

Diseño e implementación de un Autoencoder para la supresión de ruido de diferente naturaleza en imágenes de color

Design and implementation of an Autoencoder for the suppression of noise of different nature in color images

Armando Adrián **Miranda-González**¹, Alberto Jorge **Rosales-Silva**², Dante **Mújica-Vargas**³
Ponciano Jorge **Escamilla-Ambrosio**⁴, Jean Marie **Vianney-Kinani**⁵, Floriberto **Ortiz-Rodríguez**⁶

¹Instituto Politécnico Nacional, MÉXICO
<https://orcid.org/0000-0003-4673-0490> | amirandag1100@alumno.ipn.mx

²Instituto Politécnico Nacional, MÉXICO
<https://orcid.org/0000-0001-8436-3025> | arosaleess@ipn.mx

³Tecnológico Nacional de México, MÉXICO
<https://orcid.org/0000-0001-8665-4096> | dante.mv@cenidet.tecnm.mx

⁴Instituto Politécnico Nacional, MÉXICO
<https://orcid.org/0000-0003-3772-3651> | pescamillaa@ipn.mx

⁵Instituto Politécnico Nacional, MÉXICO
<https://orcid.org/0000-0001-5237-3050> | jkinani@ipn.mx

⁶Instituto Politécnico Nacional, MÉXICO
<https://orcid.org/0000-0002-2608-8077> | flortiz@ipn.mx

Recibido 02-12-2022, aceptado 15-02-2023.

Resumen

El presente artículo se centra en la propuesta de una red neuronal de tipo Autoencoder para la supresión de diversos tipos de ruido presente en imágenes de color. Existen diversos tipos de algoritmos para la supresión de ruido en imágenes como lo son las Redes Neuronales Convolucionales (CNN) los cuales requieren tanto de una gran cantidad de datos para su entrenamiento como de mayor carga computacional, aunque el mayor problema que estos presentan es que comúnmente se centran en un solo tipo de ruido causando que en algunas ocasiones la imagen no sea procesada de manera adecuada y al finalizar su tratamiento contenga píxeles corruptos los cuales causen que haya pérdida de detalles importantes dentro de la imagen. Por este motivo, esta propuesta proporciona evidencia de que el uso de autoencoders para la supresión de diversos tipos de ruidos en imágenes es factible, a través de este artículo se mostrarán resultados subjetivos y objetivos que determinarán la factibilidad del uso de este tipo de redes neuronales.

Palabras clave: autoencoder, convoluciones, redes neuronales, supresión de ruido.

Abstract

This article focuses on the proposal of a neural network called Autoencoder for the suppression of various kinds of noise present in color images. There are several kinds of algorithms for denoising images, such as Convolutional Neural Networks (CNN), which require a large amount of data for their training and a greater computational complexity, although the biggest problem they present is that they commonly focus on a single kind of noise, sometimes causing that the image to not be processed properly and at the end of its treatment it contains corrupted pixels which cause the loss of important details within the image. For this reason, this proposal provides evidence that the use of Autoencoders for the denoising of several kinds of noises is feasible, through this article subjective and objective results will be shown that will determine the feasibility of using this kind of network neural.

Index terms: autoencoder, convolutions, neural networks, denoising.

I. INTRODUCCIÓN

En las últimas décadas, la comunidad científica se ha centrado en el desarrollo de algoritmos capaces de suprimir el ruido presentado en imágenes, los cuales en su mayoría están centrados en resolver la problemática de un solo tipo de ruido [1], [2], [3]. Las fuentes de ruido difieren en cuanto a su tipo y pueden ser generadas principalmente durante la adquisición o transmisión de la imagen, debido a la problemática de la existencia de diversos tipos de ruido y que a su vez la naturaleza de su origen difiere es común que no siempre se utilice un filtro adecuado para la supresión del ruido y que la imagen al finalizar su preprocesamiento aún contenga píxeles corruptos o haya perdido detalles importantes.

