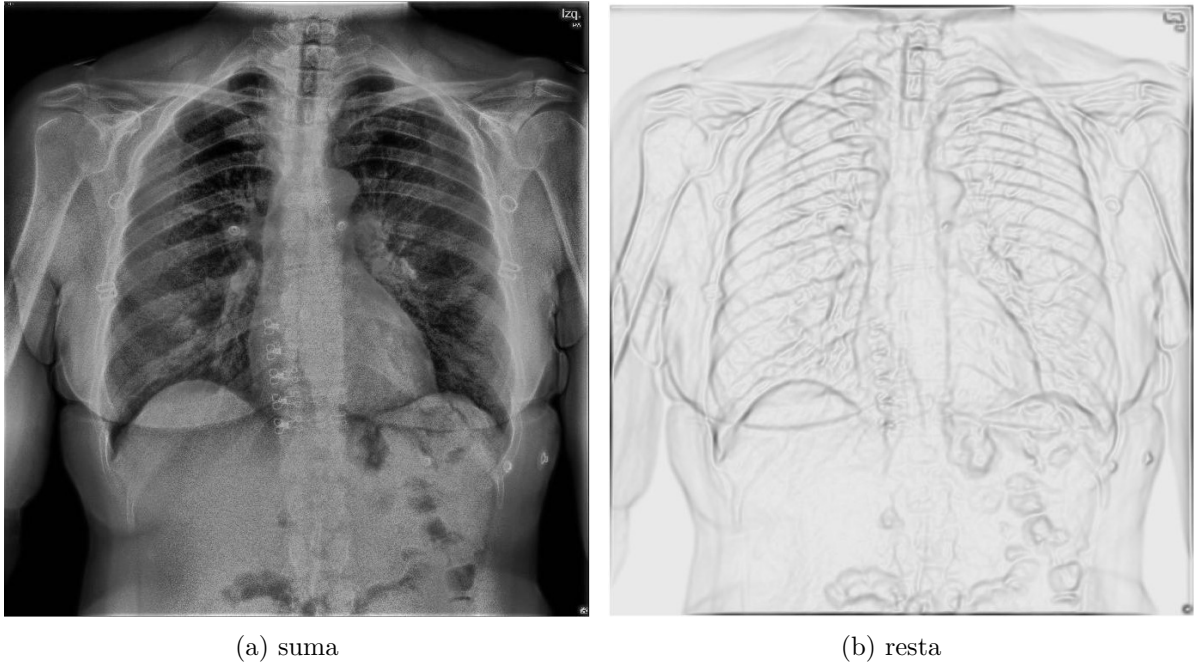


Fig. 4.46. (a) suma de la imagen “covid2” con el detector de bordes clásico y la curvatura principal. (b) resta de la imagen “covid2” con el detector de bordes clásico y la curvatura principal.

La Figura 4.47b presenta los resultados de procesar la imagen “covid3” utilizando el detector de bordes clásico definido en la expresión (2.33), obteniendo un error cuadrático medio de $MSE = 0.000375$. Los parámetros optimizados utilizados fueron: $\alpha = 0.138638$, $\beta = 0.4$, $\gamma = 0.9999$ y $\sigma = 0.502664$. Luego, se aplicó la curvatura principal.

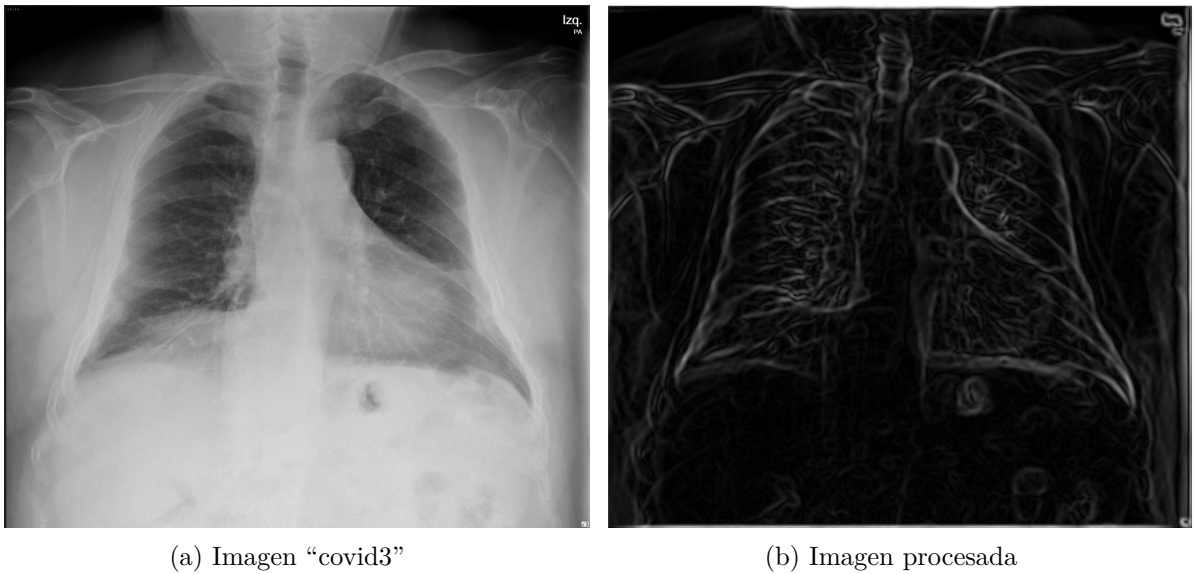


Fig. 4.47. (a) Imagen “covid3”. (b) Imagen “covid3” procesada con la curvatura principal.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.48 se presentan los resultados de sumar y restar la imagen “covid3” con el resultado de aplicar el detector de bordes clásico y la curvatura principal a dicha imagen.

