

DUnet-MI Corrección de ruido en imágenes médicas implementando redes neuronales artificiales tipo autoencoder

DUnet-MI Noise Correction in Medical Images by Implementing an Autoencoder-Type Artificial Neural Networks

Andrea Paola Ardila Sanchez , Luis Carlos Torres Vega 
Universidad Francisco de Paula Santander, Colombia

Ghiordy Ferney Contreras Contreras 
Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional,
México

Byron Medina Delgado , Dinael Guevara Ibarra 
Universidad Francisco de Paula Santander, Colombia

OPEN  ACCESS

Recibido:
09/04/2024
Aceptado:
10/06/2024
Publicado:
22/07/2024

Correspondencia:
andrea Paolaas@ufps.edu.co

DOI:
<https://doi.org/10.17081/invinno.12.2.6786>



Copyright 2024 by
Investigación e Innovación en
Ingenierías

Resumen

Objetivo: Presentar un método de aprendizaje profundo denominado Denoising Unet for Medical Image, DUnet-MI, enfocado en la corrección de diferentes niveles de ruido en imágenes médicas en las modalidades de Rayos X, Tomografía Computarizada y Resonancia Magnética. **Metodología:** Se aborda una solución para reducir ruido Gaussiano y ruido sal y pimienta, que se suele agregar a la imagen médica por el proceso de obtención, transmisión y/o recepción. DUnet-MI es una adaptación del modelo Unet con variaciones en las capas, filtros e hiperparámetros para corregir ruido e incrementar los valores de las métricas de calidad de relación señal a ruido pico, PSNR por sus siglas en inglés, y la medida del índice de similitud estructural, SSIM por sus siglas en inglés. **Resultados:** se analizó el promedio de 200 imágenes para procesar 7 niveles de ruido. DUnet-MI logró estimaciones promedias de hasta 29,39dB en PSNR y valores de SSIM de 0,890 al procesar ruido Gaussiano y, 38,57dB en PSNR y 0,928 en SSIM al corregir ruido sal y pimienta. **Conclusiones:** DUnet-MI demostró mejoras en las métricas de calidad que superan las obtenidas por los métodos clásicos presentes en la literatura, demostrando su funcionalidad para mejorar la calidad de las imágenes médicas.

Palabras claves: Aprendizaje profundo, Corrección de ruido, Imágenes médicas, Métricas de calidad, Procesamiento a imágenes.

Abstract

Objective: The objective of this work is to present a deep learning method called Denoising Unet for Medical Image (DUnet-MI), focused on correcting different levels of noise in medical images from X-ray, Computed Tomography, and Magnetic Resonance Imaging modalities. **Methodology:** The proposed solution aims to reduce Gaussian noise and salt-and-pepper noise, which are commonly introduced into medical images during the acquisition, transmission, and/or reception processes. DUnet-MI is an adaptation of the Unet model with variations in layers, filters, and hyperparameters to correct noise and enhance the values of quality metrics such as Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM). **Results:** The study analyzed an average of 200 images to process 7 levels of noise. The DUnet-MI method achieved average PSNR estimates of up to 29.39 dB and SSIM values of 0.890 when processing Gaussian noise. Additionally, it achieved average PSNR estimates of 38.57 dB and SSIM values of 0.928 when correcting salt-and-pepper noise. **Conclusions:** DUnet-MI demonstrated improvements in quality metrics surpassing those obtained by classical methods reported in the literature, proving its functionality in enhancing the quality of medical images.

Keywords: Deep Learning, Image Processing, Medical Imaging, Noise Correction, Quality Metrics.

Como citar (IEEE): A.P. Ardila Sanchez, L.C. Torres Vega, G.F. Contreras Contreras, B. Medina Delgado., y D. Guevara Ibarra, "DUnet-MI Corrección de ruido en imágenes médicas implementando redes neuronales artificiales tipo autoencoder", Investigación e Innovación en Ingenierías, vol. 12, no. 2, pp. 21-37, 2024, doi: <https://doi.org/10.17081/invinno.12.2.6786>

Introducción

Hoy en día, existen diferentes métodos de adquisición de imágenes médicas digitales [1]. Entre estos se encuentran aquellos que utilizan la radiación ionizante como la radiografía digital o Rayos X (RX) [2], la Tomografía Computarizada (TC) [3] y sus derivados: Tomografía por Emisión de Positrones (TEP) [4], Tomografía Computarizada por Emisión de Fotón Único (TCEFU) [5], entre otros [6]. Y, aquellos métodos que no requieren radiación ionizante como las ecografías e Imágenes por Resonancia Magnética (IRM) [7]. En general, las imágenes médicas permiten capturar información de la estructura interna del paciente para visualizarla, con el fin de que el personal médico especializado pueda efectuar diferentes diagnósticos [1].

Independientemente del modo de adquisición y obtención de la imagen médica digital suelen ocurrir alteraciones que distorsionan la información al agregar ruido [2, 5]. Este suele tener una apariencia granulada que altera las características de la imagen médica [8], como la textura, forma, estructura y/o bordes representativos, información que se considera relevante al momento de realizar cualquier valoración por parte del especialista [9]. La adición de ruido a una imagen médica se describe formalmente como:

$$X = Y + N, \quad (1)$$

donde X es la imagen ruidosa, Y es la imagen libre de ruido (ground-truth) y N representa el ruido adicionado.

Entre las causas de adición de ruido a una imagen médica se encuentran: las fallas de los equipos y/o sensores electrónicos utilizados en el momento de la adquisición [10], interferencias por factores externos como la luz o emisiones por radiofrecuencia [11], interferencias electromagnéticas [12], presencia de fotones aleatorios cuando se trabaja con radiación ionizante [13], o la baja dosis de radiación que suelen utilizar las modalidades de TC, TEP y TCEFU para evitar afectar la salud del paciente [4, 5].

Con el avance de los diferentes sistemas tecnológicos, han surgido técnicas o sistemas de corrección de ruido enfocadas en mejorar la calidad de la imagen médica digital [9]. Uno de los primeros métodos de corrección de ruido fueron los filtros lineales, como el filtro de media [14], Gaussiano [15] y Wiener [16], que se destacan por suavizar o difuminar la imagen con el fin de corregir el ruido; sin embargo, estos filtros se ven limitados en la conservación de características relevantes [17]. Para solventar este inconveniente surgieron los filtros no lineales, como el filtro de mediana [18], bilateral [19], o aquellos filtros que se basan en lógica difusa [20], los cuales, tienden a conservar las características de la imagen, pero son poco funcionales cuando se abordan distintos tipos de ruido [18].

