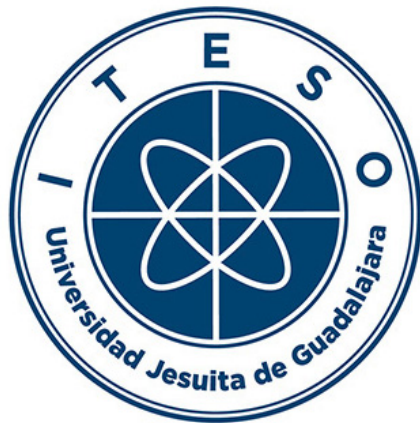


Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática
Maestría en Sistemas Computacionales



Análisis comparativo de métodos de reducción de ruido para imágenes generadas por radares de apertura sintética (SAR)

TRABAJO RECEPCIONAL que para obtener el **GRADO** de
MAESTRO EN SISTEMAS COMPUTACIONALES

Presenta: **MARTÍN FERNANDO LÓPEZ RUBIO**

Director: **DR. IVÁN ESTEBAN VILLALÓN TURRUBIATES**

Tlaquepaque, Jalisco. Agosto de 2024.

AGRADECIMIENTOS

El autor desea dar las gracias al ITESO, no sólo como institución escolar sino también por el apoyo económico recibido. Al Dr. Iván Villalón y los grandes profesores de la maestría. También quisiera agradecer al CONACYT, por el apoyo 2022-000018-02NACF-11477 recibido para cursar la maestría. Por último, el autor le da las gracias a IBM México, por el espacio y tiempo que permitieron dar seguimiento a sus responsabilidades escolares.

DEDICATORIA

El autor dedica esta tesis a Martín y Beatriz, Octavio y Alicia, Ramón y Ofelia, Carlos, y particularmente a Lucía, que sin su ayuda nada habría sido posible.

RESUMEN

En este trabajo, a manera de metaanálisis se comparan métodos para reducir el ruido “salt and pepper” en imágenes, analizando técnicas como filtros de promedio, transformadas wavelet, y métodos basados en inteligencia artificial (IA) como machine learning y deep learning contextualizado en especial para imágenes obtenidas por radares de apertura sintética (SAR), dónde es común encontrar este tipo de ruido específico. Con el objetivo de encontrar los métodos más eficaces y vislumbrar patrones de comportamiento entre ellos.

El análisis se basa en una recopilación de estadísticas de 59 métodos de reducción de ruido. En la investigación se concluye una selección de métodos mejor eficacia y se explora un patrón aparente en el que los métodos basados en inteligencia artificial tienden a ser más efectivos en niveles altos de ruido, además de revelar una falta de estandarización en los indicadores de evaluación y en los conjuntos de datos utilizados.

Se proponen posibles artículos científicos a publicar como trabajos futuros a partir de este metaanálisis.

TABLA DE CONTENIDO

INTRODUCCIÓN	10
1.1. ANTECEDENTES	11
1.2. JUSTIFICACIÓN	11
1.3. PROBLEMA	11
OBJETIVOS	11
1.4. NOVEDAD CIENTÍFICA, TECNOLÓGICA O APORTACIÓN	11
2. ESTADO DEL ARTE O DE LA TÉCNICA.....	12
2.1. ARTÍCULOS SOBRE MÉTODOS DE REDUCCIÓN DE RUIDO	13
2.1.1. ANATOMÍA DE UN ARTÍCULO	13
2.1.2. ARTÍCULOS DESTACABLES.....	15
2.1.3. ADMINISTRACIÓN DE LOS DATOS	15
2.1.4. EXCLUSIÓN DURANTE LA RECOPIACIÓN	15
2.1.5. INCONSISTENCIAS EN LOS BANCOS DE PRUEBA	15
2.1.6. ARTÍCULOS CON MÚLTIPLES RESULTADOS	16
2.1.7. RANKING POR BANCO DE PRUEBAS	16
3. MARCO TEÓRICO/CONCEPTUAL.....	17
3.1. RADAR DE APERTURA SINTÉTICA (SAR).....	18
3.2. RUIDO	18
3.2.1. RUIDO GAUSSIANO	18
3.2.2. RUIDO “SALT AND PEPPER”	18
3.3. REDUCCIÓN DE RUIDO	18
3.4. FILTROS POR PROMEDIO	18
3.5. FILTROS WAVELET	18
3.6. FILTROS WEINER	19
3.7. DIFUSIÓN ANISOTRÓPICA	19
3.8. FILTROS MORFOLÓGICOS.....	19
3.9. APRENDIZAJE AUTOMÁTICO	19
3.9.1. APRENDIZAJE PROFUNDO	19
3.10. CONVOLUCIÓN.....	19
3.11. MATRICES DE CONVOLUCIÓN	20
3.12. REDES CONVOLUCIONALES	20
3.13. AUTOENCODER	20
3.13.1. CODIFICADOR	20
3.13.1. DECODIFICADOR	20
3.14. ERROR CUADRADO PROMEDIO	21
3.15. RELACIÓN SEÑAL/RUIDO (SNR)	21
3.16. PROPORCIÓN MÁXIMA DE SEÑAL A RUIDO (PSNR).....	21
3.17. ÍNDICE DE PRESERVACIÓN DE CONTORNOS (EPI)	22
3.18. ÍNDICE DE SIMILITUD ESTRUCTURAL (SSIM).....	22
3.18.1. ÍNDICE DE SIMILITUD ESTRUCTURAL (MS-SSIM).....	23

3.19.	HERRAMIENTAS UTILIZADAS	23
3.20.	OBTENCIÓN DE IMÁGENES SAR	23
3.21.	INDICADORES UTILIZADOS	24
4.	DESARROLLO METODOLÓGICO	25
4.1.	HISTORIAL DE DESARROLLO	26
4.2.	CÓDIGO FUENTE.....	26
4.3.	DIAGRAMA ARQUITECTÓNICO	27
5.	RESULTADOS Y DISCUSIÓN	28
5.1.	RESULTADOS	29
5.1.1.	MÉTODOS POR MÉTODO COMPUTACIONAL	29
5.1.2.	AUSENCIA DE DATOS DE <i>HARDWARE</i>	29
5.1.3.	ANÁLISIS COMPARATIVO POR INDICADOR INDIVIDUAL	29
5.1.3.1.	MÉTODOS POR PSNR	30
5.1.3.2.	MÉTODOS POR SNR.....	32
5.1.3.3.	MÉTODOS POR EPI	33
5.1.3.4.	MÉTODOS POR SSIM.....	35
5.2.	DISCUSIÓN	37
5.2.1.	RESULTADOS DEL ANÁLISIS	37
5.2.2.	IMPLEMENTACIÓN CON INTELIGENCIA ARTIFICIAL	38
5.2.2.1.	DESCRIPCIÓN DEL MODELO	38
5.2.2.2.	INTRODUCCIÓN DE RUIDO.....	39
5.2.2.3.	ENTRENAMIENTO DEL MODELO	39
5.2.2.4.	RESULTADOS DE LAS PREDICCIONES DEL MODELO	39
5.2.3.	INDICADOR NUEVO.....	41
6.	CONCLUSIONES	42
6.1.	CONCLUSIONES.....	43
6.2.	TRABAJO FUTURO	43

LISTA DE FIGURAS

Figura 1. Comparativa visual de métodos de reducción de ruido	13
Figura 2. Pseudocódigo de un método en específico	14
Figura 3. Ejemplo de la operación de convolución	20
Figura 4. Diagrama arquitectónico de un <i>autoencoder</i>	21
Figura 5. Diagrama Arquitectónico del desarrollo	27
Figura 6. Métodos por método computacional.	29
Figura 7. Mapa de color por PSNR.	31
Figura 8. Mapa de calor por SNR.	33
Figura 9. Mapa de calor por EPI.	34
Figura 10. Métodos por SSIM.	36
Figura 11. Eficiencia por agrupación de métodos	37
Figura 12. Código fuente para introducir ruido.	39
Figura 13. Resultados del autoencoder con 1% de ruido.	40
Figura 14. Resultados del autoencoder con 10% de ruido.	40
Figura 15. Resultados del <i>autoencoder</i> con 20% de ruido.	40
Figura 16. Resultados de <i>autoencoder</i> con 30% de ruido.	41
Figura 17. Resultados del autoencoder con 40% de ruido.	41

LISTA DE TABLAS

Tabla 1. Resultados numéricos de métodos listados en un sólo artículo.....	14
Tabla 2. Ranking de bancos de imágenes.....	16
Tabla 3. Variables recuperadas de los artículos.	24
Tabla 4. Estadísticas acumuladas por PSNR pt.1.....	30
Tabla 5. Estadísticas acumuladas por PSNR pt.2.....	30
Tabla 6. Mejores métodos por PSNR.	30
Tabla 7. Estadísticas acumuladas por SNR.	32
Tabla 8. Mejores métodos por SNR.	32
Tabla 9. Estadísticas acumuladas por EPI.....	33
Tabla 10. Mejores métodos por EPI.	34
Tabla 11. Estadísticas acumuladas por SSIM.....	35
Tabla 12. Mejores métodos por SSIM.....	35
Tabla 13. Capas del modelo <i>autoencoder</i>	38

LISTA DE ACRÓNIMOS Y ABREVIATURAS

SAR	Synthetic Aperture Radar (Radar de apertura sintética)
PSNR	Peak Signal to Noise Ratio (Proporción máxima de señal a ruido)
SNR	Signal to Noise Ratio (Proporción de señal a ruido)
SSIM	Structural Similarity (Índice de similitud estructural)
CNN	Convolutional Neural Network (Red Neuronal Convolucional)
MS-SSIM	Multi-scale Structural Similarity (Índice de similitud estructural)

1. INTRODUCCIÓN

Resumen: *En este capítulo se presenta brevemente los antecedentes del objeto de estudio, justificación del objeto de estudio, justificación y la definición del problema.*

1.1. Antecedentes

Múltiples artículos presentando métodos de reducción de ruido individuales, o comparando un método a otros conocidos existen y son publicados constantemente. Éstos se tomaron en cuenta tanto como parte de la investigación del estado del arte, como los propios datos por analizar.

1.2. Justificación

Al momento de realizar este trabajo, no se encontró ningún artículo publicado que compare más de 5 métodos, ni con la intención de presentar un metaanálisis amplio y profundo. La mayoría de los artículos se limitan a presentar estadísticas de sus propios métodos y compararlos con métodos convenientes para sus métricas.