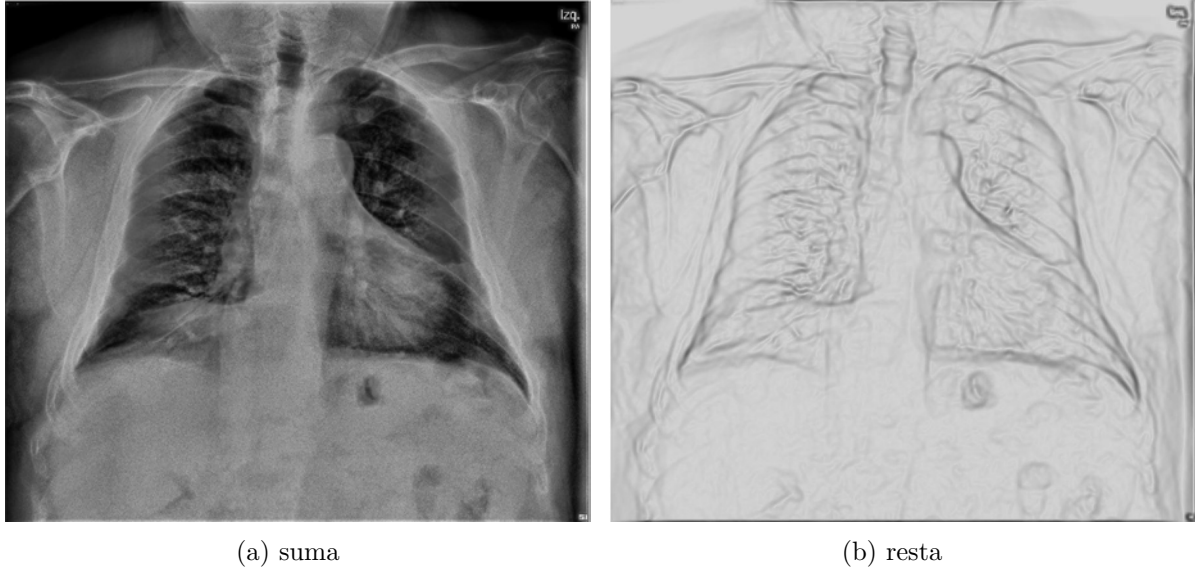


Fig. 4.48. (a) suma de la imagen “covid3” con el detector de bordes clásico y la curvatura principal. (b) resta de la imagen “covid3” con el detector de bordes clásico y la curvatura principal.

La Figura 4.49b presenta los resultados de procesar la imagen “covid4” utilizando el detector de bordes clásico definido en la expresión (2.33), obteniendo un error cuadrático medio de $MSE = 0.000287$. Los parámetros optimizados utilizados fueron: $\alpha = 0.1$, $\beta = 0.1$, $\gamma = 0.9999$ y $\sigma = 0.500544$. Luego, se aplicó la curvatura principal.

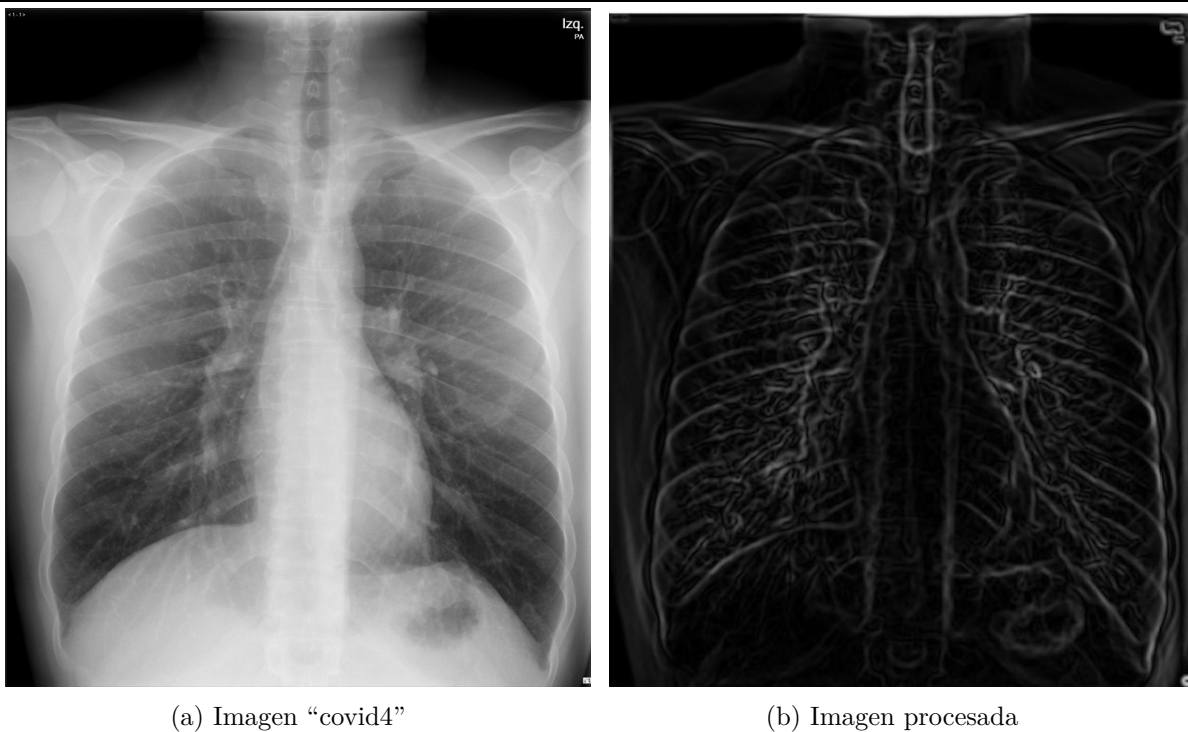


Fig. 4.49. (a) Imagen "covid4". (b) Imagen "covid4" procesada con la curvatura principal.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.50 se presentan los resultados de sumar y restar la imagen "covid4" con el resultado de aplicar el detector de bordes clásico y la curvatura principal a dicha imagen.