Recientemente se han destacado los métodos de corrección de ruido basados en aprendizaje profundo, desarrollando algoritmos de redes neuronales que buscan un equilibrio entre la tarea de reducción de ruido y un buen rendimiento del modelo [21]. El autoencoder para eliminación de ruido (DAE, por sus siglas en inglés) [8] reduce la necesidad de configurar manualmente los parámetros de la red y permite una buena restauración de la imagen. Para mejorar el proceso de corrección de ruido enfocado en comportamientos de la vida real, se utilizan Redes Neuronales Convolucionales (CNN, por sus siglas en inglés) [22], estas permiten mejorar la extracción de características reduciendo inconvenientes en términos de gradiente y costo computacional. La unión de DAE y CNN permitió desarrollar modelos de corrección de ruido más potentes aprovechando las ventajas de estos dos métodos [9]. Sin embargo, uno de los inconvenientes presentados por este tipo de redes se debe a la optimización de los hiperparámetros, dado que se ven

limitados por el conocimiento empírico del investigador [23]. Para dar solución a este proceso se ha buscado incorporar algoritmos de optimización que conlleven un mayor costo computacional y tiempo de ejecución [24].

En este documento se presenta un método de corrección de ruido para imágenes médicas denominado DUnet-MI, descrito como:

$$\hat{Y} = M_{\theta}(X), \quad (2)$$

donde \hat{Y} es la imagen estimada, X la imagen a procesar y M el modelo de red neuronal con θ parámetros entrenables. Dicho modelo, como su nombre lo indica, se basa en la arquitectura U-net [25] utilizada en tareas de segmentación de imágenes médicas. Es un autoencoder convolucional que concatena la fase de codificación y decodificación. Las contribuciones de DUnet-MI bajo el conocimiento de los autores se listan a continuación:

- Reconstruye y corrige el ruido en imágenes médicas de RX, TC, IRM, que pueden ser afectadas por ruido blanco aditivo Gaussiano (AWGN, por sus siglas en inglés) o ruido sal y pimienta (SPN, por sus siglas en inglés).
- Mejora la calidad de la imagen médica frente diferentes niveles de ruido utilizando una sola probabilidad en el proceso de entrenamiento.
- Validó la funcionalidad de trabajar con el error cuadrático medio (MSE, por sus siglas en inglés) como función de pérdida [26], donde la convergencia del error fue de $5,0617 \times 10^{-4}$ al utilizar 100 épocas de entrenamiento para el 19,11% de muestras de una base de datos con 5232 imágenes para entrenar. Donde la función de coste utilizada se expresa como:

$$\arg \min_{\theta} \|Y - M_{\theta}(X)\|_2^2. \quad (3)$$

- Demostró que el uso de la función de activación ReLU (Rectified linear unit) [27] en cada una de las capas de la red y el uso de GPU contribuyen a obtener un menor costo computacional y un tiempo de ejecución de 56 segundos por época cuando se utilizaron 7.782.657 de parámetros entrenables.

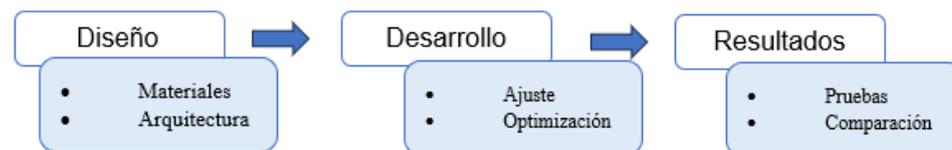
Los resultados fueron evaluados con las métricas de calidad objetivas PSNR y SSIM [28]. Así mismo, se trabajó con el lenguaje de programación Python y con bases de datos de libre acceso y sin restricciones de uso. Finalmente, se llevó a cabo una comparación del modelo desarrollado frente algunos métodos existentes en la literatura.

En el transcurso del documento se presenta: la metodología empleada, los resultados de eficiencia del modelo con base en las métricas PSNR y SSIM, las conclusiones de la investigación y las referencias.

Metodología

En la Figura 1, se presenta de forma simplificada la metodología utilizada en este trabajo, donde se incluye el diseño empleado y el desarrollo llevado a cabo para obtener los resultados.

Figura 1. Metodología propuesta



Fuente: Elaboración propia.

Materiales y recursos

Se optó por trabajar con Python [29] y las librerías Tensorflow [30] y Keras [31], la cuales, permitieron construir cada una de las capas del modelo de red neuronal. Así mismo, para la ejecución, entrenamiento y prueba se utilizó el entorno de Google Colab [32] que brindó acceso a recursos de hardware como CPU, RAM, GPU y almacenamiento.

Bases de datos

Se revisaron diferentes bases de datos (datasets) teniendo en cuenta que fueran de libre acceso. Para la selección se buscó que las imágenes almacenadas se encontraran en un formato estándar (PNG, JPEG, entre otros) que permitiera la visualización desde el equipo donde se estaban desarrollando las pruebas. En esta investigación se utilizaron tres bases de datos que contienen imágenes médicas de RX [33], IRM [34] y TC [35], y se encuentran disponibles en la plataforma Kaggle [36]. La Tabla 1, muestra la distribución de cada una de las bases de datos de acuerdo con la cantidad de imágenes de entrenamiento, prueba y etiquetado correspondiente a la enfermedad.

Tabla 1. Bases de datos de acuerdo con la cantidad de imágenes y enfermedad

Dataset	Etiqueta	Train	Test	Total
Chest X-Ray Images	Normal	1349	234	1583
	Pneumonia	3883	390	4273
	Total	5232	624	5856
Brain tumor classification (MRI)	Glioma	826	100	926
	Meningioma	822	115	937
	No tumor	395	105	500
	Pituitary	827	74	901
	Total	2870	394	3264
Large COVID-19 CT	Covid-19	6074	1519	7593
	Normal	5514	1379	6893
	CAP	2094	524	2618
	Total	13682	3422	17104

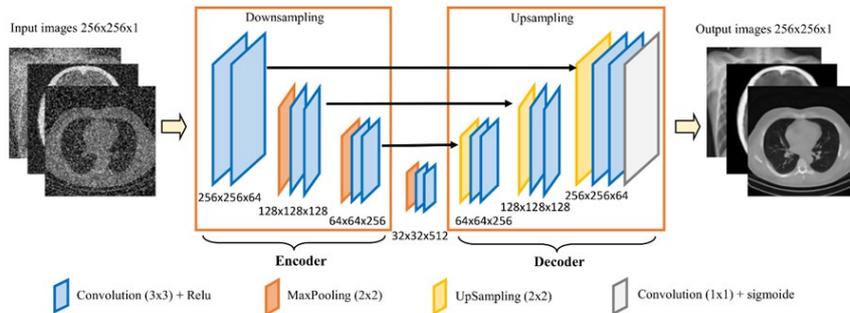
Fuente: Elaboración propia con base en la información de: [33, 34, 35].

El primer dataset seleccionado fue “Chest X-Ray images (Pneumonia)” [33], que contiene un total de 5856 imágenes de RX (anterior-posterior) en formato JPEG. Seguidamente se seleccionó el dataset “Brain Tumor Classification (MRI)” [34], que es una recopilación de diferentes bases de datos en línea que cuenta con un total de 3264 imágenes con secuencias T1, T2 y FLAIR en sus vistas sagital, coronal y axial. Finalmente, se utilizó la base de datos “Large COVID-19 CT scan Slice” [35], esta es una compilación de 7 bases de datos de libre acceso que contiene un total de 17108 imágenes, cuyo tamaño está ajustado a $224 \times 224 \text{ px}^2$. En el apartado de kaggle donde se encuentran las imágenes no dividen las bases de datos entre entrenamiento y prueba, sin embargo, se menciona la importancia de dividir el dataset cuando se trabaja con tareas de deep learning, por lo que se utilizó un 80% para imágenes de entrenamiento y un 20% para imágenes de prueba.