1.3. Problema

Las ondas electromagnéticas emitidas por sensores activos viajan en fase. Después de interactuar con el objetivo, pueden ya no están en fase, aunque se mantengan coherentes en frecuencia. Esto puede suceder por diversos factores, como la diferencia en la distancia que recorren las ondas de regreso desde diferentes objetivos, o la dispersión de un solo rebote contra múltiples rebotes; incluso el movimiento de la antena sintética misma puede generar ese desfase.

Las ondas desfasadas interfieren constructiva o destructivamente, causando una variación de intensidad de píxel, que se manifiesta como un patrón granular tipo "sal y pimienta" ("*salt and pepper*" en inglés) o "*speckle*".

Al presentarse este ruido, aparecen dificultades para comprender, manipular y utilizar estas imágenes, sobre todo al tratarse de los detalles finos en éstas [1].

Objetivos

1.3.1. Objetivo General:

Encontrar los métodos que presenten mejor eficacia en la reducción de ruido y buscar patrones de comportamiento entre grupos de éstos, separados según su método de computación.

1.3.2. Objetivos Específicos:

Generar una recomendación sólida sobre qué métodos utilizar en cualquier implementación futura.

1.4. Novedad científica, tecnológica o aportación

Se busca aportar una referencia canónica para el problema, bajo el contexto de imágenes SAR. Y ponerlo a disposición del público en general para futuras implementaciones.

2. ESTADO DEL ARTE O DE LA TÉCNICA

Resumen: *En este capítulo se presenta un resumen de los trabajos relacionados con la eficacia de distintos métodos de reducción de ruido en imágenes.*

2.1. Artículos sobre métodos de reducción de ruido

Dada la naturaleza del metaanálisis, la investigación del estado del arte contiene a su vez los datos de estudio.

Se recolectaron estadísticas de rendimiento de 59 métodos descritos en 17 artículos científicos recientes. Estos métodos varían en su manera de acercarse al problema: filtros convolucionales, filtros de promedio y aprendizaje, por ejemplo; así como sus bancos de imágenes de prueba.

Algunos de estos artículos presentan métodos individualmente, mientras que otros presentan comparativas con 2 o 3 métodos propios o referenciados [2-21].

2.1.1. Anatomía de un artículo

La mayoría de los artículos comparte una estructura similar, presentando el contexto necesario para comprender el objetivo de cada uno de estos métodos. Esta información incluye introducciones al problema, ejemplos numéricos o gráficos; algunos incluso contienen pseudocódigo. Junto a esta información, incluyen los resultados obtenidos al probar su método propuesto y a su vez, comparaciones con métodos conocidos (algunos definidos en otros artículos referenciados en este mismo trabajo).

A continuación, algunos ejemplos:

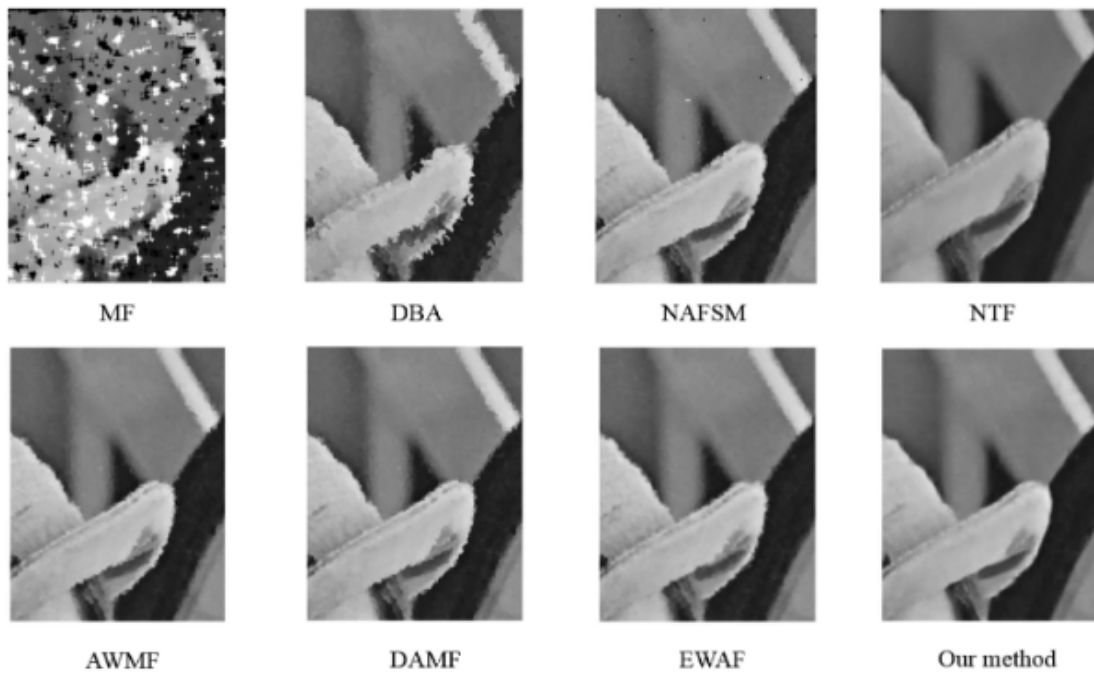


Figura 1. Comparativa visual de métodos de reducción de ruido [9].

```

1: Input: image with salt and pepper noise:  $y$ 
2: Compute the size of  $y$ :  $M \times N$ 
3: Initialization: count=0
4: for  $y(i, j)$  in  $y$  do
5:   if  $y(i, j) == 0 || y(i, j) == 255$  then
6:     record the index  $(i, j)$  in  $P$ , and let the pixel be 0
7:      $count = count + 1$ 
8:   end if
9: end for
10: Compute the noise density of  $y$ :  $\rho = \frac{round(\frac{(10-count)}{MN})}{10}$ 
11: Use a detector[19] to estimate the number of the misclassified
    points: :  $num$ 
12: if  $(num/MN) \leq threshold$  then
13:   if  $\rho \leq 0.5$  then
14:     choose: denoiser(0.50), get network output:  $Y$ 
15:   end if
16:   if  $0.5 < \rho \leq 0.7$  then
17:     choose: denoiser(0.70), get network output:  $Y$ 
18:   end if
19:   if  $0.7 < \rho$  then
20:     choose: denoiser(0.90), get network output:  $Y$ 
21:   end if
22: end if
23: if  $(num/MN) > threshold$  then
24:   if  $\rho \leq 0.3$  then
25:     choose: denoiser(0.10), get network output:  $Y$ 
26:   end if
27:   if  $0.3 < \rho$  then
28:     choose: denoiser(0.90), get network output:  $Y$ 
29:   end if
30: end if
31: Backfill the pixels in  $y$  which are not 0 and 255 into the output

```

Figura 2. Pseudocódigo de un método en específico [9].

Mean PSNR and SSIM of several algorithms for the 15 traditional images with different SPN ratios.

Algorithm	Criterion	10%	20%	30%	40%	50%	60%	70%	80%	90%
PSMF	PSNR	31.61	28.38	25.29	22.25	18.19	12.60	9.17	7.46	6.08
	SSIM	0.9028	0.8715	0.8018	0.6988	0.4903	0.1882	0.0633	0.0318	0.0139
DBA	PSNR	32.82	29.05	26.18	23.72	21.49	19.27	17.09	14.81	12.16
	SSIM	0.9079	0.8664	0.8097	0.7376	0.6521	0.5552	0.4567	0.3623	0.2937
MDBUTMF	PSNR	31.24	28.05	26.68	26.73	26.84	26.18	24.86	21.42	14.07
	SSIM	0.8841	0.7994	0.7443	0.7657	0.7963	0.7880	0.7501	0.6443	0.3052
NAFSM	PSNR	33.88	31.05	29.30	28.04	26.95	25.93	24.89	23.59	20.78
	SSIM	0.9147	0.8916	0.8669	0.8409	0.8124	0.7796	0.7403	0.6872	0.5736
DAMF	PSNR	37.48	34.14	31.95	30.30	28.91	27.63	26.32	24.80	22.71
	SSIM	0.9253	0.9113	0.8946	0.8752	0.8523	0.8244	0.7892	0.7398	0.6572

Tabla 1. Resultados numéricos de métodos listados en un sólo artículo [11].

2.1.2. Artículos destacables

Durante esta investigación 6 artículos destacaron particularmente sobre todos los demás independientemente de sus resultados en materia de eficiencia. Éstos artículos presentan soluciones innovadoras y relevantes en el campo de reducción de ruido.

1. **Fu et al.:** El artículo presenta un método que utiliza métodos de *deep learning*, específicamente redes convolucionales para la reducción de ruido. Probando su eficacia. El patrón de varianza entre métodos de filtrado y de inteligencia está presente entre sus resultados [15].
2. **Neeru y Kaur:** Presentan filtros de decisión basados en reducciones asimétricas. Sus resultados son **muy favorables** y su banco de pruebas es satisfactorio. Sus métodos se comportan mejor que la mayoría que otros métodos, incluyendo tradicionales y de inteligencia artificial [3].
3. **Zhu et al.:** Presenta un regularizador hiper-Laplaciano, sus resultados muestran una recuperación de información considerable. Sin embargo, mantiene sólo niveles bajos de densidad de ruido en sus pruebas [12].
4. **Limshuebchuey et al.:** El artículo presenta resultados de varios procesos de *Deep Learning*, incluyendo los más populares como ResNet y U-Net. El patrón de comportamiento entre estos es vital para comprender el beneficio de éstos contra métodos de filtrado en imágenes con alto porcentaje de ruido [10].

2.1.3. Administración de los Datos

Los datos se recopilaron en un sólo archivo separado por comas. En el archivo se permiten valores nulos.

2.1.4. Exclusión durante la recopilación

Algunos métodos fueron excluidos durante la fase de recopilación por alguna de las siguientes razones:

1. No se mencionan indicadores en el artículo
2. El artículo no contiene información sobre el set de datos o banco de imágenes fueron utilizaron para la evaluación
3. El artículo no indica la densidad de ruido del set de datos o banco de imágenes
4. El artículo no define el tipo de ruido.

2.1.5. Inconsistencias en los bancos de prueba

Cada artículo define su banco de pruebas arbitrariamente. No hay dos artículos que hayan probado sus métodos con las mismas imágenes.