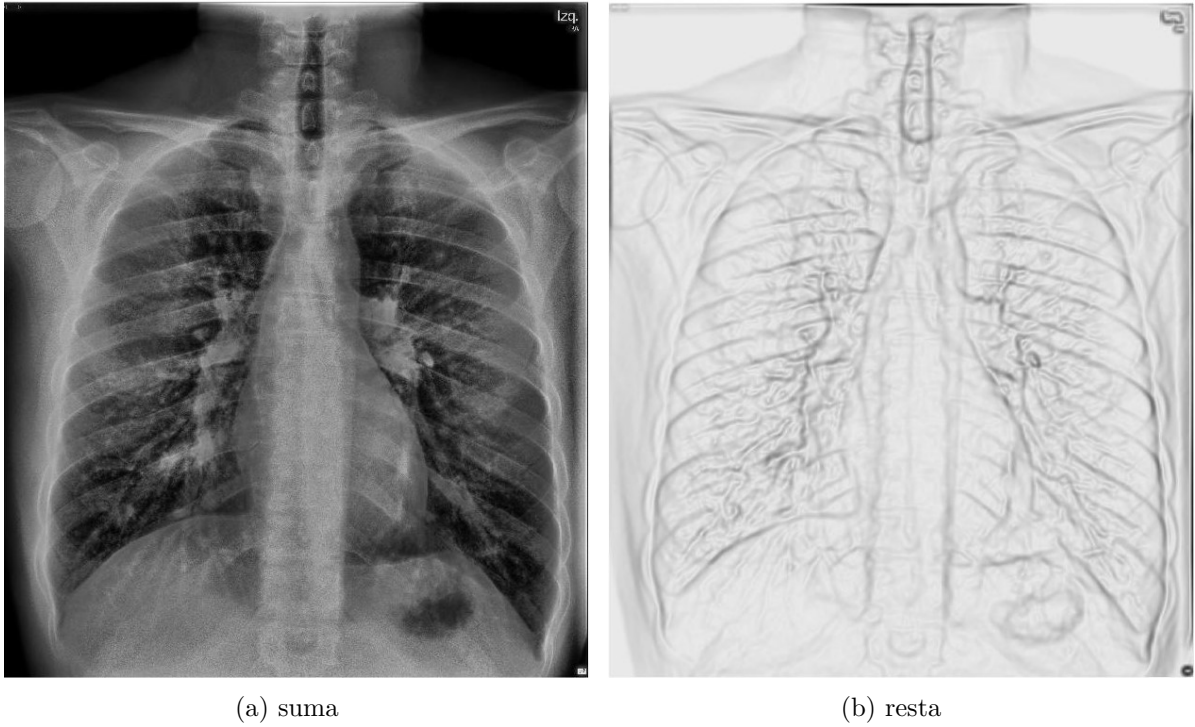


Fig. 4.50. (a) suma de la imagen “covid4” con el detector de bordes clásico y la curvatura principal. (b) resta de la imagen “covid4” con el detector de bordes clásico y la curvatura principal.

La Figura 4.51b presenta los resultados de procesar la imagen “covid5” utilizando el detector de bordes clásico definido en la expresión (2.33), obteniendo un error cuadrático medio de $MSE = 0.000344$. Los parámetros optimizados utilizados fueron: $\alpha = 0.399018$, $\beta = 0.1$, $\gamma = 0.648481$ y $\sigma = 0.460605$. Luego, se aplicó la curvatura principal.

CAPÍTULO 4. GRADIENTE GAUSSIANO \mathcal{M} -TRUNCADO FRACCIONARIO APLICADO A IMÁGENES DIGITALES

La Figura 4.53b presenta los resultados de procesar la imagen “covid6” utilizando el detector de bordes clásico definido en la expresión (2.33), obteniendo un error cuadrático medio de $MSE = 0.000449$. Los parámetros optimizados utilizados fueron: $\alpha = 0.292458$, $\beta = 0.389073$, $\gamma = 0.992907$ y $\sigma = 0.505505$. Luego, se aplicó la curvatura principal.

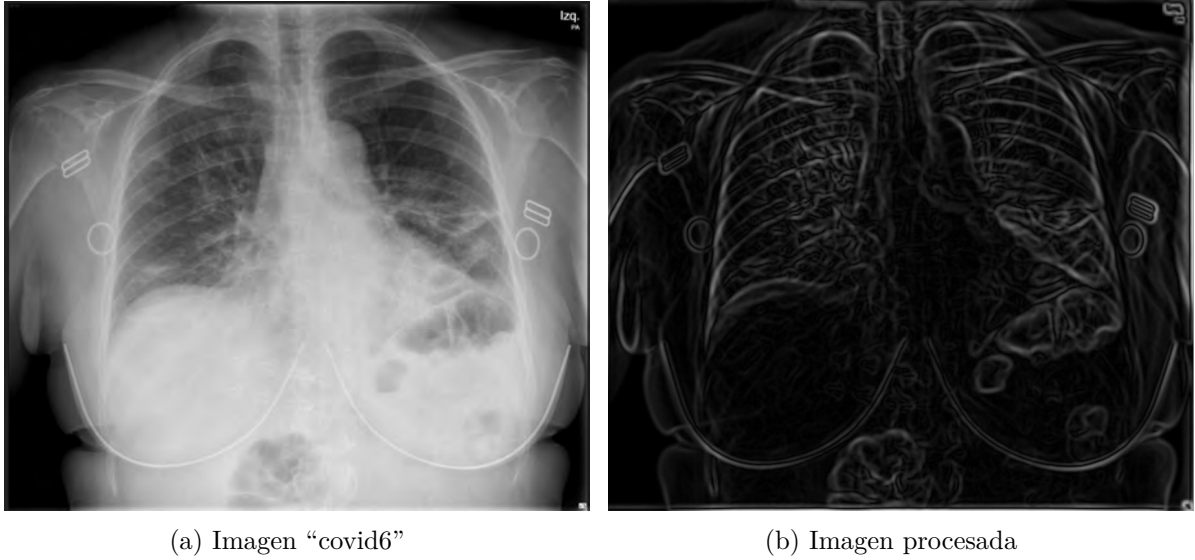


Fig. 4.53. (a) suma de la imagen original con el detector de bordes generalizado. (b) resta de la imagen original con el detector de bordes generalizado.

A continuación, se sigue el procedimiento ilustrado en el diagrama de la Figura 3.3. En la Figura 4.54 se presentan los resultados de sumar y restar la imagen “covid4” con el resultado de aplicar el detector de bordes clásico y la curvatura principal a dicha imagen.



(a) suma

(b) resta

Fig. 4.54. (a) suma de la imagen “covid6” con el detector de bordes clásico y la curvatura principal. (b) resta de la imagen “covid6” con el detector de bordes clásico y la curvatura principal.

4.3.3. Análisis de resultados

De los resultados anteriores se observa que, al procesar imágenes médicas con el detector de bordes y aplicar posteriormente la curvatura principal, se mejora la textura de los bordes, haciéndolos más visibles.

Capítulo 5

Conclusiones

Los trabajos de detección de bordes con el detector propuesto tuvieron buenos resultados, destacándose especialmente al introducir la matriz hessiana. Este enfoque mejoró la precisión en la detección de bordes, también resaltó características en imágenes médicas. Las imágenes procesadas mostraron una mayor claridad y definición en las estructuras, bordes y contornos.

Adicionalmente, la incorporación del algoritmo de optimización PSO permitió obtener los mejores resultados visuales. Con este algoritmo se optimizaron los parámetros, resultando en imágenes con bordes más definidos y características mejor resaltadas. Al tener tres grados de libertad posibilita la adaptación a la naturaleza de la imagen de entrada, lo que conlleva a mejoras visuales que se pueden medir cuantitativamente para comparar los resultados con los de otros métodos descritos en la literatura.