Los filtros especializados para la supresión de ruido presente en imágenes se clasifican dependiendo del dominio en que estos trabajen como lo son el dominio espacial y el dominio de la frecuencia [1]. En el dominio espacial se encuentra una nueva división entre filtros lineales y no lineales, estos tipos de filtros en su mayoría solo son útiles para un solo tipo de ruido, por ejemplo, podemos encontrar el filtro de media para ruido gaussiano y el filtro de mediana para ruido impulsivo [4].

El uso de Inteligencia Artificial (*Artificial Intelligence*, AI) dentro del filtrado de imágenes tuvo sus orígenes durante la década de los 50, pero no fue hasta el año 1986 por medio de Rumelhart, Hinton y Williams asentaron las bases de lo que hoy en día conocemos [5].

La AI tiene, principalmente, dos tipos de clasificaciones: el Machine Learning (*Aprendizaje Máquina*, ML) y Deep Learning (*Aprendizaje Profundo*, DL), dentro de esta última clasificación podemos encontrar herramientas que han logrado éxito en aplicaciones como la detección de bordes, extracción de características, clasificación de imágenes, reconocimiento de patrones, entre otras.

Las redes neuronales artificiales son la parte fundamental del DL, este tipo de redes buscan emular el funcionamiento de una red neuronal biológica, estas aprenden recibiendo una gran cantidad de datos y mediante cada fase del entrenamiento aprenden patrones los cuales al finalizar el proceso de aprendizaje se vuelven capaces de reproducir la tarea determinada para la cual fue entrenada la red. Al igual que en los filtros clásicos existen diversos tipos de redes neuronales capaces de suavizar el ruido presente en las imágenes, pero estos tipos de ruido requieren de gran cantidad de datos durante la etapa de aprendizaje [6].

El objetivo de este artículo es diseñar y validar el rendimiento de una red neuronal artificial de tipo Autoencoder, este tipo de red neuronal se denomina como no supervisada ya que los datos utilizados durante la fase de entrenamiento y evaluación no requieren de etiquetas [7].

II. METODOLOGÍA

El Autoencoder trabaja con información que se ingresa a la entrada de la red y reduce dimensionalmente dicha información, conocido como el proceso de codificación, hasta un punto en que esa información esta presentada en pequeñas dimensiones y se ubica en un punto que se conoce como espacio latente (Z), una vez la información es almacenada en Z se procede a la reconstrucción de la información, proceso de decodificación, hasta llegar a sus dimensiones originales como se muestra en la Fig. 1 [8]. Al finalizar este proceso se busca que la información obtenida en la capa de salida sea igual a la información introducida en la capa de entrada, como en la ecuación (1).

$$F(x) = G(\tilde{x}) \quad (1)$$

donde, $F(x)$ = función de codificación con respecto a la información de entrada x y $G(\tilde{x})$ = función de decodificación con respecto a la información proveída por el espacio latente (Z).

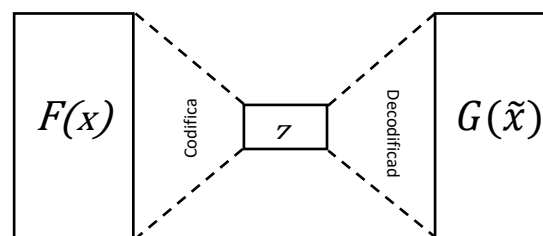


Fig. 1. Arquitectura Autoencoder [6].

La propuesta de este trabajo es la utilización de una variante del autoencoder denominado Denoising Autoencoder (DA) [9], en esta variante el entrenamiento se realiza por medio de imágenes que contengan algún tipo de ruido y que al final del entrenamiento se realice una comparación de la imagen generada por la red con respecto a la imagen original sin ruido. Durante el proceso tanto de codificación como de decodificación el algoritmo aprende patrones los cuales son capaces, para esta tarea en específico, de detectar píxeles corruptos dentro de una imagen y sustituirlos por los valores que crea adecuados que dicha imagen presentaba originalmente antes de ser corrompida por ruido.