Para fines de este metaanálisis, se tomarán indiscriminadamente dependiendo de su banco de pruebas y se incluyen métodos con ruido tipo "*speckle*", "*salt and pepper*" y "*gaussiano*". En la sección de trabajos futuros se propone estandarizar la generación de ruido en próximas comparaciones.

2.1.6. Artículos con múltiples resultados

En los casos dónde aparecen resultados correspondientes a distintos bancos de pruebas en un sólo artículo, se tomaron los métodos más pertinentes a partir de:

1. El ruido es tipo "*speckle*" o "*salt and pepper*".
2. Las imágenes pertenecen a un banco conocido.
3. La longitud del banco de pruebas; entre más extenso, mejor.

2.1.7. Ranking por banco de pruebas

Con motivo de discriminar los métodos se generó un ranking ponderado otorgados por la relevancia de su banco de pruebas.

Conjunto	Ranking
BSD300	3
Promedio de 11 imágenes aleatorias	7
Lena 512	9
SAR 3x(160x160) & 3X(420x420)	4
40 TESTIMAGES	5
Una imagen individual	12
Promedio de 14 imágenes y BSD300	1
Escaneo CT individual	13
BDS500	2
200 ultrasonidos	6
House 256	11
Lin 256	10
Una imagen SAR individual	8

Tabla 2. Ranking de bancos de imágenes

"BSD300", "BSD500", "TESTIMAGES" son bancos de imágenes estándar, conocidos en la industria. A su vez, "Lena", "Lin" y "House" son imágenes individuales estándar.

3. MARCO TEÓRICO/CONCEPTUAL

Resumen: *En este capítulo se presentan las bases teóricas y conceptuales sobre la eficacia de distintos métodos de reducción de ruido en imágenes.*

3.1. Radar de Apertura Sintética (SAR)

Tipo de recolección activa de datos donde un sensor produce su propia energía y luego registra la cantidad de esa energía reflejada después de interactuar con la Tierra. Mientras que las imágenes ópticas son similares a una fotografía, los datos SAR responden a características de la superficie como la estructura, materiales y la humedad [22].

3.2. Ruido

Un error que se superpone a una señal verdadera. El ruido puede ser aleatorio o sistemático [23].

3.2.1. Ruido Gaussiano

Cuando una variación eléctrica sigue una distribución gaussiana, como en el caso del movimiento térmico mencionado anteriormente, se le llama ruido gaussiano [24].

3.2.2. Ruido “*Salt and Pepper*”

El ruido “*Salt and Pepper*” se refiere a una amplia variedad de procesos que resultan en la misma degradación básica de la imagen: solo unos pocos píxeles son ruidosos, pero son muy ruidosos. El efecto es similar a esparcir puntos blancos y negros, sal y pimienta, sobre la imagen [25].

3.3. Reducción de Ruido

La reducción de ruido sirve para mejorar la señal en una imagen al disminuir las imperfecciones inherentes del sistema de imagen, como píxeles pegados en la cámara o manchas no específicas; la reducción de ruido se puede lograr de varias formas, incluyendo: funciones morfológicas, contracción, expansión, filtros de promedio o mediana y filtros de señal wavelet [1].

3.4. Filtros por Promedio

El filtro por promedio es uno de los filtros de estadística de orden más conocidos debido a su buen rendimiento para algunos tipos específicos de ruido como el ruido “Gaussiano”, “aleatorio” y “sal y pimienta”. Según el filtro por promedio, el píxel central de un vecindario $M \times M$ se reemplaza por el valor mediano de la ventana correspondiente [1].

3.5. Filtros Wavelet

Una transformada wavelet (WT) es una descomposición de una señal en un conjunto de funciones base que consisten en contracciones, expansiones y traslaciones. Puede calcularse mediante una convolución repetida de la señal con la wavelet elegida mientras esta se traslada a lo largo de la dimensión temporal, para sondear la variación en el tiempo, y mientras se estira o comprime, para sondear diferentes frecuencias.

Para reducir el ruido, se hace uso de la matriz tiempo-frecuencia-amplitud creada por la transformada wavelet. Se basa en la suposición de que el ruido no deseado se separará de la señal deseada por sus rangos de frecuencia [26].

3.6. Filtros Wiener

El filtro de Wiener se utiliza en el procesamiento de imágenes para eliminar el ruido y el desenfoque de una imagen. Es un filtro de magnitud que se utiliza frecuentemente como herramienta para filtrar el ruido multiplicativo [27].

3.7. Difusión Anisotrópica

También conocida como difusión de Perona-Malik, es una técnica utilizada en el procesamiento de imágenes para reducir el ruido mientras se preservan características importantes como los bordes. Es particularmente útil en tareas como la eliminación de ruido y la detección de bordes [28].

3.8. Filtros Morfológicos

Los filtros morfológicos son una clase de técnicas de procesamiento de imágenes utilizadas para procesar y analizar las formas y estructuras dentro de las imágenes. A diferencia de los filtros lineales, los morfológicos funcionan sondeando la imagen con un *kernel estructurante* para extraer o suprimir características basadas en su forma y tamaño [29].

3.9. Aprendizaje Automático

El aprendizaje automático (ML, por sus siglas en inglés) es una rama de la inteligencia artificial (AI) y la ciencia de la computación que se centra en el uso de datos y algoritmos para permitir que la AI imite la forma en que los humanos aprenden, mejorando gradualmente su precisión. Por lo general, estos se basan en algoritmos de decisión a partir de una función de costo y son entrenados bajo un proceso de optimización [30].

3.9.1. Aprendizaje Profundo

Un segundo acercamiento a aprendizaje automático se diferencia al poder usar conjuntos de datos etiquetados, también conocidos como aprendizaje supervisado, informan a su algoritmo, pero no necesariamente requiere un conjunto de datos etiquetados.

El proceso de aprendizaje profundo puede procesar datos no estructurados en su forma original (por ejemplo, texto o imágenes), y puede determinar automáticamente el conjunto de características que distinguen las diferentes categorías de datos entre sí.

Este aprendizaje utiliza redes neuronales profundas, que son sistemas inspirados en el cerebro humano y compuestos por múltiples capas de nodos interconectados que procesan y transforman datos [30].

3.10. Convolución

La convolución es fundamental en el procesamiento de señales, visión por computadora y aprendizaje automático. Aplica un filtro o *kernel* a una imagen de entrada o señal y extrae características relevantes.

Las implementaciones típicas utilizan una operación de ventana deslizante donde el *kernel* se mueve a lo largo de la imagen de entrada. Para cada posición del *kernel* K en una imagen dada A , calculamos el producto punto del kernel y los píxeles de la imagen que se superponen [31].

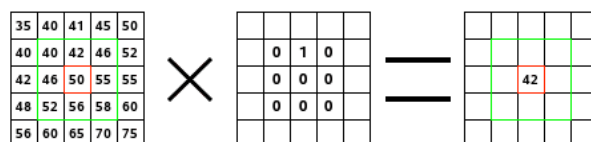


Figura 3. Ejemplo de la operación de convolución [29].

3.11. Matrices de convolución

Las matrices de convolución consisten en valores numéricos dispuestos en una estructura tipo cuadrícula. Estos valores determinan cómo se transforman los datos de entrada mediante la convolución. Cada elemento de los datos de entrada se multiplica por el elemento correspondiente en la matriz de convolución, y los resultados se suman para producir un valor de salida. Repitiendo este proceso en toda la entrada, se obtiene una representación transformada llamada mapa de características [32].

3.12. Redes Convolucionales

Las redes neuronales convolucionales se distinguen de otras redes neuronales por su rendimiento superior con entradas de imágenes, voz o señales de audio.

Tienen tres tipos principales de capas, que son:

- Capa convolucional.
- Capa de agrupación.
- Capa completamente conectada.

La capa convolucional es la primera capa de una red convolucional. Mientras que las capas convolucionales pueden ser seguidas por capas adicionales convolucionales o capas de agrupación, la capa completamente conectada es la capa final. Con cada capa, la CNN aumenta en complejidad, identificando partes más rasgos de la información [32].

3.13. Autoencoder

Los modelos *autoencoder* son un tipo de arquitectura de redes neuronales artificiales utilizados para el aprendizaje no supervisado. El propósito de un *autoencoder* es aprender una representación eficiente, o codificación, de los datos de entrada.

La red consta de un codificador y un decodificador en una sola red neuronal:

3.13.1. Codificador

- Toma los datos de entrada y crea una representación de dimensionalidad reducida.
- La representación de dimensionalidad reducida (codificación) captura las características esenciales de los datos de entrada.

3.13.1. Decodificador

- Toma la representación codificada y reconstruye los datos de entrada originales.
- El objetivo del decodificador es generar una salida que se asemeje a la entrada.

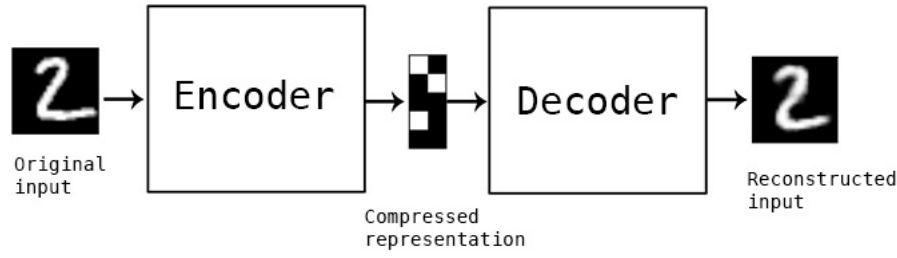


Figura 4. Diagrama arquitectónico de un *autoencoder*.

Algunas de las características principales de los *autoencoder* son:

- Específicos a sus datos - Significa que solo podrán comprimir datos similares a aquellos en los que fueron entrenados.
- *Lossy* (Con Pérdida) - Significa que las salidas descomprimidas serán degradadas en comparación con las entradas originales.
- Aprendizaje no supervisado - No requiere ingeniería adicional, ni datos etiquetados, solo datos de entrenamiento apropiados [33].

3.14. Error Cuadrado Promedio

El valor de MSE denota la diferencia promedio de los píxeles en toda la imagen. Un valor más alto de MSE indica una mayor diferencia entre la imagen original y la imagen procesada.