Con el detector de bordes con órdenes variables y la optimización del parámetro σ , se obtuvieron muy buenos resultados, inclusive superiores en las métricas de PSNR y SSIM, comparados con el detector de bordes generalizado.

La matriz hessiana permite resaltar características con mayor precisión. Este enfoque, en particular, facilita la identificación de estructuras, bordes y contornos en imágenes de resonancia magnética y tomografía computarizada.

5.1. Trabajos Futuros

En esta sección se exploran las posibles direcciones para investigaciones futuras basadas en el estudio realizado en esta tesis.

El objetivo de esta sección es identificar áreas de mejora y extensión para el trabajo realizado, así como proponer nuevas líneas de investigación que puedan aprovechar los resultados obtenidos.

Aunque esta investigación ha logrado importantes avances en la detección de bordes de una imagen mediante el uso de la derivada fraccionaria \mathcal{M} -truncada, hay algunas limitaciones que deben tenerse en cuenta. Por ejemplo, el empleo de órdenes complejos.

Una extensión natural de este trabajo sería aplicar la derivada \mathcal{M} -truncada con órdenes complejos para el mejoramiento de imágenes digitales además de aplicar la matriz Hessiana en dicho resultado, también se debe de explorar el uso de la matriz Hessiana usando el detector de bordes generalizado usado en esta tesis.

En conclusión, a pesar de que esta investigación ha logrado importantes avances en la detección de bordes en una imagen utilizando la derivada \mathcal{M} -truncada, hay numerosas direcciones prometedoras para futuras investigaciones. Al abordar las limitaciones actuales y explorar nuevas aplicaciones y mejoras metodológicas, futuras investigaciones pueden ampliar y profundizar los hallazgos presentados en esta tesis.

Bibliografía

- [1] U. D. Bosco, “*Introducción a matlab y toolbox image processing*”. Facultad de ingeniería, 2013.
- [2] L. A. Castellanos Sanabria and S. Niño Gómez, “Interfaz gráfica tipo educativo para el procesamiento de imágenes dicom,” *Facultad de Ingenierías*, 2019.
- [3] M. Sánchez, R. Vaquero, and C. Vásquez, “Control de un robot móvil con visión: rectificación de imágenes para agricultura de precisión,” *Proyecto de Sistemas Informáticos, Universidad Complutense de Madrid*, 2012.
- [4] R. Reynaga and W. Mayta, “Introducción al reconocimiento de patrones,” *Fides et Ratio-Revista de Difusión cultural y científica de la Universidad La Salle en Bolivia*, vol. 3, no. 3, pp. 41–44, 2009.
- [5] C. Platero, “Introducción a la visión artificial,” *UPM*, 2009. [Online]. Available: http://www.elai.upm.es/webantigua/spain/Asignaturas/MIP_VisionArtificial/ApuntesVA/cap1IntroVA.
- [6] J. E. Elizondo and L. P. Maestre, “Fundamentos de procesamiento de imágenes,” *Mexicali: Universidad Autónoma de Baja California*, 2005.
- [7] W. Gao, X. Zhang, L. Yang, and H. Liu, “An improved sobel edge detection,” in *2010 3rd International conference on computer science and information technology*, vol. 5. IEEE, 2010, pp. 67–71.
- [8] J. Canny, ““a computational approach to edge detection,” in readings in computer vision.” *Elsevier*, pp. 184–203, 1987.
- [9] N. Senthilkumaran and R. Rajesh, “Image segmentation-a survey of soft computing approaches,” in *2009 International Conference on Advances in Recent Technologies in Communication and Computing*. IEEE, 2009, pp. 844–846.
- [10] D. Sierociuk, T. Skovranek, M. Macias, I. Podlubny, I. Petras, A. Dzielinski, and P. Ziubinski, “Diffusion process modeling by using fractional-order models,” *Applied Mathematics and Computation*, vol. 257, pp. 2–11, 2015.
- [11] A. Nandal, H. Gamboa-Rosales, A. Dhaka, J. M. Celaya-Padilla, J. I. Galvan-Tejada, C. E. Galvan-Tejada, F. J. Martinez-Ruiz, and C. Guzman-Valdivia, “Image edge detection using fractional calculus with feature and contrast enhancement,” *Circuits, Systems, and Signal Processing*, vol. 37, pp. 3946–3972, 2018.

- [12] J. D. Pereyra Guzman *et al.*, “Diseño de filtros fraccionarios y su aplicación en la detección de características y reconstrucciones de imágenes,” Ph.D. dissertation, Tecnológico Nacional de México, 2019.
- [13] J. E. Solis Perez, “Cálculo fraccionario aplicado en procesamiento de imágenes y señales,” Ph.D. dissertation, Tecnológico Nacional de México, 2020.
- [14] J. Solís-Pérez, J. F. Gómez-Aguilar, R. F. Escobar-Jiménez, and J. Reyes-Reyes, “Blood vessel detection based on fractional hessian matrix with non-singular mittag-leffler gaussian kernel,” *Biomedical Signal Processing and Control*, vol. 54, p. 101584, 2019.
- [15] J. Lavín-Delgado, J. Solís-Pérez, J. F. Gómez-Aguilar, and R. F. Escobar-Jiménez, “A new fractional-order mask for image edge detection based on caputo-fabrizio fractional-order derivative without singular kernel,” *Circuits, Systems, and Signal Processing*, vol. 39, pp. 1419–1448, 2020.
- [16] J. Lavín-Delgado, J. Solís-Pérez, J. Gómez-Aguilar, and R. Escobar-Jiménez, “Image edge detection using fractional conformable derivatives in liouville-caputo sense for medical image processing,” *Fractional Calculus in Medical and Health Science*, pp. 1–54, 2020.
- [17] J. G. Puente-Córdova, “La derivada conformable y sus aplicaciones en ingeniería,” *Ingenierías*, vol. 23, no. 88, pp. 20–31, 2020.
- [18] L. Roberts, “Machine perception of three-dimensional solids.[w:] jt tippet et al.(red.), optical and electo-optical information processing,” 1965.
- [19] J. Prewitt, “Object enhancement and extraction picture processing and psychopictorics. 1970,” 1970.
- [20] R. Roesser, “A discrete state-space model for linear image processing,” *IEEE Transactions on Automatic Control*, vol. 20, no. 1, pp. 1–10, 1975.
- [21] J. Canny and J. Ponce, “Readings in computer vision,” *A Computational Approach to Edge Detection; Morgan Kaufmann: San Francisco, CA, SUA*, 1987.
- [22] G. Deng and L. Cahill, “An adaptive gaussian filter for noise reduction and edge detection,” in *1993 IEEE conference record nuclear science symposium and medical imaging conference*. IEEE, 1993, pp. 1615–1619.
- [23] F. Russo, “Edge detection in noisy images using fuzzy reasoning,” in *IMTC/98 Conference Proceedings. IEEE Instrumentation and Measurement Technology Conference. Where Instrumentation is Going (Cat. No. 98CH36222)*, vol. 1. IEEE, 1998, pp. 369–372.
- [24] M. Gudmundsson, E. A. El-Kwae, and M. R. Kabuka, “Edge detection in medical images using a genetic algorithm,” *IEEE transactions on medical imaging*, vol. 17, no. 3, pp. 469–474, 1998.
- [25] M. Sharifi, M. Fathy, and M. T. Mahmoudi, “A classified and comparative study of edge detection algorithms,” in *Proceedings. International conference on information technology: Coding and computing*. IEEE, 2002, pp. 117–120.