Arquitectura de DUnet-MI

Se propuso un modelo no lineal presentado en la ecuación 2 con base en la arquitectura U-net [25]. La Figura 2, sintetiza y proporciona una visualización de cada uno de los componentes y capas interconectadas para construir la fase de codificación y decodificación del modelo propuesto en este trabajo.

Figura 2. Arquitectura del modelo DUnet-MI. Muestra de imágenes médicas tomadas de: [33, 34, 35]



Fuente: Elaboración propia.

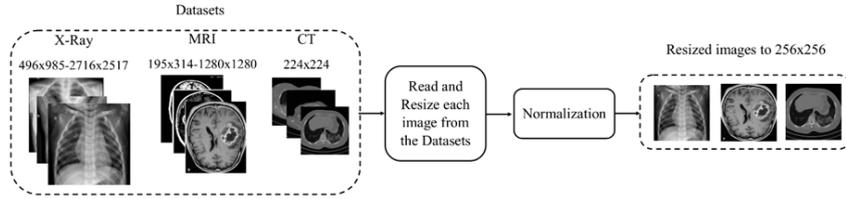
Para la arquitectura presentada en la Figura 2, se establecieron tres capas convolucionales tanto en la fase de codificación como en la de decodificación, donde cada una de las capas tiene dos convoluciones con un kernel de 3×3 y una función de activación de tipo ReLU. Adicionalmente, las convoluciones poseen padding de un píxel para mantener el tamaño de la imagen, de modo que las dimensiones de entrada y salida sean iguales ($256 \times 256 \text{ px}^2$). Así mismo, en la etapa de Encoder se agregó MaxPooling 2×2 para ir reduciendo las dimensiones de las imágenes a la mitad a medida que se duplica la cantidad de filtros en cada capa.

En el proceso de decodificación, se utilizan capas de Upsampling 2×2 seguidas de cada convolución para ir aumentando el tamaño de la imagen hasta lograr las dimensiones de entrada, en este caso, la cantidad de filtros va disminuyendo a medida que aumenta el tamaño de la imagen. Se añaden capas de BatchNormalization seguidas de la última convolución y dropout de 0.25 entre cada convolución con la finalidad de reducir los efectos del overfitting y obligar a que la red aprenda y no memorice la información de los datos de entrenamiento. Así mismo, la última capa del decoder posee una convolucional adicional con un kernel de 1×1 con función de activación sigmoide para asegurar que las imágenes de salida se encuentren en el rango establecido. Una vez compilada la arquitectura se obtuvieron 7.782.657 parámetros entrenables.

Preparación de la base de datos

Dadas las características de las bases de datos, fue necesario ajustar las imágenes tanto para la fase de entrenamiento como para la de prueba. La Figura 3 muestra de forma sintetizada el proceso realizado para llevar a cabo la respectiva modificación.

Figura 3. Síntesis del proceso de lectura y redimensionamiento de los datos. Muestra de imágenes médicas tomadas de: [33, 34, 35]



Fuente: Elaboración propia

La lectura y redimensionamiento de la imagen se realizó con la librería Open-CV [37]. Se trabajaron imágenes a escala de grises (monocromáticas) ajustadas a un tamaño de $256 \times 256 \text{ px}^2$, lo que permitió reducir el costo computacional y, por ende, el tiempo de ejecución. Se utilizó el método de interpolación cv2.InterLinear que se ajusta a cualquier modificación de tamaño, factor importante, debido a que las dimensiones de las imágenes de RX e IRM varían considerablemente ($496 \times 985 \text{ px}^2$ a $2716 \times 2517 \text{ px}^2$; $195 \times 314 \text{ px}^2$ a $1280 \times 1280 \text{ px}^2$, respectivamente). Las imágenes se almacenaron en una lista que posteriormente se convirtió en un array de Numpy [38] de tipo float64. Dichas imágenes se encuentran normalizadas a valores de intensidad de píxel en el rango de 0 a 1 para que puedan ser procesadas por la red.

Optimización de la red neuronal

Para optimizar el modelo de red neuronal propuesto, se utilizó la función de coste de la ecuación 3. Al trabajar con MSE se pudo disminuir significativamente el error al momento de entrenar llegando a un valor mínimo de $5,0617 \times 10^{-4}$ al comparar la imagen estimada respecto con la imagen libre de ruido. La métrica MSE se define formalmente como:

$$MSE = \frac{1}{l} \sum_{i=1}^l (Y - \hat{Y})_2^2, \quad (4)$$

donde l es el número total de muestras que se analizan. Dada esta función, se seleccionó el optimizador ADAM modificando el learning rate a 3×10^{-3} para el ajuste de los parámetros de la red (pesos y sesgos).

Una vez optimizada la red, se ajustaron las condiciones de entrenamiento, indicando que las imágenes con ruido se comportan como entrada y las imágenes aparentemente libres de ruido como salida. En este caso de investigación, se trabajó con un número pequeño de datos de entrenamiento, seleccionando un total de 1000 imágenes para cada modalidad. De igual manera, se utilizaron 92 imágenes de validación, 100 épocas de entrenamiento y un tamaño de lote de 10. Estos valores fueron establecidos de manera empírica buscando optimizar los recursos de RAM y GPU disponibles, evitando la saturación del sistema y logrando tiempos de entrenamientos con 56 segundos por época.

Resultados

Los resultados se evaluaron con la métrica PSNR descrita como:

$$PSNR[dB] = 10 \log_{10} \left(\frac{\max_i^2}{MSE} \right) \quad (5)$$

donde \max_i representa el valor de máxima intensidad del píxel que depende directamente del tipo de dato utilizado. Adicionalmente, se utilizó la métrica SSIM:

$$SSIM = \frac{(2\mu_y\mu_x+c_1)(\sigma_{y,x}+c_2)}{(\mu_y^2+\mu_x^2+c_1)(\sigma_y^2+\sigma_x^2+c_2)}, \quad (6)$$

donde μ_y y σ_y son la media y desviación estándar de la imagen de referencia; mientras que μ_x y σ_x son la media y desviación estándar de la imagen con ruido o estimada por la red, dependiendo de qué dato se esté comparando; finalmente, c_1 y c_2 son los coeficientes de estabilidad [28]. Para implementar las métricas mencionados se utilizaron las funciones de la librería Scikit-image [39, 40] manejando la siguiente estructura: $PSNR(Y, \hat{Y})$ o $PSNR(Y, X)$ y $SSIM(Y, \hat{Y})$ o $SSIM(Y, X)$. Cabe mencionar que entre más alto sea el valor de PSNR y SSIM más acertada es la estimación de la red, teniendo en cuenta que para la métrica SSIM el valor máximo posible es 1.