El error cuadrático medio entre dos imágenes f_1, f_2 es:

$$\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f_1(x, y) - f_2(x, y)]^2$$

[2]

3.15. Relación señal/ruido (SNR)

Define la relación entre la información y el ruido en una imagen, con base en la imagen original limpia. Se puede definir como:

$$SNR_{ms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f_1(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f_1(x, y) - f_2(x, y)]^2}$$

[2]

3.16. Proporción Máxima de Señal a Ruido (PSNR)

Define la relación entre la máxima energía posible de una señal y el ruido que afecta a su representación fidedigna. Debido al rango dinámico, el PSNR se expresa generalmente en escala logarítmica, utilizando como unidad el decibelio. Se puede definir cómo:

$$PSNR = 20 \cdot 10 \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right)$$

Dónde:

MAX_I : Máximo valor que puede tomar un píxel en la imagen.

MSE : Error Cuadrático Medio, promedio de los errores al cuadrado.

[2]

3.17. Índice de preservación de Contornos (EPI)

Índice útil para determinar la preservación del contenido geométrico a partir de sus contornos extraídos por un método de detección de contornos. Se define como:

$$\beta = \frac{\Gamma(\Delta I - \overline{\Delta I}, \widehat{\Delta I} - \overline{\widehat{\Delta I}})}{\sqrt{\Gamma(\Delta I - \overline{\Delta I}, \Delta I - \overline{\Delta I}) \cdot \Gamma(\widehat{\Delta I} - \overline{\widehat{\Delta I}}, \widehat{\Delta I} - \overline{\widehat{\Delta I}})}}$$

Dónde:

$$\Gamma(I_1, I_2) = \sum_{t=0}^K I_1 \cdot I_2$$

$\overline{\Delta I}$ y $\overline{\widehat{\Delta I}}$: Filtro de paso alto de las imágenes 1 y 2, respectivamente, obtenidos con un *kernel* de Laplace de 3x3 píxeles u otro detector de línea.

$\overline{\Delta I}$ y $\overline{\widehat{\Delta I}}$: Valor promedio de la imagen 1 y el valor promedio de la versión filtrada de la imagen 2, respectivamente.

[2]

3.18. Índice de Similitud Estructural (SSIM)

Es un método que facilita la predicción de la calidad percibida de una imagen o transmisión digital. Se puede definir cómo:

$$SSIM = (x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2) + C_2}$$

Dónde:

μ_x : Valor medio de los píxeles de la imagen x .

μ_y : Valor medio de los píxeles de la imagen y .

σ_x^2 : Varianza de la imagen x .

σ_y^2 : Varianza de la imagen y .

σ_{xy} : Covarianza de ambas imágenes.

$c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ Variables estabilizadoras a partir del denominador débil, dónde:

L : Rango dinámico de los píxeles.

$k_1 = 0.01, k_2 = 0.03$: variables estándar [2].

3.18.1. Índice de Similitud Estructural (MS-SSIM)

El método multi-escala divide el proceso en sub-segmentos, dando resultados similares al MS-SSIM [2]

3.19. Herramientas utilizadas

- *Python* - Lenguaje de Programación
- *Jupyter Notebooks* - Framework de presentación de resultados
- *Pandas* - Librería para el manejo de datos
- *Matplotlib* - Librería para la generación de gráficas estadísticas
- *Seaborn* - Librería para la generación de gráficas estadísticas avanzadas
- *ESA Sentinel Toolkit* – Suite para manipular datos SAR
- *ESA SNAP* - Para poder visualizar la información y exportar fragmentos de éstas en un formato manipulable, se utilizó la aplicación *Sentinel SNAP* de la agencia espacial europea.

3.20. Obtención de imágenes SAR

Para la implementación del método de reducción de ruido se obtuvieron los archivos de imágenes SAR del sitio *ASF data search vertex*, de la NASA [34].

3.21. Indicadores utilizados

Para la construcción de la base de datos, dado a que rara vez coinciden en cuáles indicadores utilizan o cómo definen sus divisiones porcentuales, se tomó la decisión de incluir todos los porcentajes de ruido presentados, soportando valores nulos en todas las variables de indicadores. Posteriormente serán filtradas a partir de su pertinencia en el análisis.

Las variables recuperadas son:

Tabla 3. Variables recuperadas de los artículos.

PSNR	SNR	SSIM	MS-SSIM	General
PSNR-10	SNR-10	SSIM-10	MS-SSIM-10	Tipo
PSNR-20	SNR-20	SSIM-20		Tipo de ruido
PSNR-25				DATASET
PSNR-30	SNR-30	SSIM-30	MS-SSIM-30	Año de Publicación
PSNR-40	SNR-40	SSIM-40		Desviación estándar
PSNR-50	SNR-50	SSIM-50	MS-SSIM-50	
PSNR-60	SNR-60	SSIM-60		
PSNR-70	SNR-70	SSIM-70	MS-SSIM-70	
PSNR-80	SNR-80	SSIM-80		
PSNR-90	SNR-90	SSIM-90	MS-SSIM-90	

Las decenas sufixas de cada uno de los indicadores indican el porcentaje de ruido inicial de las imágenes de prueba.

4. DESARROLLO METODOLÓGICO

Resumen: *En este capítulo se presenta en detalle el desarrollo metodológico que incluye el proceso de desarrollo de software para la investigación sobre la eficacia de distintos métodos de reducción de ruido en imágenes.*

4.1. Historial de desarrollo

- Para este desarrollo se comenzó con una investigación amplia sobre el estado del arte, así como una recopilación de la información en una base de datos local.
- Desde el inicio se planeó para funcionar de manera incremental, con forme más información se recolectaba.
- Se trabajó de manera incremental en un proyecto de Python con pandas para el análisis de datos, iterando y generando nuevos resultados cada que aparecía información nueva.
- A partir de las primeras iteraciones de información, se tomó la decisión de implementar una prueba de concepto para buscar replicar el patrón encontrado la información.
- Se construyó un cuaderno de *Jupyter* para entrenar modelos *autoencoder* y se entrenó el modelo de la misma manera, incrementalmente
- Se desarrolló una aplicación interactiva en *streamlit* para probar el modelo con imágenes en tiempo real.
- Se iteró hasta llegar a un estado satisfactorio de información, pasando entonces a registrar los resultados obtenidos.

4.2. Código fuente

El código fuente para el análisis de datos, entrenamiento de modelos *autoencoder* y la aplicación interactiva están disponibles en el siguiente repositorio público:

<https://github.com/fer-lr/mscs-tog>

4.3. Diagrama arquitectónico

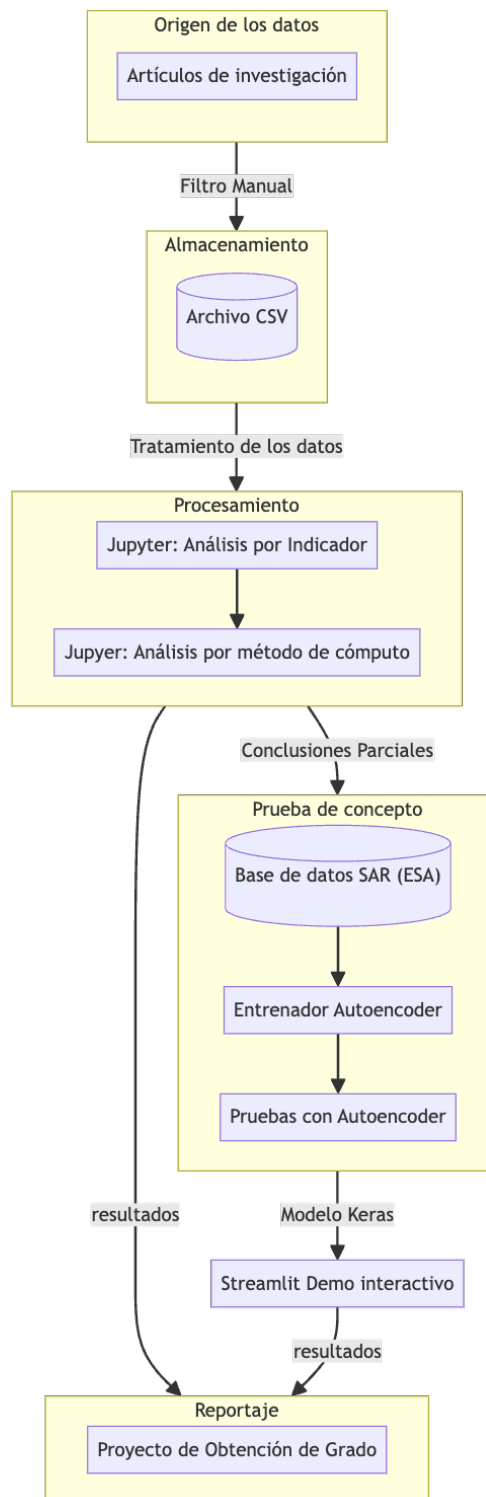


Figura 5. Diagrama Arquitectónico del desarrollo

5. RESULTADOS Y DISCUSIÓN

Resumen: *En este capítulo se presentan los resultados obtenidos del desarrollo de este trabajo y una discusión sobre la eficacia de distintos métodos de reducción de ruido en imágenes.*

5.1. Resultados

5.1.1. Métodos por método computacional

Métodos por método computacional. Los métodos obtenidos se pueden separar en tres grupos principales:

1. Filtros (Promedio, Media, Weiner, Morfológico).
2. Inteligencia Artificial (*Deep Learning*, *CNN*).
3. Procesamiento de Señales (*wavelet*).

La siguiente gráfica describe la distribución de los métodos en los datos.

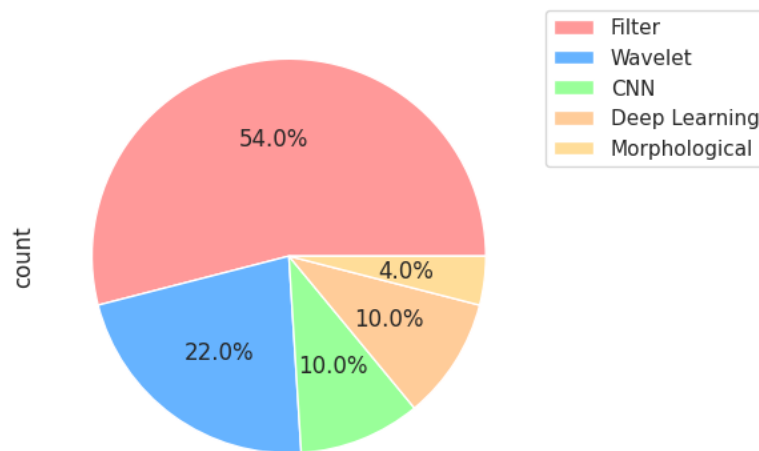


Figura 6. Métodos por método computacional.