- [26] S.-H. Yun, G. J. Tearney, J. F. de Boer, N. Iftimia, and B. E. Bouma, “High-speed optical frequency-domain imaging,” *Optics express*, vol. 11, no. 22, pp. 2953–2963, 2003.
- [27] H. Nezamabadi-Pour, S. Saryazdi, and E. Rashedi, “Edge detection using ant algorithms,” *Soft computing*, vol. 10, pp. 623–628, 2006.
- [28] B. M. Vinagre and C. A. Monje, “Introducción al control fraccionario,” *Revista Iberoamericana de Automática e Informática Industrial*, vol. 3, no. 3, pp. 5–23, 2010.
- [29] C. Gao, J. Zhou, J. Hu, and F. Lang, “Edge detection of colour image based on quaternion fractional differential,” *IET Image Processing*, vol. 5, no. 3, pp. 261–272, 2011.
- [30] D. Tian, J. Wu, and Y. Yang, “A fractional-order edge detection operator for medical image structure feature extraction,” in *The 26th Chinese Control and Decision Conference (2014 CCDC)*. IEEE, 2014, pp. 5173–5176.
- [31] Q. Yang, D. Chen, T. Zhao, and Y. Chen, “Fractional calculus in image processing: a review,” *Fractional Calculus and Applied Analysis*, vol. 19, no. 5, pp. 1222–1249, 2016.
- [32] J. Dabass and R. Vig, “Biomedical image enhancement using different techniques-a comparative study,” in *Data Science and Analytics: 4th International Conference on Recent Developments in Science, Engineering and Technology, REDSET 2017, Gurgaon, India, October 13-14, 2017, Revised Selected Papers 4*. Springer, 2018, pp. 260–286.
- [33] J. M. Cruz-Duarte, J. Rosales-Garcia, C. R. Correa-Cely, A. Garcia-Perez, and J. G. Avina-Cervantes, “A closed form expression for the gaussian-based caputo-fabrizio fractional derivative for signal processing applications,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 61, pp. 138–148, 2018.
- [34] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. Van Gool, “Exploring cross-image pixel contrast for semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7303–7313.
- [35] X. Gong, X. Xia, W. Zhu, B. Zhang, D. Doermann, and L. Zhuo, “Deformable gabor feature networks for biomedical image classification,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 4004–4012.
- [36] R. C. Gonzalez, *Digital image processing*. Pearson education india, 2009.
- [37] F. Mainardi, “Fractional calculus: Theory and applications,” p. 145, 2018.
- [38] J. Havil, *Gamma: exploring Euler’s constant*. Princeton University Press, 2010.
- [39] I. Podlubny, “Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications,” *Elsevier*, vol. 198, 1998.
- [40] S. Salahshour, A. Ahmadian, S. Abbasbandy, and D. Baleanu, “M-fractional derivative under interval uncertainty: Theory, properties and applications,” *Chaos, Solitons & Fractals*, vol. 117, pp. 84–93, 2018.

- [41] J. E. Solís-Pérez and J. F. Gómez-Aguilar, “Novel fractional operators with three orders and power-law, exponential decay and mittag-leffler memories involving the truncated m-derivative,” *Symmetry*, vol. 12, no. 4, p. 626, 2020.
- [42] H. Deng, W. Zhang, E. Mortensen, T. Dietterich, and L. Shapiro, “Principal curvature-based region detector for object recognition,” in *2007 IEEE conference on computer vision and pattern recognition*. IEEE, 2007, pp. 1–8.
- [43] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, pp. 91–110, 2004.
- [44] W. K. Pratt, *Digital image processing: PIKS Scientific inside*. Wiley Online Library, 2007, vol. 4.
- [45] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, “A statistical evaluation of recent full reference image quality assessment algorithms,” *IEEE Transactions on image processing*, vol. 15, no. 11, pp. 3440–3451, 2006.
- [46] Z. Wang and Q. Li, “Information content weighting for perceptual image quality assessment,” *IEEE Transactions on image processing*, vol. 20, no. 5, pp. 1185–1198, 2010.
- [47] A. Žilinskas, “Practical mathematical optimization: An introduction to basic optimization theory and classical and new gradient-based algorithms,” 2006.
- [48] J. Stewart and J. H. Romo, *cálculo*. Cengage Learning, 2017.

Anexos

Anexo A

Pseudocódigos

Pseudocódigo 1 Optimización de los parámetros $\alpha, \beta, \gamma, \xi, \eta, \lambda, \theta, \phi$ y σ con PSO

Definir la función objetivo

fun \leftarrow @(x, img) rms_regresionlineal(x, img)

Cargar y preparar la imagen**Parámetros del PSO**

num_particulas, num_dimensiones, max_iteraciones, w, c1, c2

Definir límites

[limite superior de $\alpha, \beta, \gamma, \xi, \eta, \lambda, \theta, \phi, \sigma$]

[limite inferior de $\alpha, \beta, \gamma, \xi, \eta, \lambda, \theta, \phi, \sigma$]

Inicializar partículas y velocidades

posicion \leftarrow limites_inferior + (limites_superior - limites_inferior) · rand(num_particulas, num_dimensiones)
velocidad \leftarrow rand(num_particulas, num_dimensiones)