Se trabajaron dos tipos de ruidos para apreciar la respuesta del modelo DUnet-MI al procesar las tres modalidades de imágenes médicas (RX, IRM y TC). En el caso de AWGN se estableció $\mu = 0$, mientras que σ fue el único parámetro modificable para variar el nivel de ruido presente en la imagen. Así mismo, toda la escala de σ utilizada en este documento se encuentra dividida en 255 con el fin de poder trabajar con las imágenes normalizadas que están en un rango de 0 a 1 y, comparar con modelos de la literatura que normalmente trabajan escalas de 10 a 80. Para trabajar con SPN se buscaron valores promedios equivalentes con los niveles de AWGN utilizados para evaluar la respuesta bajo condiciones similares. La Tabla 2 muestra las equivalencias obtenidas de manera experimental al comparar el PSNR promedio de 200 imágenes consideradas para pruebas.

Tabla 2. Equivalencias de AWGN y SPN obtenidas con la métrica PSNR

Ruido	Equivalencia						
	20	30	40	50	60	70	80
AWGN (σ)	20	30	40	50	60	70	80
SPN ($Prob$)	0,015	0,032	0,055	0,082	0,113	0,146	0,180

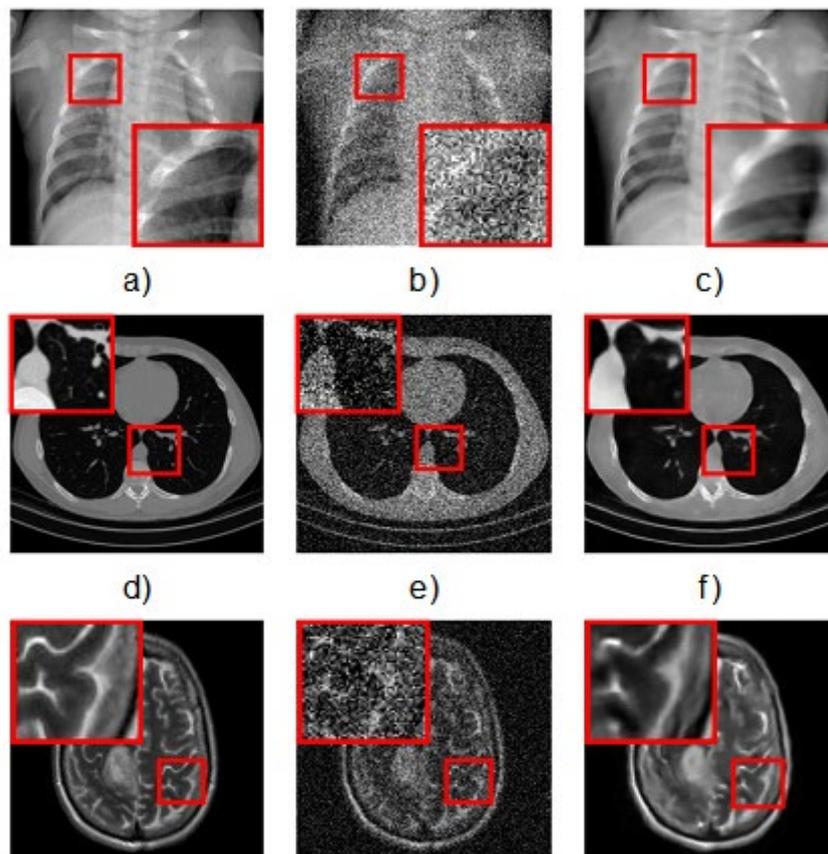
Fuente: Elaboración propia

Una vez establecidas las escalas se realizan las primeras pruebas de funcionamiento del modelo, en este caso se distorsionan las imágenes médicas con AWGN a $\sigma=50$, valor utilizado al momento de entrenar la red. La Figura 4, relaciona la imagen médica aparentemente libre de ruido, la imagen afectada por AWGN y la estimación que brinda el método DUnet-MI para cada una de las modalidades utilizadas en este trabajo. Se agregan los valores de PSNR y SSIM en la descripción de la figura con el fin de analizar cuantitativamente la mejora en la calidad. En cada una de las imágenes se resalta una pequeña parte en color rojo para apreciar con detalle la reconstrucción que realiza el modelo, se muestra como se ve afectada la imagen por los pixeles que contienen ruido y como después de procesarla dichos pixeles se reconstruyen logrando similitudes con la imagen original.

En todos los casos analizados, la imagen se recupera reduciendo el ruido y, por ende, aumentando los valores de las métricas utilizadas. De acuerdo con la información suministrada por la Figura 4, se puede afirmar que para AWGN el modelo brinda una mejor estimación cuando procesa una imagen de RX, dado que, se obtiene un valor de 30,34dB mejorando la calidad hasta 15,22dB en la métrica de PSNR, valor que está por encima de los dB obtenidos en las otras modalidades analizadas. De igual manera, es relevante mencionar que, aunque la imagen de TC obtuvo un valor más alto de SSIM en comparación con la imagen de RX e IRM, la ganancia

de la imagen de RX es superior, debido a que se obtiene una diferencia de 0,723 lo que equivaldría a un 72,3% de recuperación de la imagen según el índice de similitud estructural.

Figure 4. Pruebas unitarias de DUnet-MI con AWGN a $\sigma = 50$. Evaluado con las métricas *PSNR/SSIM*. a) RX, b) RX con ruido (15,12dB/0,187), c) RX estimada (30,34dB/0,910), d) TC, e) TC con ruido (15,87dB/0,248), f) TC estimada (29,72dB/0,925), g) IRM, h) IRM con ruido (15,97dB/0,174), i) IRM estimada (30,06/0,907).



Fuente: Elaboración propia

Se realiza la Tabla 3 con la intención de apreciar la respuesta del modelo frente a diferentes niveles de ruido (σ) al corregir imágenes médicas afectadas por AWGN. Según la información obtenida, se reafirma lo mencionado en las pruebas iniciales, destacando que para este tipo de ruido el modelo DUnet-MI da una mejor respuesta cuando trabaja con imágenes de RX. Para todos los niveles de ruido utilizados en este estudio los valores de las métricas aumentaron, lo que indica que la red desarrollada es capaz de corregir imágenes médicas independientemente de la cantidad de ruido que estas posean, incluso cuando se entrena con un único valor de σ .

Por otro lado, se resaltan las estimaciones e incrementos (Δ) significativos en cada una de las modalidades utilizadas, notando que el valor más alto de las estimaciones no está siempre relacionado con la mayor recuperación de la imagen médica. En el caso de imágenes de RX se obtuvieron estimaciones elevadas cuando se trabajó con un nivel de ruido de $\sigma=40$, que es inferior al valor usado para el entrenamiento, mientras que la mayor recuperación al comparar la imagen ruidosa

y la estimada se presentó cuando se procesaron imágenes con $\sigma=60$. Caso similar ocurrió con las imágenes de TC. Mientras que, para las IRM la mayor recuperación o ganancia se presentó cuando se trabajó con el mismo nivel de ruido utilizado en el entrenamiento y las estimaciones con mejor valor de PSNR y SSIM se obtuvieron para $\sigma=40$ y $\sigma=50$ respectivamente.