5.1.2. Ausencia de datos de *hardware*

En los artículos explorados no se incluyen datos del hardware usado durante sus pruebas.

5.1.3. Análisis comparativo por indicador individual

Se agruparon los métodos por indicador. Para cada uno de éstos, se eliminaron todos los métodos que no contengan datos sobre el indicador en cuestión. A su vez, se elimina cualquier columna que no tenga ninguna entrada.

5.1.3.1. Métodos por PSNR

Estadísticas acumuladas:

Tabla 4. Estadísticas acumuladas por PSNR pt.1.

	PSNR-10	PSNR-20	PSNR-25	PSNR-30	PSNR-40
cuenta	19	25	9	20	12
promedio	46.9693	44.5831	26.1011	37.1724	37.824
desviación	15.686	9.4133	14.0827	10.2573	10.4026
min	15.75	27.55	5.21	25.74	23.71
25%	37.205	43.0134	11.48	29.9875	32.2325
50%	44.6578	44.2014	32.67	32.335	36.2243
75%	57.4297	44.2523	35.62	43.8864	39.7023
máx.	75.1109	64.5097	39.51	61.6224	57.6338

Tabla 5. Estadísticas acumuladas por PSNR pt.2.

	PSNR-50	PSNR-60	PSNR-70	PSNR-80	PSNR-90
cuenta	30	6	10	4	7
promedio	29.0841	26.2567	23.506	20.0975	20.02
desviación	10.3297	6.07699	5.19837	8.59378	7.22666
min	4.32	14.32	9.49	7.55	6.07
25%	26.2675	26.7225	23.105	18.125	17.93
50%	29.6065	28	24.845	23.225	23.56
75%	33.0446	29.3075	26.1075	25.1975	24.365
máx.	53.5684	31.18	28.16	26.39	25.92

Mejores Métodos:

Tabla 6. Mejores métodos por PSNR.

Densidad de Ruido	Valor Máximo	Nombre
10%	75.1109	OCF - CS
20%	64.5097	BPDF - CS
25%	39.51	N2C-Resnet
30%	61.6224	BPDF - CS
40%	57.6338	BPDF - CS
50%	53.5684	BPDF - MC
60%	31.18	YA-CNN
70%	28.16	DAMF
80%	26.39	DAMF
90%	25.92	DFT

Mapa de Calor:

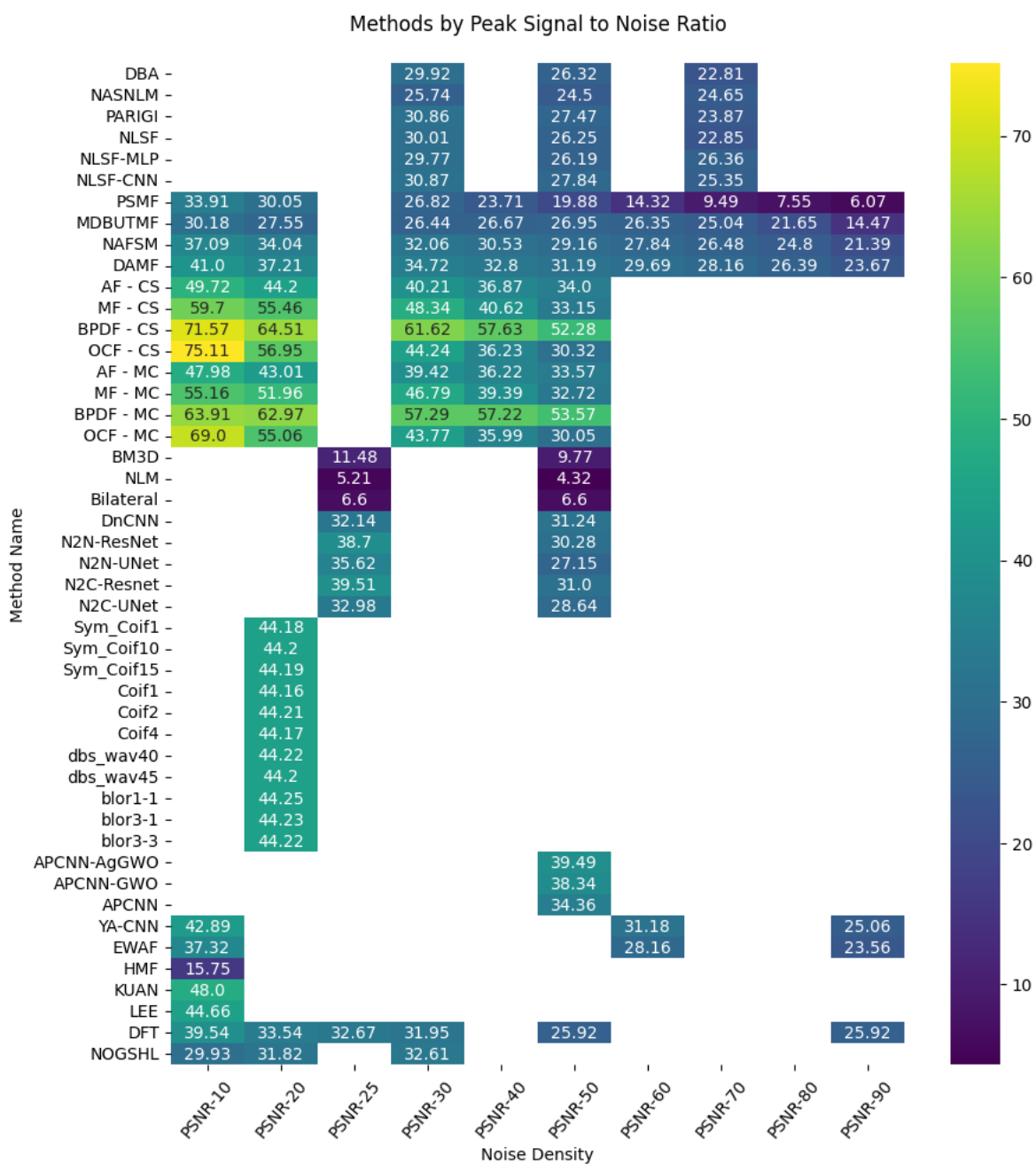


Figura 7. Mapa de color por PSNR.

5.1.3.2. Métodos por SNR

Estadísticas acumuladas:

Tabla 7. Estadísticas acumuladas por SNR.

	SNR-10	SNR-30	SNR-50	SNR-70	SNR-90
cuenta	8	5	5	5	5
promedio	24.925	19.644	13.576	4.878	-2.912
desviación	10.2599	5.75876	7.34608	6.2445	4.37618
min	3.24	9.4	1.05	-4.26	-7.63
25%	24.1475	21.2	13.66	2.48	-5.68
50%	27.17	22.34	15.52	4.89	-4.72
75%	28.8775	22.45	18.64	10.24	1.4
máx.	36.64	22.83	19.01	11.04	2.07

Mejores Métodos:

Tabla 8. Mejores métodos por SNR.

Densidad de Ruido	Valor Máximo	Nombre
10%	36.64	KUAN
30%	22.83	DBUTMPF
50%	19.01	DBUTMPF
70%	11.04	DBUTMPF
90%	2.07	DBUTMPF

Mapa de calor:

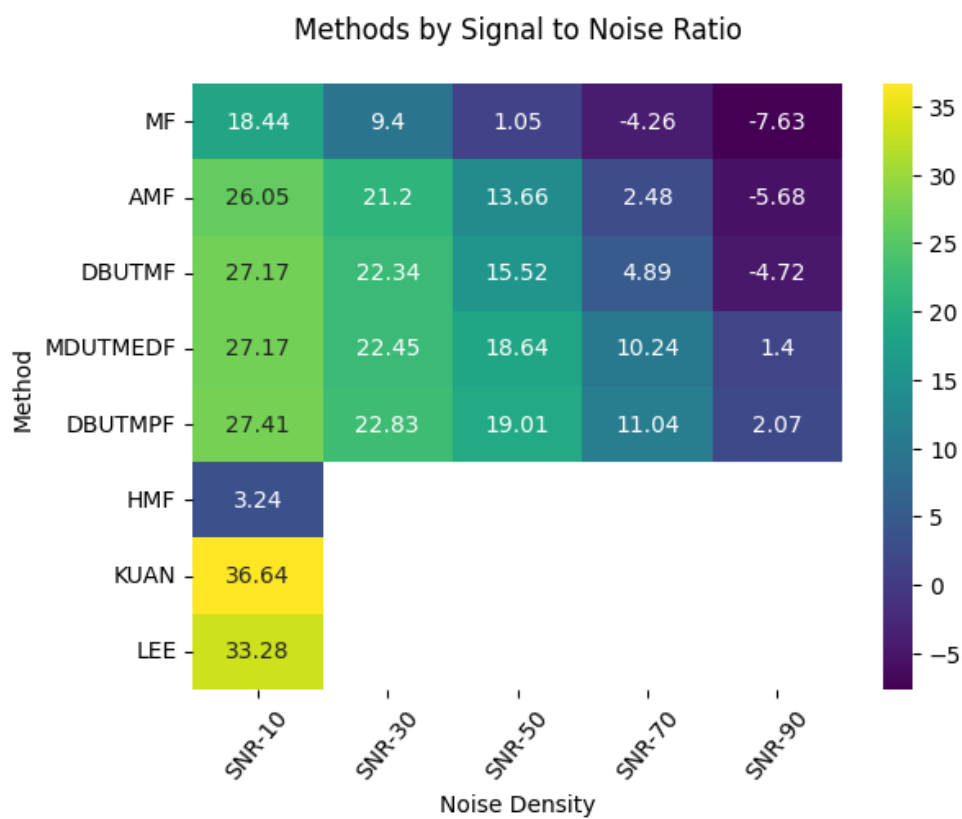


Figura 8. Mapa de calor por SNR.

5.1.3.3. Métodos por EPI

Estadísticas acumuladas:

Tabla 9. Estadísticas acumuladas por EPI.