Inicializar pbest y gbest

pbest \leftarrow posicion

gbest \leftarrow posicion(1, :)

mejor_fitness \leftarrow inf

for iteracion = 1 to max_iteraciones do

fitness \leftarrow zeros(num_particulas, 1)

for i = 1 to num_particulas do

fitness(i) \leftarrow fun(posicion(i, :), img)

if fitness(i) < fun(pbest(i, :), img) **then**

pbest(i, :) \leftarrow posicion(i, :)

if fitness(i) < mejor_fitness **then**

mejor_fitness \leftarrow fitness(i)

gbest \leftarrow posicion(i, :)

for i = 1 to num_particulas do

velocidad(i, :) \leftarrow w · velocidad(i, :) + c1 · rand · (pbest(i, :) - posicion(i, :)) + c2 · rand · (gbest - posicion(i, :))

posicion(i, :) \leftarrow posicion(i, :) + velocidad(i, :)

for k = 1 to num_dimensiones do

if posicion(i, k) > limites_superior(k) **then**

posicion(i, k) \leftarrow limites_superior(k)

else if posicion(i, k) < limites_inferior(k) **then**

posicion(i, k) \leftarrow limites_inferior(k)

Mostrar la mejor solución

La mejor solución encontrada para $\alpha, \beta, \gamma, \xi, \eta, \lambda, \theta, \phi$ y σ

Valor de MSE en la mejor solución

Pseudocódigo 2 Gradiente Generalizado

$Ne \leftarrow$ numero impar

$x \leftarrow$ linspace($-Ne, Ne$, numero deseado de puntos)

$h \leftarrow x(2) - x(1)$

$\mu \leftarrow 0$

Calcular los valores y de la función gaussiana para cada valor x

$y_{\text{exacta}} \leftarrow \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

$g \leftarrow$ derivada del kernel gaussiano $-\frac{(x-\mu)}{\sigma^3\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

$g_0 \leftarrow y_{\text{exacta}}[1]$

$g'_0 \leftarrow y_{\text{exacta}}[1]$

$N \leftarrow \text{length}(x) - 1$

Inicializa un vector para almacenar los valores de g

$g_{\text{values}} \leftarrow$ vector de ceros con longitud de x

$g_{\text{values}}[1] \leftarrow g_0$

for $n \leftarrow 0$ hasta N do

$\text{suma} \leftarrow 0$

for $j \leftarrow 0$ hasta n do

$\text{suma} \leftarrow \text{suma} + \left[((n+1-j)^\gamma \cdot (n-j+2+\gamma) - (n-j)^\gamma \cdot (n-j+2+2\gamma)) \cdot g(x[j+1]) \cdot x[j+1]^{(1-\alpha)} \right. \\ \left. + \left[((n-j+1)^{(\gamma+1)} - (n-j)^\gamma \cdot (n-j+1+\gamma)) \cdot g(x[j+1]) \cdot x[j+1]^{(1-\alpha)} \right] \right]$

$g_{\text{values}}[n+2] \leftarrow g_0 + \frac{h^\gamma}{\Gamma(\gamma+2) \cdot \Gamma(\beta+1)} \cdot \text{suma}$

Ajusta g_{values} para que tenga la misma longitud que x

$g_{\text{values}} \leftarrow g_{\text{values}}[1 : \text{length}(x)]$

$N \leftarrow$ numero impar

▷ Tamaño del kernel

$\text{indice} \leftarrow$ rango desde $-(N/2)$ hasta $(N/2)$

Recuperar puntos del kernel gaussiano continuo

$F_r G \leftarrow \text{interp1}(x, g_{\text{values}}, \text{indice})$

Normalizar el kernel gaussiano

$F_r G \leftarrow F_r G / \sum(F_r G)$

$F_r G^2 d \leftarrow F_r G \cdot F_r G'$

▷ Multiplicación elemento a elemento

$F_r G^2 d \leftarrow \text{abs}(F_r G^2 d)$

$\text{mascara}_x \leftarrow [-1 \ 0 \ 1]$

$\text{mascara}_y \leftarrow [-1 \ 0 \ 1]'$

$\text{bordes } x \leftarrow F_r G^2 d * \text{mascara}_x * \text{img}$

$\text{bordes } y \leftarrow F_r G^2 d * \text{mascara}_y * \text{img}$

Pseudocódigo 3 Continuación del algoritmo

$NN \leftarrow \text{length}(x) - 1$

Inicializa un vector para almacenar los valores de g

$g_{\text{values_prima}} \leftarrow$ vector de ceros con longitud de x

$g_{\text{values_prima}}[1] \leftarrow g'_0$

for $n \leftarrow 0$ **hasta** NN **do**

$\text{suma} \leftarrow 0$

for $j \leftarrow 0$ **hasta** n **do**

$\text{suma} \leftarrow \text{suma} + \left[\left((n+1-j)^\lambda \cdot (n-j+2+\lambda) - (n-j)^\lambda \cdot (n-j+2+2\lambda) \right) \cdot g(x[j+1]) \cdot x[j+1]^{(1-\xi)} \right]$
 $\left[\left((n-j+1)^{(\lambda+1)} - (n-j)^\lambda \cdot (n-j+1+\lambda) \right) \cdot g(x[j+1]) \cdot x[j+1]^{(1-\xi)} \right]$

$g_{\text{values_prima}}[n+2] \leftarrow g'_0 + \frac{h^\lambda}{\Gamma(\lambda+2) \cdot \Gamma(\eta+1)} \cdot \text{suma}$

Ajusta $g_{\text{values_prima}}$ para que tenga la misma longitud que x

$g_{\text{values_prima}} \leftarrow g_{\text{values_prima}}[1 : \text{length}(x)]$

$N \leftarrow$ numero impar

▷ Tamaño del kernel

$\text{indice} \leftarrow$ rango desde $-(N/2)$ hasta $(N/2)$

Recuperar puntos del kernel gaussiano continuo

$FrG_{\text{prima}} \leftarrow \text{interp1}(x, g_{\text{values_prima}}, \text{indice})$

Normalizar el kernel gaussiano

$FrG_{\text{prima}} \leftarrow FrG_{\text{prima}} / \sum(FrG_{\text{prima}})$

$FrG_{\text{prima2d}} \leftarrow FrG_{\text{prima}} \cdot FrG'_{\text{prima}}$

▷ Multiplicación elemento a elemento

$FrG_{\text{prima2d}} \leftarrow \text{abs}(FrG_{\text{prima2d}})$

$\text{mascara}_{xx} \leftarrow [-1 \ 0 \ 1]$

$\text{mascara}_{yy} \leftarrow [-1 \ 0 \ 1]'$

$\text{bordes}_x_{\text{prima}} \leftarrow FrG_{\text{prima2d}} * \text{mascara}_{xx} * \text{img}$