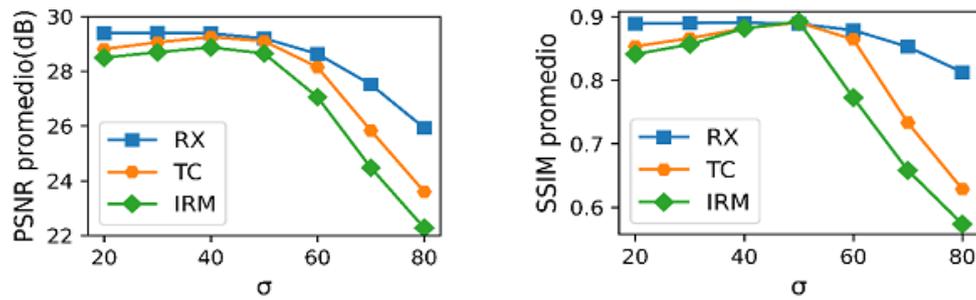
Tabla 3. Valores promedios de PSNR (dB) y SSIM al evaluar 200 imágenes con AWGN. Valor más alto de las estimaciones mostrado en rojo y mayor incremento mostrado en azul

σ	Métrica	RX			TC			IRM		
		Ruido	Est.	Δ	Ruido	Est.	Δ	Ruido	Est.	Δ
20	PSNR	22,36	29,39	7,030	23,3	28,81	5,510	23,09	28,50	5,410
	SSIM	0,543	0,889	0,346	0,536	0,853	0,317	0,545	0,841	0,296
30	PSNR	18,95	29,40	10,45	19,93	29,06	9,130	19,71	28,70	8,990
	SSIM	0,366	0,890	0,524	0,372	0,866	0,494	0,393	0,856	0,463
40	PSNR	16,60	29,37	12,77	17,56	29,25	11,69	17,37	28,88	11,51
	SSIM	0,260	0,890	0,630	0,272	0,882	0,610	0,230	0,882	0,652
50	PSNR	14,87	29,19	14,32	15,77	29,10	13,33	15,59	28,65	13,06
	SSIM	0,194	0,889	0,695	0,207	0,890	0,683	0,238	0,892	0,654
60	PSNR	13,52	28,64	15,12	14,35	28,13	13,78	14,2	27,06	12,86
	SSIM	0,151	0,879	0,728	0,163	0,864	0,701	0,196	0,773	0,577
70	PSNR	12,45	27,51	15,06	13,19	25,83	12,64	13,05	24,47	11,42
	SSIM	0,122	0,852	0,73	0,132	0,733	0,601	0,165	0,658	0,493
80	PSNR	11,59	25,94	14,35	12,21	23,61	11,40	12,10	22,28	10,18
	SSIM	0,101	0,813	0,712	0,109	0,629	0,520	0,141	0,574	0,433

Fuente: Elaboración propia

La Figura 5 muestra las curvas obtenidas en relación al nivel de ruido, el PSNR y SSIM obtenido. En dichas curvas se aprecia de manera simplificada como en todos los casos la respuesta de la red al trabajar con RX es superior en comparación con las otras modalidades de imágenes médicas. Así mismo, se destaca el hecho de que para valores de σ por encima de 50 las estimaciones de TC e IRM tienden a disminuir, lo que indica que, para estas dos modalidades, DUnet-MI presenta mayor dificultad al corregir AWGN cuando se trabaja con valores muy altos o por encima del entrenamiento.

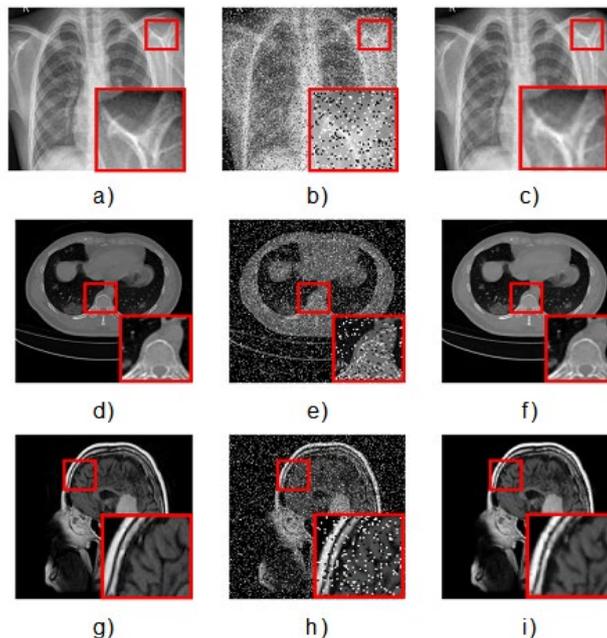
Figura 5. Valores promedio del modelo DUnet-MI con AWGN. a) PSNR, b) SSIM



Fuente: Elaboración propia

Luego de analizar el comportamiento de DUnet-MI frente a imágenes distorsionadas con AWGN, se presentan los resultados al procesar imágenes con SPN. Se evalúa una imagen específica de las bases de datos con una probabilidad de ruido de 0,146 que corresponde a uno de los valores expuestos en la escala de equivalencias de la Tabla 2. La Figura 6, muestra la relación entre las imágenes aparentemente limpias, las imágenes afectadas con SPN y las estimaciones de DUnet-MI. El modelo fue capaz de reconstruir satisfactoriamente las características representativas aislando la información que no era propia de la imagen médica. Para este tipo de ruido el modelo brinda resultados superiores cuando procesa TC logrando estimaciones de hasta 35,29dB en PSNR y un 97,3% de similitud estructural para la probabilidad de ruido utilizada.

Figure 6. Pruebas unitarias de DUnet-MI con SPN a $prob = 0,146$. Evaluado con las métricas $PSNR/SSIM$, a) RX, b) RX con ruido (13,87dB/0,186), c) RX estimada (33,64dB/0,959), d) TC, e) TC con ruido (12,23dB/0,162), f) TC estimada (35,49dB/0,972), g) IRM, h) IRM con ruido (12,31dB/0,180), i) IRM estimada (33,92dB/0,964).



Fuente: Elaboración propia

De acuerdo con la información contenida en la Tabla 4, los valores más altos en las estimaciones se obtuvieron para las probabilidades de ruido más bajas, logrando imágenes en promedio con un 98,8% de similitud con base en la métrica SSIM al comparar la imagen original con la estimada por DUNet-MI. A medida que aumenta la probabilidad de ruido van disminuyendo los valores promedios de PSNR y SSIM, sin embargo, todas las estimaciones están por encima de los 32dB incluso cuando se aborda el nivel más alto de probabilidad utilizado. Con este tipo de ruido la mayor ganancia o mejora en la imagen se evidenció al procesar la imagen con el nivel de ruido de 0,146 cuando se utilizó RX y el nivel de 0,180 en TC e IRM, logrando ganancias de hasta 23,10dB en PSNR y una mejora del 83,4% en SSIM.