	EPI-10	EPI-30	EPI-50	EPI-70	EPI-90
cuenta	5	5	5	5	5
promedio	0.77972	0.58414	0.31546	0.08186	0.39926
desviación	0.160888	0.264866	0.17611	0.0555438	0.538576
min	0.4987	0.1118	0.036	0.014	0.0028
25%	0.8023	0.6671	0.29	0.0536	0.0063
50%	0.8313	0.71	0.3194	0.0632	0.0088
75%	0.8823	0.7101	0.4649	0.1384	0.9808
máx.	0.884	0.7217	0.467	0.1401	0.9976

Mejores métodos:

Tabla 10. Mejores métodos por EPI.

Densidad de Ruido	Valor Máximo	Nombre
10%	0.884	DBUTMPF
30%	0.7217	MDUTMEDF
50%	0.467	DBUTMPF
70%	0.1401	DBUTMPF
90%	0.9976	DBUTMPF

Mapa de calor:

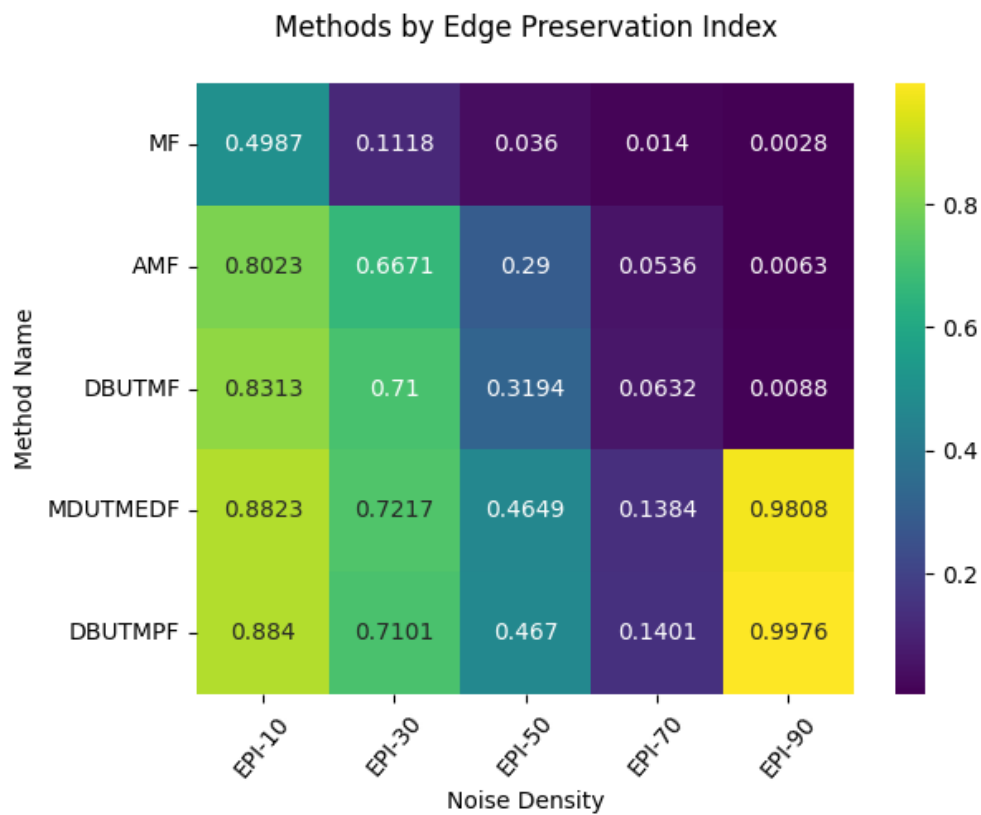


Figura 9. Mapa de calor por EPI.

5.1.3.4. Métodos por SSIM

Estadísticas acumuladas:

Tabla 11. Estadísticas acumuladas por SSIM.

	SSIM-10	SSIM-20	SSIM-30	SSIM-40
cuenta	12	5	10	4
promedio	0.9707	0.91896	0.91393	0.859025
desviación	0.0325803	0.0571034	0.0883591	0.0975365
min	0.8835	0.8349	0.764	0.7435
25%	0.969975	0.9014	0.8621	0.797425
50%	0.98075	0.9164	0.9558	0.86815
75%	0.991275	0.9602	0.97995	0.92975
máx.	0.9976	0.9819	0.9901	0.9563

Mejores métodos:

Tabla 12. Mejores métodos por SSIM.

Densidad de ruido	Valor Máximo	Nombre
10%	0.9976	DBUTMPF
20%	0.9819	DAMF
30%	0.9901	DBUTMPF
40%	0.9563	DAMF
50%	0.9846	APCNN-AgGWO
60%	0.9174	DAMF
70%	0.8885	DAMF
80%	0.8451	DAMF
90%	0.7595	DAMF

Mapa de Calor:

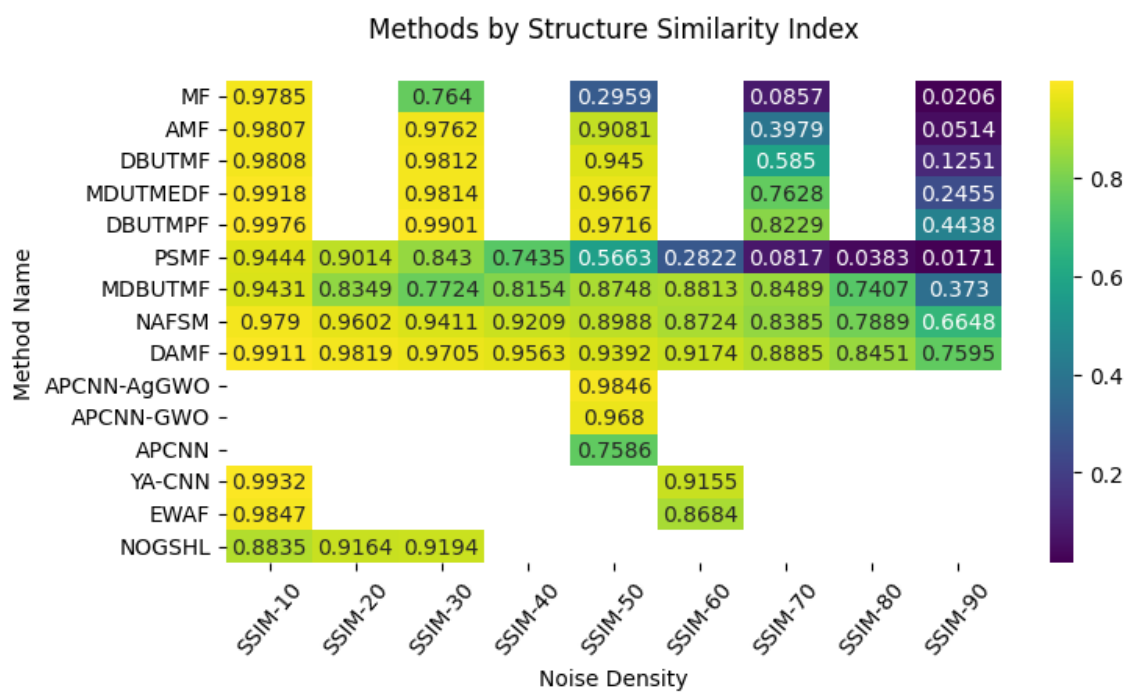


Figura 10. Métodos por SSIM.

5.2. Discusión

5.2.1. Resultados del análisis

El método *Different Applied Median Filter (DAMF)*, tomando en cuenta que se probó con 40 imágenes (hay mejores estadísticas, pero con bancos menos relevantes) se presenta como el mejor método de filtro para PSNR y SSIM.

Mientras que el método *Decision Based Unsymmetric Trimmed Midpoint Filter (DBUTMPF)*, presenta mejores resultados en SNR, y EPI.

A su vez, a través de estos datos, se comienza a revelar un patrón relevante, dónde la eficacia de los métodos de inteligencia artificial no supera a los métodos de filtrado, antes de la marca de 50% de ruido.

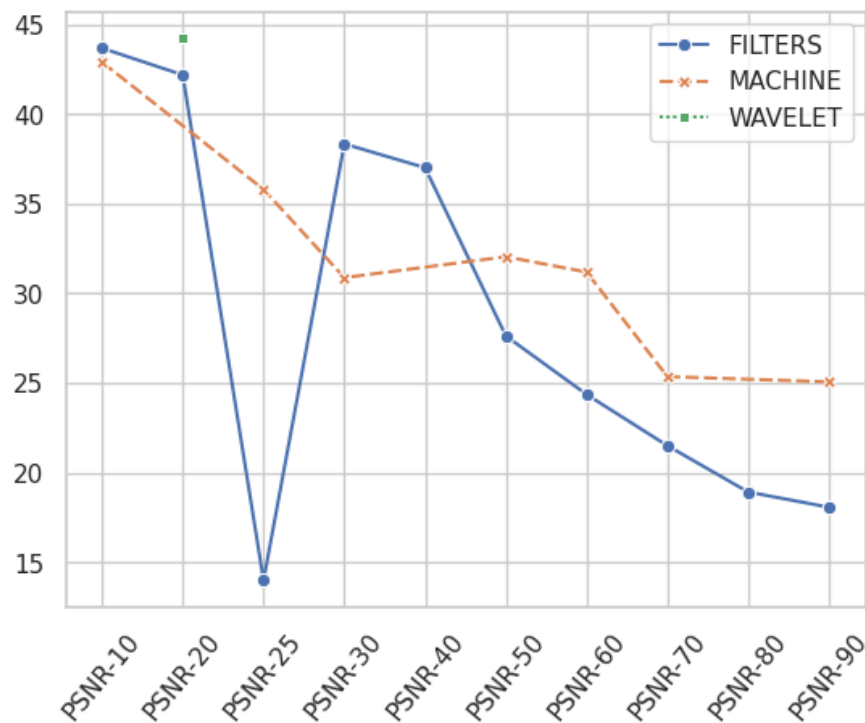


Figura 11. Eficiencia por agrupación de métodos

El patrón revela que mientras exista suficiente información contextual local, los filtros de promedio logran su tarea, sin embargo, a partir de la marca del 50% su eficacia cae mucho más rápido que métodos de *machine learning* y *deep learning*. Esto probablemente se debe al poder de generalización de contexto global de los métodos de *machine learning*.

Sin embargo, a causa de la pobre relación entre la evidencia, dado a los problemas previamente mencionados. El análisis es insuficiente para deducir una conclusión concreta.