$\text{bordes}_y_{\text{prima}} \leftarrow FrG_{\text{prima2d}} * \text{mascara}_{yy} * \text{img}$

Normalizar los valores resultantes para mostrar la imagen

$M \leftarrow \text{abs}(\text{bordes}_x \cdot \cos(\theta)) + \text{abs}(\text{bordes}_y \cdot \sin(\theta)) +$

$\text{abs}(\text{bordes}_x_{\text{prima}} \cdot \cos(\phi)) + \text{abs}(\text{bordes}_y_{\text{prima}} \cdot \sin(\phi))$

Pseudocódigo 4 Cálculo de RMS Regresión Lineal

Función: rms_regresionlineal(x , img)**Entrada:** x , imagen**Salida:** F Error cuadrático medio (MSE)**Verificar límites de x** $limites_inferior \leftarrow [\alpha, \beta, \gamma, \xi, \eta, \lambda, \theta, \phi, \sigma]$ $limites_superior \leftarrow [\alpha, \beta, \gamma, \xi, \eta, \lambda, \theta, \phi, \sigma]$ $x \leftarrow \text{máx}(\text{mín}(x, limites_superior), limites_inferior)$ **Extraer los parámetros** $\alpha \leftarrow x(1)$ $\beta \leftarrow x(2)$ $\gamma \leftarrow x(3)$ $\xi \leftarrow x(4)$ $\eta \leftarrow x(5)$ $\lambda \leftarrow x(6)$ $\theta \leftarrow x(7)$ $\phi \leftarrow x(8)$ $\sigma \leftarrow x(9)$ **Calcular la imagen procesada**imagen \leftarrow Gradcomp($\alpha, \beta, \gamma, \xi, \eta, \lambda, \theta, \phi, \sigma$, img)**Calcular el MSE entre la imagen original y la imagen procesada** $F \leftarrow \text{mse}(\text{imagen}, \text{img})$

Pseudocódigo 5 Detección de Bordes usando la derivada \mathcal{M} -truncada fraccionaria Gradiente Generalizado

Inicio**Cargar y preparar la imagen****Deteccion de bordes con la derivada M-truncada**imagen \leftarrow Grad($\alpha, \beta, \gamma, \xi, \eta, \lambda, \theta, \phi, \sigma$, img)**Operaciones aritméticas con imágenes**suma \leftarrow imagen original + imagenresta \leftarrow imagen original - imagen**Evaluación con metricas de la imagen procesada****Fin**

Pseudocódigo 6 Optimización del Parámetro σ con PSO**Definir la función objetivo**

```
fun ← @(x, img) rms_regresionlineal(x, img)
```

Cargar y preparar la imagen**Parámetros del PSO**

```
num_particulas, num_dimensiones, max_iteraciones, w, c1, c2
```

Definir límites

```
[limite inferior de  $\sigma$ , limite superior de  $\sigma$ ]
```

Inicializar partículas y velocidades

```
posicion ← limites_inferior + (limites_superior - limites_inferior)
·rand(num_particulas, num_dimensiones)
velocidad ← rand(num_particulas, num_dimensiones)
```

Inicializar pbest y gbest

```
pbest ← posicion
```

```
gbest ← posicion(1, :)
```

```
mejor_fitness ← inf
```

for iteracion = 1 to max_iteraciones do

```
fitness ← zeros(num_particulas, 1)
```

for i = 1 to num_particulas do

```
fitness(i) ← fun(posicion(i, :), img)
```

```
if fitness(i) < fun(pbest(i, :), img) then
```

```
    pbest(i, :) ← posicion(i, :)
```

```
if fitness(i) < mejor_fitness then
```

```
    mejor_fitness ← fitness(i)
```

```
    gbest ← posicion(i, :)
```

for i = 1 to num_particulas do

```
    velocidad(i, :) ← w · velocidad(i, :) + c1 · rand · (pbest(i, :) - posicion(i, :)) + c2 · rand · (gbest -
posicion(i, :))
```

```
    posicion(i, :) ← posicion(i, :) + velocidad(i, :)
```

for k = 1 to num_dimensiones do

```
    if posicion(i, k) > limites_superior(k) then
```

```
        posicion(i, k) ← limites_superior(k)
```

```
    else if posicion(i, k) < limites_inferior(k) then
```

```
        posicion(i, k) ← limites_inferior(k)
```

Mostrar la mejor solución

La mejor solución encontrada para σ

Valor de MSE en la mejor solución

Pseudocódigo 7 Gradiente Variable

function GRADVAR(σ , img)

$Ne \leftarrow$ numero impar

$x \leftarrow$ linspace($-Ne, Ne$, numero deseado de puntos)

$h \leftarrow x(2) - x(1)$

$\mu \leftarrow 0$

Definir funciones de α , β y γ

$\alpha(t) \leftarrow 0.5 \cdot (\tanh(t) + 1)$

$\beta(t) \leftarrow 0.5 \cdot (\tanh(t) + 1)$

$\gamma(t) \leftarrow 0.5 \cdot (\tanh(t) + 1)$

$y \leftarrow \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

derivada kernel gaussiano $g(x) \leftarrow -\frac{x-\mu}{\sigma^3\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

$g_0 \leftarrow y(1)$

$N \leftarrow$ longitud vector(x) - 1

$g_{values} \leftarrow$ zeros(1, longitud vector(x))

$g_{values}(1) \leftarrow g_0$

for $n = 0$ to N **do**

$suma \leftarrow 0$

for $j = 0$ to n **do**

$suma \leftarrow suma + \left((n+1-j)^{\gamma(x(j+1))} \cdot (n-j+2+\gamma(x(j+1))) - (n-j)^{\gamma(x(j+1))} \cdot (n-j+2+2\gamma(x(j+1))) \right) \cdot g(x(j+1)) \cdot x(j+1)^{1-\alpha(x(j+1))}$
 $- \left((n-j+1)^{\gamma(x(j+1))+1} - (n-j)^{\gamma(x(j+1))} \cdot (n-j+1+\gamma(x(j+1))) \right) \cdot g(x(j+1)) \cdot x(j+1)^{1-\alpha(x(j+1))}$

$g_{values}(n+2) \leftarrow g_0 + \frac{h^{\gamma(x(n+1))}}{\Gamma(\gamma(x(n+1))+2) \cdot \Gamma(\beta(x(n+1))+1)} \cdot suma$

$g_{values} \leftarrow g_{values}(1 : \text{length}(x))$

$N \leftarrow$ numero impar

$indice \leftarrow -(N/2)$ hasta $(N/2)$

Interpolación

$FrG \leftarrow$ interp1($x, g_{values}, indice$)