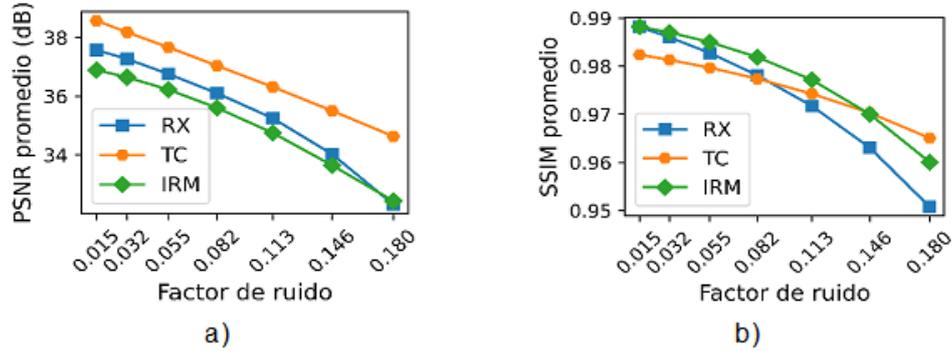
Tabla 4. Valores promedios de PSNR (dB) y SSIM al evaluar 200 imágenes con SPN. Valor más alto de las estimaciones mostrado en rojo y mayor incremento mostrado en azul.

Prob	Métrica	RX			TC			IRM		
		Ruido	Est.	Δ	Ruido	Est.	Δ	Ruido	Est.	Δ
0,015	PSNR	23,42	37,57	14,15	22,45	38,57	16,22	22,62	36,90	14,28
	SSIM	0,760	0,988	0,228	0,760	0,928	0,168	0,777	0,988	0,211
0,032	PSNR	20,04	37,27	17,23	19,05	38,18	19,13	19,24	36,65	17,41
	SSIM	0,577	0,986	0,409	0,573	0,981	0,395	0,602	0,987	0,385
0,055	PSNR	17,96	36,80	19,13	16,72	37,67	20,95	16,89	36,23	19,34
	SSIM	0,418	0,983	0,565	0,409	0,980	0,571	0,447	0,985	0,538
0,082	PSNR	15,94	36,14	20,20	14,97	37,04	22,07	15,14	35,62	20,48
	SSIM	0,305	0,978	0,673	0,293	0,977	0,684	0,334	0,982	0,648
0,113	PSNR	14,56	35,21	20,65	13,58	36,31	22,73	13,75	34,73	20,98
	SSIM	0,229	0,972	0,743	0,214	0,974	0,760	0,267	0,977	0,710
0,146	PSNR	13,45	34,04	20,59	12,47	35,51	23,04	12,65	33,66	20,98
	SSIM	0,178	0,963	0,785	0,164	0,970	0,806	0,205	0,970	0,765
0,180	PSNR	12,52	32,34	19,82	11,54	34,64	23,10	11,72	33,43	21,71
	SSIM	0,144	0,951	0,807	0,131	0,965	0,834	0,170	0,960	0,790

Fuente: Elaboración propia

La red neuronal desarrollada fue capaz de dar una buena respuesta a las diferentes probabilidades de SPN utilizadas para RX, TC e IRM. Sin embargo, como se muestra en la Figura 7, las estimaciones más altas se obtuvieron para las imágenes de TC e IRM según la métrica PSNR y SSIM respectivamente. Así mismo, al procesar imágenes médicas con este tipo de ruido no se presenta un mayor descenso cuando se trabaja con niveles o densidades de ruido que son superiores a la utilizadas en el entrenamiento.

Figura 7. Curvas de los valores promedio del modelo DUNet-MI con SPN. a) PSNR, b) SSIM



Fuente: Elaboración propia

Para finalizar el proceso de evaluación, se propone comparar DUNet-MI con modelos clásicos presentes en la literatura, en el caso de AWGN, se destacan los filtros lineales y no lineales por la capacidad que tienen de suavizar y/o mejorar la imagen de manera rápida. En la Tabla 5, se utiliza una imagen de cada una de las bases de datos con tres valores de σ para apreciar cómo mejora después de pasarla por cada uno de los métodos. Cuando se trabajó RX con $\sigma=20$ el filtro Wiener ofreció un valor de PSNR y SSIM superior en comparación con los otros métodos, sin embargo, para $\sigma=40;60$ DUNet-MI superó la respuesta de todos los filtros, demostrando que a diferencia de los métodos clásicos es capaz de responder a probabilidades más elevadas de ruido. En las modalidades de TC e IRM el modelo DUNet-MI logró los mejores resultados, adicionalmente la diferencia en PSNR y SSIM entre los niveles de ruido no es tan significativa, es decir, la respuesta del modelo no decrece de manera abrupta al corregir niveles más elevados de ruido, caso contrario a lo que sucede con los modelos clásicos, que con una σ baja, la respuesta es aceptable, pero cuando el nivel de ruido aumenta los valores de las métricas decrecen considerablemente.

Tabla 5. Comparación de DUNet-Mi con filtros clásicos utilizados en la literatura corrigiendo AWGN con $\sigma = \{20; 40; 60\}$

Imagen	σ	Ruido	Gaussiano	Media	Mediana	Wiener	DUNet-MI
XR	20	22,6/0,52	30,7/0,89	29,7/0,86	30,3/0,90	31,0/0,90	30,4/0,88
	40	16,8/0,24	26,1/0,75	25,0/0,68	26,8/0,78	25,8/0,74	30,9/0,89
	60	13,7/0,13	22,8/0,61	21,7/0,52	24,1/0,66	22,4/0,61	30,4/0,91
CT	20	22,4/0,54	30,8/0,88	29,9/0,86	30,4/0,89	30,5/0,89	31,1/0,92
	40	16,7/0,25	26,3/0,75	25,1/0,67	26,9/0,79	25,8/0,75	30,9/0,93
	60	13,7/0,14	23,1/0,62	22,0/0,54	24,2/0,66	22,6/0,62	29,5/0,89
MRI	20	22,4/0,51	29,5/0,84	28,7/0,81	28,4/0,87	30,3/0,84	29,2/0,92
	40	16,5/0,23	25,6/0,70	24,5/0,64	26,0/0,76	25,6/0,70	32,4/0,95
	60	13,4/0,13	22,9/0,59	21,7/0,50	23,7/0,64	22,7/0,58	31,3/0,93

Fuente: Elaboración propia

Se propone realizar una comparación similar trabajando con imágenes afectadas por SPN. Según la literatura, uno de los filtros que da buena respuesta a este tipo de ruido es el filtro de mediana. Esto se reafirma en la Tabla 6, donde logra corregir las tres probabilidades utilizadas alcanzando incrementos en la calidad de 14dB en PSNR y 0,74 en SSIM. Sin embargo, aunque la respuesta del filtro de Mediana fue buena, el modelo DUnet-MI tiene la capacidad de reconstruir más características de la imagen alcanzando valores de 41,1dB en PSNR y 99% de similitud estructural que se ven reflejadas en mejoras de la calidad de la imagen médica con una ganancia de 23,9dB cuando se utiliza la métrica PSNR. Por otro lado, aunque los demás filtros utilizados tratan de corregir imágenes distorsionadas con SPN, no poseen la capacidad de dar valores por encima de los 26,6dB en PSNR y 0,77 en SSIM cuando trabajan con densidades de ruido altas, dado que los valores obtenidos van decreciendo significativamente. Esto no sucede con DUnet-MI puesto que, para las tres modalidades de imágenes médicas y las tres probabilidades de ruido alcanzó valores por encima de los 36dB en PSNR y 0,97 en SSIM.