En el proceso de aprendizaje del DA, se le suministra toda la información de imágenes corruptas para que, a través de convoluciones contenidas tanto en las capas de codificación como de decodificación, vaya generando nuevos valores calculados por medio de la ecuación (2).

$$net_i = \sum_i W_i x_i + b, \quad (2)$$

donde, net_i = suma ponderada de las entradas con respecto a los pesos, W_i = peso de la conexión de la neurona i , x_i = valor de entrada de la neurona i , b = sesgo.

Estos nuevos valores generados son sometidos a una función de activación denominada Unidad Lineal Rectificada (ReLU) [10] la cual se describe en la ecuación (3).

$$f(net_j) = \begin{cases} net_j & \text{si } net_j > 0 \\ 0 & \text{si } net_j \leq 0 \end{cases}, \quad (3)$$

donde, $f(net_j)$ = salida de la j -ésima neurona.

Obtenidos los valores de salida, estos son comparados con las imágenes originales sin ruido a través del cálculo de la función de pérdida la cual es el Error Cuadrático Medio (MSE) como se muestra en la ecuación (4).

$$MSE = \frac{1}{3Q} \sum_{i=1}^Q \sum_{k \in \{R,G,B\}} (x_i^k - \hat{x}_i^k)^2, \quad (4)$$

donde, MSE = error cuadrático medio, Q = número total de píxeles contenidos en la imagen, x_i^k y \hat{x}_i^k denotan los valores en los canales RGB de los píxeles originales y restaurados respectivamente.

Con los resultados del MSE se procede a realizar la propagación hacia atrás (*Back Propagation*, BP) con la cual por medio del error obtenido se actualizarán todos los pesos de los filtros de convolución contenidos en el autoencoder [11]. En la Fig. 2 se muestra el algoritmo de funcionamiento del DA.

Para la fase de entrenamiento del DA se hizo uso de una base de datos denominada Best Artworks of All Time [12]. La base de datos se corrompió con cuatro diferentes tipos de ruido los cuales fueron: *ruido gaussiano*, *ruido moteado*, *ruido poisson* y *ruido sal y pimienta*, los modelos matemáticos de los tipos de ruido se describen en [13]. Para el ruido gaussiano, moteado y sal y pimienta las imágenes se corrompieron con diferentes densidades de distribución que iban desde 0.1 hasta 0.5, esto con la finalidad de que la red sea capaz de suprimir imágenes que se encuentren corrompidas con una alta densidad de ruido. Toda la información de la nueva base de datos generada se dividió en un 80% para la fase de entrenamiento y un 20% para la fase de pruebas; al no requerir de etiquetas para para el cálculo del error [14] las imágenes reconstruidas por el DA se comparan con las imágenes originales.

Algoritmo: Autoencoder

```

Datos: (x = imagen corrupta)
Alimentar la capa de entrada (capa 1) del autoencoder con x
mientras capaActual no sea capa decodificación
    Calcular neti y salidas de todas las neuronas de la capa actual
    Codificar los datos
    capaActual=capaActual +1
fin mientras
Z=Espacio latente
mientras queden capas por hacer
    Calcular neti y salidas de todas las neuronas de la capa actual
    Decodificar los datos
    capaActual=capaActual +1
fin mientras
Calcular función de pérdida
Actualizar pesos
devolver resultado en la capa de salida
fin
    
```

Fig. 2. Algoritmo del funcionamiento del autoencoder.

III. PRUEBAS Y RESULTADOS

A. Pruebas

Una vez entrenado el DA se genera un modelo, el cual se aplica directamente sobre los datos de prueba previamente definidos. Para cuantificar los resultados, se hizo uso de dos métricas como lo son la Relación Pico Señal a Ruido (*PSNR*) la cual proporciona la relación entre la máxima energía posible de una señal y el ruido que la afecta, una imagen al ser una señal digital puede ser analizada mediante esta métrica, por otra parte, se usa también el Índice de Similitud Estructural (*SSIM*) el cual es una métrica que cuantifica la degradación de la calidad de la imagen provocada por el procesamiento. Estas dos métricas se describen en las ecuaciones (5) y (6), respectivamente [15].