$FrG \leftarrow \frac{FrG}{\sum FrG(:)}$

$FrG2d \leftarrow FrG \cdot FrG'$

$FrG2d \leftarrow \text{abs}(FrG2d)$

$mascara_x \leftarrow [-1, 0, 1]$

$mascara_y \leftarrow [-1, 0, 1]'$

Convolución

bordes x $\leftarrow FrG2d * mascara_x * img$

bordes y $\leftarrow FrG2d * mascara_y * img$

$M \leftarrow \text{abs}(\text{bordes x}) + \text{abs}(\text{bordes y})$

return M

Pseudocódigo 8 Función rms_regresionlineal

Definir límites de x

[limite inferior de σ , limite superior de σ]

Verificar y ajustar x dentro de los límites

$x \leftarrow \max(\min(x, \text{limites_superior}), \text{limites_inferior})$

Extraer los parámetros

$\sigma \leftarrow x[1]$

Calcular la imagen procesada

$\text{imagen} \leftarrow \text{Gradvar}(\sigma, \text{img})$

Calcular el MSE entre la imagen original y la imagen procesada

$F \leftarrow \text{immse}(\text{imagen}, \text{img})$

Retornar el valor de F

Pseudocódigo 9 Detección de Bordes usando la derivada \mathcal{M} -truncada fraccionaria órdenes variables

Inicio

Cargar y preparar la imagen

Definir funciones para α , β y γ

$\alpha(t) \leftarrow \frac{1}{1+\exp(-t)}$

$\beta(t) \leftarrow \frac{1}{1+\exp(-t)}$

$\gamma(t) \leftarrow \frac{1}{1+\exp(-t)}$

Deteccion de bordes con la derivada M-truncada

$\text{imagen} \leftarrow \text{Gradvar}(\sigma, \text{img})$

Operaciones aritméticas con imágenes

$\text{suma} \leftarrow \text{imagen original} + \text{imagen}$

$\text{resta} \leftarrow \text{imagen original} - \text{imagen}$

Evaluación con metricas de la imagen procesada

Fin

Pseudocódigo 10 Cálculo de los valores propios de la imagen

function IMAGEEIGENVALUES(img, σ , α , β , γ)

Entrada:

img

▷ Imagen de entrada

σ

▷ Parámetro de escala

α, β, γ

▷ Parámetros para el cálculo del gradiente

Salida:

lamda2

▷ Curvatura principal máxima

Calcular el Hessian 2D

gradx \leftarrow MGx(α , β , γ , img, σ)

Hxx \leftarrow MGx(α , β , γ , gradx, σ)

grady \leftarrow MGy(α , β , γ , img, σ)

Hyy \leftarrow MGy(α , β , γ , grady, σ)

Hxy \leftarrow MGy(α , β , γ , gradx, σ)

Inicializar matrices de valores propios

[row, col] \leftarrow size(img)

lamdaplus \leftarrow zeros(row, col)

lamdaminus \leftarrow zeros(row, col)

Encontrar los valores propios de la matriz Hessian

for r \leftarrow 1 **to** row **do**

for c \leftarrow 1 **to** col **do**

 [lamdaplus(r, c), lamdaminus(r, c)] \leftarrow lamdafind(Hxx(r, c), Hyy(r, c), Hxy(r, c))

Obtener la curvatura principal máxima

lamda2 \leftarrow lamdaplus

Retornar lamda2

Pseudocódigo 11 Cálculo de los valores propios de la matriz Hessiana

function LAMDAFINd(gxx1, gyy1, gxy1)**Entrada:**

gxx1

▷ Segunda derivada respecto a x

gyy1

▷ Segunda derivada respecto a y

gxy1

▷ Derivada mixta respecto a x e y **Salida:**

lamdaplus

▷ Valor propio máximo

lamdaminus

▷ Valor propio mínimo

Construir la matriz Hessiana

$$H \leftarrow \begin{bmatrix} gxx1 & gxy1 \\ gxy1 & gyy1 \end{bmatrix}$$

Obtener los valores propios de la matriz Hessiana $\lambda \leftarrow \text{eig}(H)$ **Determinar el valor propio máximo y mínimo****if** $\lambda(1) > \lambda(2)$ **then** lamdaplus $\leftarrow \lambda(1)$ lamdaminus $\leftarrow \lambda(2)$ **else if** $\lambda(1) < \lambda(2)$ **then** lamdaplus $\leftarrow \lambda(2)$ lamdaminus $\leftarrow \lambda(1)$ **else** lamdaplus $\leftarrow \lambda(1)$ lamdaminus $\leftarrow \lambda(2)$ **Retornar** lamdaplus, lamdaminus

Anexo B

Productos obtenidos



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

EL TECNOLÓGICO NACIONAL DE MÉXICO A TRAVÉS DEL CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO

OTORGA EL PRESENTE

RECONOCIMIENTO

A

Arturo García Morelos, Jesús Emmanuel Solís Pérez, Daniela Hernández Castañeda, José Francisco Gómez Aguilar

Por su invaluable presentación del artículo Improving the quality of digital images using the truncated-M fractional Gaussian gradient en el 1er Congreso Internacional de Tecnología y Ciencia Aplicada, desarrollado en el Tecnológico Nacional de México/Centro Nacional de Investigación y Desarrollo Tecnológico, TecNM/CENIDET, del 15 al 17 de noviembre de 2023.

Cuernavaca, Morelos, Noviembre de 2023



DR. ARTURO ERNESTO MARES GARDEA
DIRECTOR DEL CENTRO NACIONAL DE INVESTIGACIÓN
Y DESARROLLO TECNOLÓGICO

Sello Digital:

v9EAHcv7R7iSvxZnMTKUNsvw00/jh+YnNCcTft7EaaagT9W4GW3EIrJT1K9ea5ozUqz0plk5YdxQU3P2JAj1R7kUkYuywQ1h7oUPsZxYAJNNAAUoyWZYqWTnCVDLc8bBqeQf+riiYMt3DHgpCUGUHqdM3swy9zOd47FOA7ASN/YrEua1V5CxbDnKDXS79zed05JBSDQyBbovA6uBvsLW0It1UJMcaUpvz9JndS1/ZuH7PtByxcJbi2K5gEhqzCq1S22z8Mw3qpkS79m1SrU4wN0pQ3Wf4xz17rKxghpJFGfkK5GJ4RBmnMS1X16Jp+T9Lq006DfUP54SYBnvI/i6aA==

A0234123

<http://constancias.cenidet.tecnm.mx>