Tabla 6. Comparación de DUnet-Mi con filtros clásicos utilizados en la literatura corrigiendo SPN con $Prob = \{0,015; 0,055; 0,113\}$

Imagen	Prob.	Ruido	Gaussiano	Media	Mediana	Wiener	DUnet-MI
RX	0,015	23,0/0,75	29,9/0,89	29,1/0,87	29,9/0,94	25,4/0,80	39,9/0,99
	0,055	17,5/0,10	26,6/0,77	25,5/0,72	29,9/0,94	22,7/0,67	38,4/0,98
	0,113	14,4/0,22	23,8/0,65	22,6/0,58	29,6/0,94	21,4/0,59	36,9/0,97
TC	0,015	22,3/0,75	30,2/0,88	29,4/0,86	30,6/0,93	23,9/0,76	37,9/0,98
	0,055	16,6/0,38	25,9/0,71	24,8/0,65	30,6/0,93	21,2/0,58	37,4/0,98
	0,113	13,5/0,17	22,5/0,54	21,5/0,47	30,4/0,93	19,8/0,48	36,3/0,97
IRM	0,015	22,6/0,75	29,8/0,89	92,8/0,88	30,6/0,97	25,3/0,82	41,1/0,99
	0,055	17,1/0,41	26,0/0,72	25,2/0,68	30,3/0,97	22,1/0,66	40,1/0,99
	0,113	14,0/0,23	23,0/0,58	22,1/0,53	29,9/0,97	20,4/0,57	37,9/0,98

Fuente: Elaboración propia

Conclusión

La corrección de ruido en imágenes médicas permitió aumentar la calidad de la misma con base en los puntajes de PSNR y SSIM, permitiendo observar detalles significativos necesarios al momento de efectuar un diagnóstico por parte de un especialista. El modelo DUnet-MI presenta un método basado en aprendizaje profundo capaz de corregir diferentes niveles o intensidades de AWGN y SPN. Para las pruebas se tomaron 200 imágenes de tres bases de datos con RX, TC e IMR que suelen verse afectadas por los tipos de ruidos mencionados y, se corrigieron 7 niveles o intensidades de ruido, logrando valores promedios de PSNR de hasta 29,37dB y 38,57dB y, un 89,2% y 98,8% de similitud estructural (SSIM) al procesar imágenes con AWGN y SPN, respectivamente. Los resultados al corregir los dos tipos de ruido varían considerablemente, aunque la cantidad de ruido estudiada fue similar, esto se debe a la manera en la que se manifiesta el ruido en la imagen y como dicho ruido afecta los píxeles. En las pruebas individuales, los valores de las métricas pueden estar por encima o por debajo de los valores promedios mencionados, sin embargo, en la mayoría de los casos la respuesta fue funcional

cuando se entrenó con 1000 imágenes y 100 épocas, demostrando que se puede corregir el ruido mejorando la calidad de la imagen con incrementos de hasta 15,12dB y 23,10dB para AWGN y SPN al trabajar con pocas imágenes y épocas de entrenamiento. El uso de funciones de activación ReLU brinda una mejor propagación del gradiente evitando su desvanecimiento, está en conjunto con GPUs modernas ayudan a acelerar el procesamiento de la red mejorando su rendimiento y disminuyendo el tiempo de ejecución, alcanzando un promedio de 56 segundos por época para el modelo propuesto. Por último, el modelo fue desarrollado con recursos de libre acceso que se encuentran disponibles en la web, por lo que los resultados pueden ser replicados sin necesidad de efectuar costos adicionales.

Referencias bibliográficas

- [1] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan, "Medical Image Analysis using Convolutional Neural Networks: A Review," *J Med Syst*, vol. 42, no. 11, p. 226, Oct. 2018. DOI: <https://doi.org/10.1007/s10916-018-1088-1>.
- [2] V.-G. Nguyen, "Digital Radiography with a Consumer Camera: Image Denoising and Deblurring," *IEIE Transactions on Smart Processing & Computing*, vol. 10, no. 5, pp. 398–406, Oct. 2021. DOI: <https://doi.org/10.5573/IEIESPC.2021.10.5.398>.
- [3] A. Ribeiro *et al.*, "Ionising radiation exposure from medical imaging – A review of Patient's (un) awareness," *Radiography*, vol. 26, no. 2, pp. e25–e30, May 2020. DOI: <https://doi.org/10.1016/j.radi.2019.10.002>.
- [4] I. Domingues, G. Pereira, P. Martins, H. Duarte, J. Santos, and P. H. Abreu, "Using deep learning techniques in medical imaging: a systematic review of applications on CT and PET," *Artif Intell Rev*, vol. 53, no. 6, pp. 4093–4160, Nov. 2020. DOI: <https://doi.org/10.1007/s10462-019-09788-3>.
- [5] K. Kim and Y. Lee, "Improvement of signal and noise performance using single image super-resolution based on deep learning in single photon-emission computed tomography imaging system," *Nuclear Engineering and Technology*, vol. 53, no. 7, pp. 2341–2347, Jul. 2021. DOI: <https://doi.org/10.1016/j.net.2021.01.011>.
- [6] S. V. Mohd Sagheer and S. N. George, "A review on medical image denoising algorithms," *Biomed Signal Process Control*, vol. 61, p. 102036, Aug. 2020. DOI: <https://doi.org/10.1016/j.bspc.2020.102036>.
- [7] S. U. Khan, N. Ullah, I. Ahmed, I. Ahmad, and M. I. Mahsud, "MRI Imaging, Comparison of MRI with other Modalities, Noise in MRI Images and Machine Learning Techniques for Noise Removal: A Review," *Current Medical Imaging Formerly Current Medical Imaging Reviews*, vol. 15, no. 3, pp. 243–254, Feb. 2019. DOI: <https://doi.org/10.2174/1573405614666180726124952>.
- [8] W.-H. Lee, M. Ozger, U. Challita, and K. W. Sung, "Noise Learning-Based Denoising Autoencoder," *IEEE Communications Letters*, vol. 25, no. 9, pp. 2983–2987, Sep. 2021. DOI: <https://doi.org/10.1109/LCOMM.2021.3091800>.
- [9] A. Saleh Ahmed, W. H. El-Behaidy, and A. A. A. Youssif, "Medical image denoising system based on stacked convolutional autoencoder for enhancing 2-dimensional

gel electrophoresis noise reduction,” *Biomed Signal Process Control*, vol. 69, p. 102842, Aug. 2021. DOI: <https://doi.org/10.1016/j.bspc.2021.102842>.