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right), \quad (5)$$

donde, *PSNR* = Relación Pico Señal a Ruido.

$$SSIM_{(x,y)} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (6)$$

$$\sigma_x = \frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}, \quad (7)$$

$$\sigma_{xy} = \frac{(x_i - \bar{x})(y_i - \bar{y})}{N}, \quad (8)$$

donde, *SSIM* = índice de similitud estructural, μ_x y μ_y = denotan el promedio de la imagen original y restaurada respectivamente, σ_x y σ_y = denotan la varianza de la imagen original y restaurada, σ_{xy} = covarianza entre la imagen original y la restaurada, C_1 y C_2 = constantes estabilizadoras.

disminución en sus métricas, sin embargo, cuando se va presentando ruido en la imagen se va obteniendo una buena supresión del ruido. Como se percibe en densidades altas de ruido el autoencoder provee una reconstrucción aceptable en comparación con las otras tres redes analizadas.

En la Fig. 4 se muestra la inspección visual de las imágenes procesadas, la densidad del ruido utilizada para ejemplificar dicha inspección es igual a 0.5, en la comparativa se percibe que el DA reconstruye las tonalidades de los píxeles que se habían visto afectados por el ruido Gaussiano a diferencia de las otras redes que mantienen la tendencia propia de este ruido. En la Fig. 5 se realiza un acercamiento para percibir de mejor manera el procesamiento realizado por cada una de las redes.

La Fig. 6 corresponde a las gráficas obtenidas de los resultados de las métricas utilizadas para las diversas densidades de ruido empleadas.

6



Fig. 4. Imágenes filtradas resultantes por el método analizado y de comparativo aplicados a diversas imágenes con un nivel de ruido gaussiano de 50%.

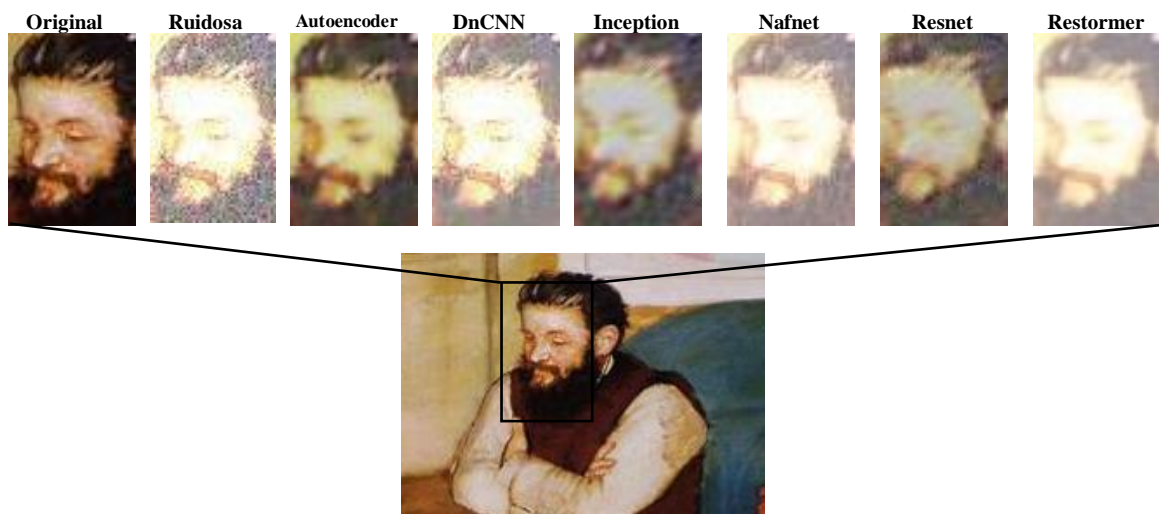


Fig. 5. Acercamiento de la imagen original contra imagen corrupta con ruido Gaussiano e imágenes procesadas mediante diversos métodos.