5.2.2. Implementación con inteligencia artificial

Con la intención de explorar este resultado aparente, se implementó un método de reducción de ruido sencillo con inteligencia artificial, con la intención de detectar el mismo patrón de eficacia, a pesar de usar un modelo con entrenamiento mínimo.

Se seleccionó un *autoencoder* como tecnología para probar su eficacia contra las estadísticas del mejor método en general por filtro.

5.2.2.1. Descripción del modelo

Tabla 13. Capas del modelo *autoencoder*.

Capa	Dimensiones de salida	Numero de parámetros
input1 (InputLayer)	[None, 256, 256, 3]	0
conv2d (Conv2D)	(None, 256, 256, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 128, 128, 64)	0
batch_normalization (Batch Normalization)	(None, 128, 128, 64)	256
conv2d_1 (Conv2D)	(None, 128, 128, 32)	18464
max_pooling2d_1 (MaxPooling2D)	(None, 64, 64, 32)	0
batch_normalization_1 (Batch Normalization)	(None, 64, 64, 32)	128
conv2d_2 (Conv2D)	(None, 64, 64, 16)	4624
max_pooling2d_2 (MaxPooling2D)	(None, 32, 32, 16)	0
conv2d_3 (Conv2D)	(None, 32, 32, 64)	9280
up_sampling2d (UpSampling2D)	(None, 64, 64, 64)	0
conv2d_5 (Conv2D)	(None, 128, 128, 16)	4624
up_sampling2d_2 (UpSampling2D)	(None, 256, 256, 16)	0
conv2d_6 (Conv2D)	(None, 256, 256, 3)	435

- Parámetros totales: 58067 (226.82 KB)
- Parámetros entrenables: 57875 (226.07 KB)
- Parámetros no entrenables: 192 (768.00 B)

5.2.2.2. Introducción de ruido

Se introdujo ruido *salt and pepper* siguiendo una distribución gaussiana en copias del conjunto de entrenamiento.

```
import random
import cv2

def sp_noise(image, prob):
    output = np.zeros(image.shape, np.uint8)
    thres = 1 - prob
    for i in range(image.shape[0]):
        for j in range(image.shape[1]):
            rdn = random.random()
            if rdn < prob:
                output[i][j] = 0
            elif rdn > thres:
                output[i][j] = 255
            else:
                output[i][j] = image[i][j]
    return output
```

Figura 12. Código fuente para introducir ruido. [35]

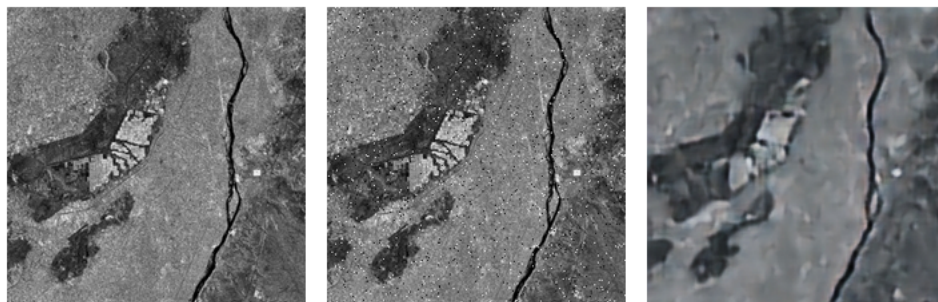
5.2.2.3. Entrenamiento del modelo

El modelo se entrenó con 3,004 imágenes de 3 canales (R, G, B) por 200 épocas. No se utilizaron imágenes generadas por SAR con la intención de que el modelo representara un método de reducción de ruido general.

5.2.2.4. Resultados de las predicciones del modelo

Como primera evidencia, se puede observar que existe una pérdida inicial de información por el hecho de pasar por el *autoencoder*. Que de manera inicial podría representar un problema, sin embargo, en cuanto aumenta la proporción de ruido, comienza a aparecer una ventaja al mantener de manera estable el PSNR, por lo menos hasta el 30% de ruido. Dónde se puede observar una recuperación importante, dado que al ojo humano ya no le es posible identificar ninguna característica de la imagen.

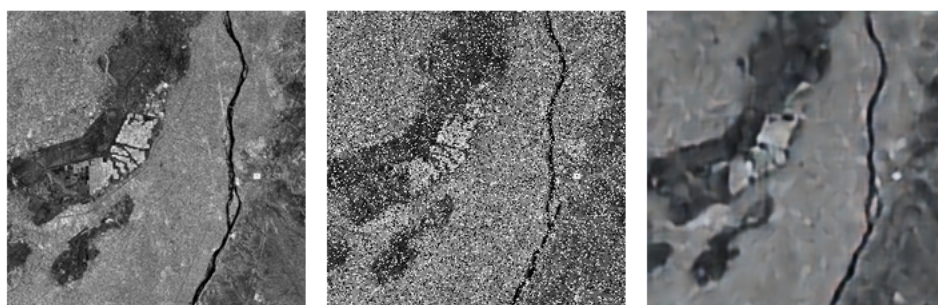
Sin embargo, a partir de 35% en adelante, el modelo deja de poder reconstruir una imagen evidentemente legible, demostrado por las siguientes imágenes:



PSNR: 70.6436%

PSNR: 70.5503%

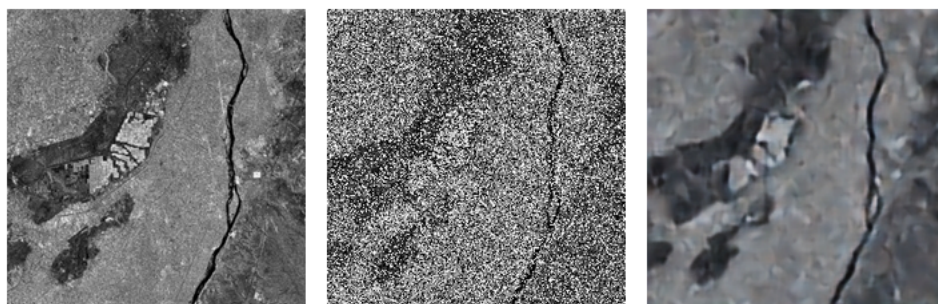
Figura 13. Resultados del autoencoder con 1% de ruido.



PSNR: 60.7068%

PSNR: 70.4980%

Figura 14. Resultados del autoencoder con 10% de ruido.



PSNR: 57.6990%

PSNR: 69.9652%

Figura 15. Resultados del *autoencoder* con 20% de ruido.

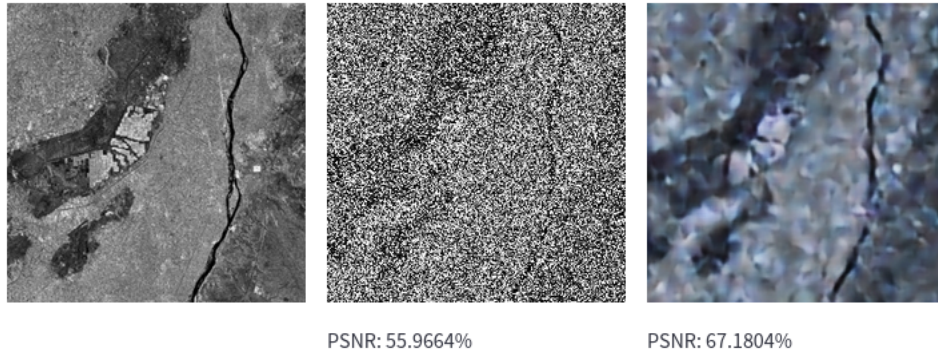


Figura 16. Resultados de *autoencoder* con 30% de ruido.

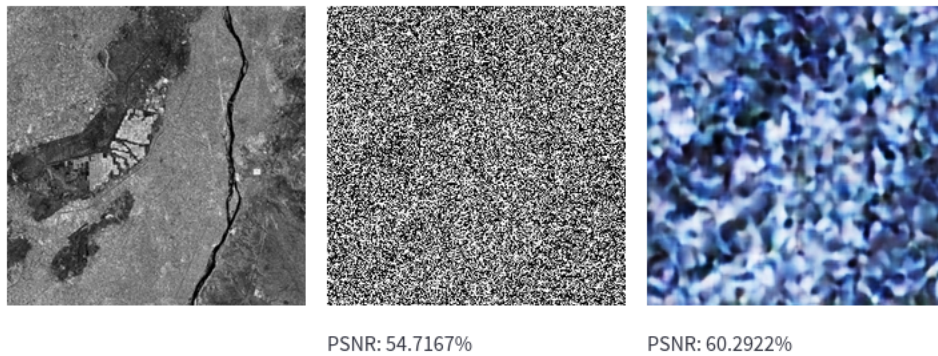


Figura 17. Resultados del *autoencoder* con 40% de ruido.

5.2.3. Indicador Nuevo

Durante la le exploración se encontró una propuesta para medir la eficacia de la reducción de ruido en imágenes particularmente generadas por SAR:

" $\alpha\beta$ -ratio", presentado por Gómez et. al. Se basa en la relación entre dos imágenes a partir de un detector de bordes. Los resultados muestran que el valor de " $\alpha\beta$ " es alto cuando un método es y que es especialmente sensible a la degradación de los bordes después de las operaciones de filtrado [36].

6. CONCLUSIONES

Resumen: *En este capítulo se presentan las conclusiones y trabajo futuro en relación con los resultados obtenidos sobre la eficacia de distintos métodos de reducción de ruido en imágenes.*

.

6.1. Conclusiones

A partir de los resultados obtenidos se pudo concluir lo siguiente:

- Los artículos que describen la eficacia de los métodos de reducción de ruido no coinciden en sus indicadores. Incluso cuando comparten el indicador, no segmentan sus datos en los mismos percentiles. Convirtiendo cualquier comparación entre artículos, en una comparación dispar.
- A su vez, los mismos artículos no coinciden en el tamaño ni el rigor de los bancos de imágenes, convirtiendo sus estadísticas en datos incomparables y no confiables. Particularmente cuando las pruebas son realizadas sobre una sola imagen.
- Para métodos de inteligencia artificial, el contexto es relevante, convirtiendo a los métodos en dependientes de éste. Por lo tanto, un modelo de reducción de ruido para SAR necesitaría entonces ser entrenado bajo el mismo contexto que las imágenes objetivo.