- [10] G. Contreras, J. Pabon, H. Garcia, F. Rojas, and H. Arguello, “Correction of Designed Compressive Spectral Imaging Measurements Using a Deep Learning-Based Method,” in *2021 XXIII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*, IEEE, Sep. 2021, pp. 1–5. DOI: <https://doi.org/10.1109/STSIVA53688.2021.9592024>.
- [11] B. Goyal, A. Dogra, S. Agrawal, and B. S. Sohi, “Noise Issues Prevailing in Various Types of Medical Images,” *Biomedical and Pharmacology Journal*, vol. 11, no. 3, pp. 1227–1237, Sep. 2018. DOI: <https://doi.org/10.13005/bpj/1484>.
- [12] O. Gulenko *et al.*, “Deep-Learning-Based Algorithm for the Removal of Electromagnetic Interference Noise in Photoacoustic Endoscopic Image Processing,” *Sensors*, vol. 22, no. 10, p. 3961, May 2022. DOI: <https://doi.org/10.3390/s22103961>.
- [13] D. Thanh, P. Surya, and L. M. Hieu, “A Review on CT and X-Ray Images Denoising Methods,” *Informatica*, vol. 43, no. 2, pp. 151–159, Jun. 2019. DOI: <https://doi.org/10.31449/inf.v43i2.2179>.
- [14] ugur erkan, D. N. H. Thanh, L. M. Hieu, and S. Enginoglu, “An Iterative Mean Filter for Image Denoising,” *IEEE Access*, vol. 7, pp. 167847–167859, 2019. DOI: <https://doi.org/10.1109/ACCESS.2019.2953924>.
- [15] N. Kumar and M. Nachamai, “Noise Removal and Filtering Techniques used in Medical Images,” *Oriental journal of computer science and technology*, vol. 10, no. 1, pp. 103–113, Mar. 2017. DOI: <https://doi.org/10.13005/ojctst/10.01.14>.
- [16] M. A. Aslam, M. A. Munir, and D. Cui, “Noise Removal from Medical Images Using Hybrid Filters of Technique,” *J Phys Conf Ser*, vol. 1518, no. 1, p. 012061, Apr. 2020. DOI: <https://doi.org/10.1088/1742-6596/1518/1/012061>.
- [17] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep Learning for Computer Vision: A Brief Review,” *Comput Intell Neurosci*, vol. 2018, pp. 1–13, 2018. DOI: <https://doi.org/10.1155/2018/7068349>.
- [18] S. A. Villar, S. Torcida, and G. G. Acosta, “Median Filtering: A New Insight,” *J Math Imaging Vis*, vol. 58, no. 1, pp. 130–146, May 2017. DOI: <https://doi.org/10.1007/s10851-016-0694-0>.
- [19] M. Elhoseny and K. Shankar, “Optimal bilateral filter and Convolutional Neural Network based denoising method of medical image measurements,” *Measurement*, vol. 143, pp. 125–135, Sep. 2019. DOI: <https://doi.org/10.1016/j.measurement.2019.04.072>.
- [20] N. Chauhan and B.-J. Choi, “Denoising Approaches Using Fuzzy Logic and Convolutional Autoencoders for Human Brain MRI Image,” *INTERNATIONAL JOURNAL of FUZZY LOGIC and INTELLIGENT SYSTEMS*, vol. 19, no. 3, pp. 135–139, Sep. 2019. DOI: <https://doi.org/10.5391/IJFIS.2019.19.3.135>.
- [21] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, “Deep learning on image denoising: An overview,” *Neural Networks*, vol. 131, pp. 251–275, Nov. 2020. DOI: <https://doi.org/10.1016/j.neunet.2020.07.025>.

- [22] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017. DOI: <https://doi.org/10.1109/TIP.2017.2662206>.
- [23] P. Liu, M. D. El Basha, Y. Li, Y. Xiao, P. C. Sanelli, and R. Fang, "Deep Evolutionary Networks with Expedited Genetic Algorithms for Medical Image Denoising," *Med Image Anal*, vol. 54, pp. 306–315, May 2019. DOI: <https://doi.org/10.1016/j.media.2019.03.004>.
- [24] M. Abd Elaziz *et al.*, "Advanced metaheuristic optimization techniques in applications of deep neural networks: a review," *Neural Comput Appl*, vol. 33, no. 21, pp. 14079–14099, Nov. 2021. DOI: <https://doi.org/10.1007/s00521-021-05960-5>.
- [25] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, pp. 234–241, 2015. Doi: https://doi.org/10.1007/978-3-319-24574-4_28.
- [26] N. Zhang, S.-L. Shen, A. Zhou, and Y.-S. Xu, "Investigation on Performance of Neural Networks Using Quadratic Relative Error Cost Function," *IEEE Access*, vol. 7, pp. 106642–106652, 2019. DOI: <https://doi.org/10.1109/ACCESS.2019.2930520>.
- [27] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," *arXiv preprint*, Mar. 2018.
- [28] U. Sara, M. Akter, and M. S. Uddin, "Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study," *Journal of Computer and Communications*, vol. 07, no. 03, pp. 8–18, 2019. DOI: <https://doi.org/10.4236/jcc.2019.73002>.
- [29] G. van Rossum, "Python," 1991. <https://www.python.org/>
- [30] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016*, pp. 265–283, 2016.
- [31] N. Ketkar, "Introduction to Keras," in *Deep Learning with Python*, Berkeley, CA: Apress, 2017, pp. 97–111. DOI: https://doi.org/10.1007/978-1-4842-2766-4_7.
- [32] Google, "Te damos la bienvenida a Colaboratory," *Google Colab*. <https://colab.research.google.com/> (accessed Jul. 02, 2023).
- [33] D. Kermany, K. Zhang, and M. Goldbaum, "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification," vol. 2, 2018. DOI: <https://doi.org/10.17632/RSCBJBR9SJ.2>.
- [34] S. Bhuvaji, A. Kadam, P. Bhumkar, and S. Dedge, "Brain Tumor Classification (MRI)," 2020.
- [35] M. Maftouni, A. Chung Chee Law, and B. Shen, "A Robust Ensemble-Deep Learning Model for COVID-19 Diagnosis based on an Integrated CT Scan Images Database," in *Proceedings of the 2021 IISE Annual Conference*, Institute of Industrial and Systems Engineers (IISE), Jun. 2021, pp. 632–637.

- [36] Kaggle, "Kaggle." <https://www.kaggle.com/>
- [37] G. Bradski, "The OpenCV Library," *Dr. Dobbs's Journal of Software Tools*, vol. 25, no. 11, pp. 122–125, 2000.
- [38] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020. DOI: <https://doi.org/10.1038/s41586-020-2649-2>.
- [39]R. Timarán Pereira, A. Hidalgo Troya, y J. Caicedo Zambrano, "Factores asociados al desempeño académico en Lectura Crítica en las pruebas Saber 11° con árboles de decisión", *Investigación e Innovación en Ingenierías*, vol. 8, n.º 3, pp. 29–37, 2020. DOI: <https://doi.org/10.17081/invinno.8.3.4701>
- [40] S. van der Walt *et al.*, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, Jun. 2014. DOI: <https://doi.org/10.7717/peerj.453>.