De manera tal, obteniendo las siguientes respuestas:

- De manera preliminar, tomando en cuenta el margen de error al compararlos de manera injusta, los métodos *DBUTMPF* y *DAMF* demostraron tener la mejor eficacia comprobado contra bancos de imágenes satisfactorios.
- Se encontró patrón con los métodos basados en inteligencia artificial, donde su eficacia se reduce de manera más lenta que para los métodos de filtro a partir del 50% de ruido presente en la imagen.

Éstas corresponden a los objetivos del trabajo, así como probando la relevancia de el mismo y sobre todo los trabajos futuros.

6.2. Trabajo Futuro

A partir de esta investigación se proponen los siguientes artículos de investigación:

1. Metaanálisis con mayor rigor y un alcance más amplio con intención de publicación.
2. Pruebas estandarizadas entre los mejores métodos de reducción de ruido por filtrado, procesamiento de señales e inteligencia artificial. Comparados bajo los mismos indicadores, contra el mismo banco de imágenes. (SAR y no SAR)
3. Pruebas de eficiencia y eficacia de métodos de reducción de ruido, con intención de encontrar la mejor combinación de hardware y software basado en su costo-beneficio.
4. Prueba de resistencia al ruido de modelos de clasificación en imágenes SAR. Con la intención de verificar si el ruido "*salt and pepper*" afecta la precisión de los modelos.

BIBLIOGRAFÍA

- [1] F. Qiu, J. Berglund, J. R. Jensen, P. Thakkar, and D. Ren, “Speckle Noise Reduction in SAR Imagery Using a Local Adaptive Median Filter,” *GIScience & Remote Sensing*, vol. 41, no. 3, pp. 244–266, Sep. 2004, doi: 10.2747/1548-1603.41.3.244
- [2] F. Argenti, A. Lapini, T. Bianchi, and L. Alparone, “A Tutorial on Speckle Reduction in Synthetic Aperture Radar Images,” *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 3, pp. 6–35, Sep. 2013, doi: 10.1109/MGRS.2013.2277512
- [3] N. Neeru and L. Kaur, “An Experimental Analysis on Removal of Salt and Pepper Noise from Digital Images,” in *Smart Trends in Information Technology and Computer Communications*, vol. 628, A. Unal, M. Nayak, D. K. Mishra, D. Singh, and A. Joshi, Eds., in *Communications in Computer and Information Science*, vol. 628. , Singapore: Springer Singapore, 2016, pp. 434–441. doi: 10.1007/978-981-10-3433-652
- [4] A. Sharadqh, B. Ayyoub, Z. Alqadi, and J. Al-Azzeh, “Experimental investigation of method used to remove salt and pepper noise from digital color image,” pp. 10–12, Jan. 2018.
- [5] D. Song, Y. Liu, X. Lin, J. Liu, and J. Tan, “Research and Comparison of OCT Image Speckle Denoising Algorithm,” in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China: IEEE, May 2019, pp. 1554–1558. doi: 10.1109/ITAIC.2019.8785813
- [6] A. Bhardwaj, S. Kaur, A. P. Shukla, and M. K. Shukla, “Performance Comparison of De-speckling filters on the Basis of Incremental Iteration in Ultrasound Imaging,” in *2019 International Conference on Power Electronics, Control and Automation (ICPECA)*, New Delhi, India: IEEE, Nov. 2019, pp. 1–5. doi: 10.1109/ICPECA47973.2019.8975532
- [7] J. Zhu, J. Wei, and B. Hao, “Ultrasound images speckle noise removal by nonconvex hybrid overlapping group sparsity model,” *Vis Comput*, Aug. 2022, doi: 10.1007/s00371-022-02627-7
- [8] R. J., M. K., and A. Rajendran, “De-noising of salt and pepper noise using deep learning-based alpha-guided grey wolf optimization,” *Applied Soft Computing*, vol. 130, p. 109649, Nov. 2022, doi: 10.1016/j.asoc.2022.109649.
- [9] Y. Xing, J. Xu, J. Tan, D. Li, and W. Zha, “Deep CNN for removal of salt and pepper noise,” *IET Image Processing*, vol. 13, no. 9, pp. 1550–1560, Jul. 2019, doi: 10.1049/iet-ipr.2018.6004.

- [10] A. Limshuebchuey, R. Duangsoithong, and M. Saejia, "Comparison of Image Denoising using Traditional Filter and Deep Learning Methods," in 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Phuket, Thailand: IEEE, jun. 2020, pp. 193–196. doi: 10.1109/ECTI-CON49241.2020.9158242.
- [11] U. Erkan, L. Gökrem, and S. Enginoğlu, "Different applied median filter in salt and pepper noise," *Computers & Electrical Engineering*, vol. 70, pp. 789–798, Aug. 2018, doi: 10.1016/j.compeleceng.2018.01.019.
- [12] A. M. Abdelmegeed and M. Y. Makkey, "Comparison between Different Methods for Noise Cancellation in Medical Image," *IRJIET*, vol. 05, no. 06, pp. 16–21, 2021, doi: 10.47001/IRJIET/2021.506004.
- [13] J. Zhu, Y. Wei, J. Wei, and B. Hao, "A Non-Convex Hybrid Overlapping Group Sparsity Model with Hyper-Laplacian Prior for Multiplicative Noise," *Fractal Fract*, vol. 7, no. 4, p. 336, Apr. 2023, doi: [10.3390/fractalfract7040336](https://doi.org/10.3390/fractalfract7040336).
- [14] J. Zhu, J. Wei, and B. Hao, "Ultrasound images speckle noise removal by nonconvex hybrid overlapping group sparsity model," *Vis Comput*, Aug. 2022, doi: [10.1007/s00371-022-02627-7](https://doi.org/10.1007/s00371-022-02627-7).
- [15] B. Fu, X. Zhao, Y. Li, X. Wang, and Y. Ren, "A convolutional neural networks denoising approach for salt and pepper noise," *Multimed Tools Appl*, vol. 78, no. 21, pp. 30707–30721, Nov. 2019, doi: 10.1007/s11042-018-6521-4.
- [16] R. H. Chan, C.-W. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1479–1485, Oct. 2005, doi: 10.1109/TIP.2005.852196
- [17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 6064, pp. 354–365, Feb. 2006, doi: 10.1117/12.643267.
- [18] A. Korzynska and M. Iwanowski, "Multistage morphological segmentation of bright-field and fluorescent microscopy image," *Opto-Electronics Review*, vol. 20, pp. 174–186, Oct. 2012, doi: 10.2478/s11772-012-0026-x
- [19] A. W. Santoso, L. Bayuaji, M. Z. Jasni, D. Pebrianti, T. S. Lim, and L. Habibah, "Speckle Noise Reduction Method Based on Fuzzy Approach for Synthetic Aperture Radar Images," 2016, Accessed: Dec. 02, 2022. [Online]. Available: <https://core.ac.uk/display/159191241?source=4>

- [20] J. Delon, A. Desolneux, and T. Guillemot, "PARIGI: a Patch-based Approach to Remove Impulse-Gaussian Noise from Images," *Image Processing On Line*, vol. 6, pp. 130–154, Jun. 2016, doi: 10.5201/ipol.2016.161
- [21] Y. Y. Zhou, Z. F. Ye, and J. J. Huang, "Improved decision-based detail-preserving variational method for removal of random-valued impulse noise," *IET Image Process.*, vol. 6, no. 7, p. 976, 2012, doi: 10.1049/iet-ipr.2011.0312.
- [22] N. Earth Science Data Systems, "What is Synthetic Aperture Radar? | Earthdata." Accessed: Jul. 13, 2024. [Online]. Available: <https://www.earthdata.nasa.gov/learn/backgrounders/what-is-sar>
- [23] E. W. Weisstein, "Noise." Accessed: Jul. 13, 2024. [Online]. Available: <https://mathworld.wolfram.com>
- [24] "Gaussian Noise." Accessed: Feb. 23, 2023. [Online]. Available: <https://www.sfu.ca/sonic-studio-webdav/handbook/GaussianNoise.html>
- [25] "Pepper Noise - an overview | ScienceDirect Topics." Accessed: Jul. 13, 2024. [Online]. Available: <https://www.sciencedirect.com/topics/engineering/pepper-noise>
- [26] "Intro. to Signal Processing:Wavelets and wavelet denoising." Accessed: Jul. 13, 2024. [Online]. Available: <https://terpconnect.umd.edu/~toh/spectrum/wavelets.html>
- [27] R. G. Brown and P. Y. C. Hwang, *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*, 3rd ed. New York: Wiley, 1997.
- [28] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, Jul. 1990, doi: 10.1109/34.56205.
- [29] N. Jamil, T. M. T. Sembok, and Z. A. Bakar, "Noise removal and enhancement of binary images using morphological operations," in *2008 International Symposium on Information Technology*, Kuala Lumpur, Malaysia: IEEE, 2008, pp. 1–6. doi: 10.1109/ITSIM.2008.4631954
- [30] "What Is Machine Learning (ML)? | IBM." Accessed: Jul. 13, 2024. [Online]. Available: <https://www.ibm.com/topics/machine-learning>.
- [31] "Convolution Matrices: Everything You Need to Know When Assessing Convolution Matrices Skills." Accessed: Jul. 13, 2024. [Online]. Available: <https://www.alooba.com/skills/concepts/deep-learning/convolution-matrices>

- [32] B. Fu, X. Zhao, Y. Li, X. Wang, and Y. Ren, “A convolutional neural networks denoising approach for salt and pepper noise,” *Multimed Tools Appl*, vol. 78, no. 21, pp. 30707–30721, Nov. 2019, doi: 10.1007/s11042-018-6521-4.
- [33] “Building Autoencoders in Keras.” Accessed: Nov. 28, 2023. [Online]. Available: <https://blog.keras.io/building-autoencoders-in-keras.html>
- [34] “ASF Data Search.” Accessed: Jul. 13, 2024. [Online]. Available: <https://search.asf.alaska.edu>
- [35] ppk28, “Answer to ‘How to add noise (Gaussian/salt and pepper etc) to image in Python with OpenCV,’” *Stack Overflow*. Accessed: Nov. 28, 2023. [Online]. Available: <https://stackoverflow.com/a/27342545>
- [36] L. Gomez, M. E. Buemi, J. C. Jacobo-Berlles, and M. E. Mejail, “A New Image Quality Index for Objectively Evaluating Despeckling Filtering in SAR Images,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 3, pp. 1297–1307, Mar. 2016, doi: 10.1109/JSTARS.2015.2465167.