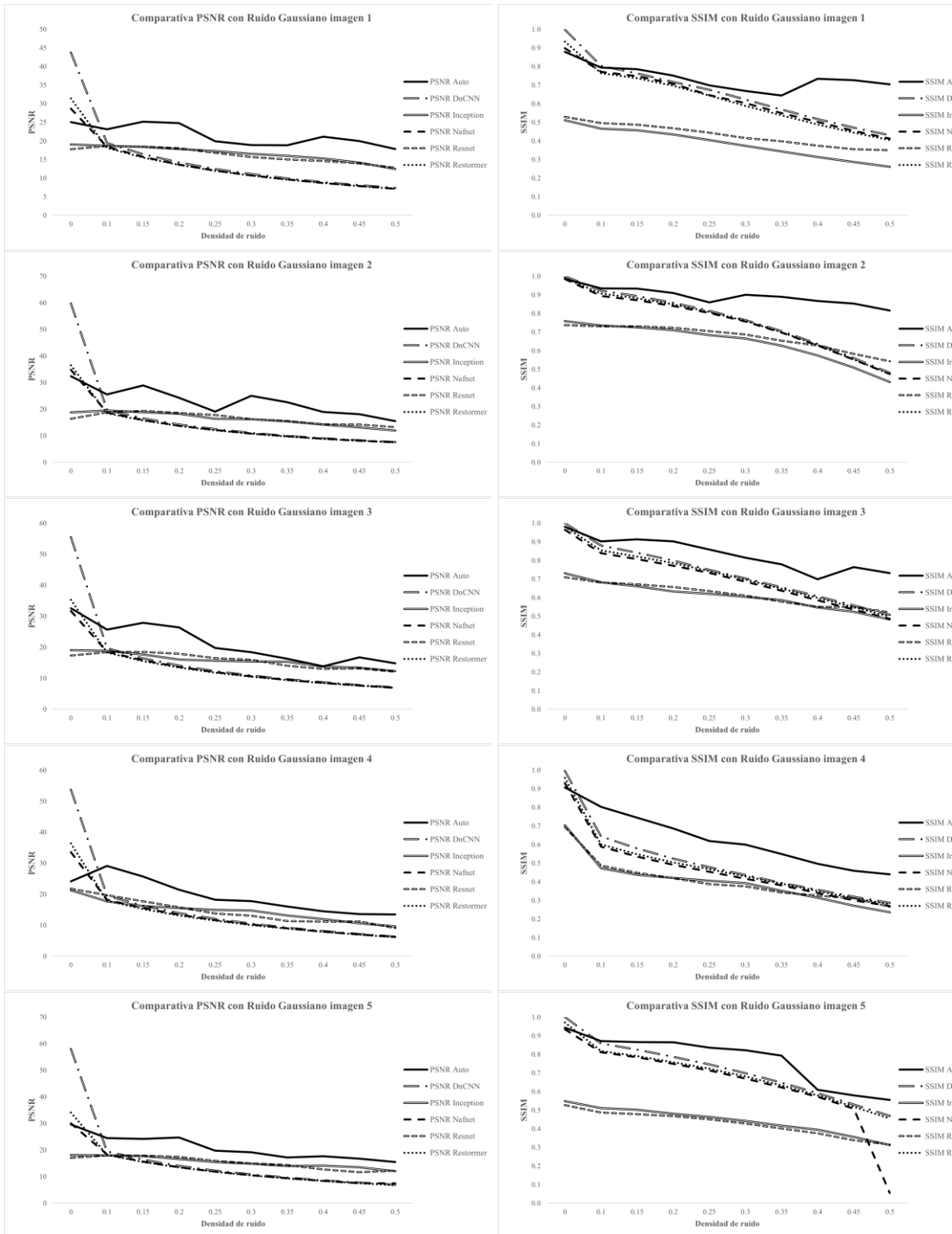


Fig. 6. Valores PSNR y SSIM para las imágenes analizadas con diversas densidades de ruido gaussiano.

9



Fig. 7. Imágenes filtradas resultantes por el método analizado y de comparativo aplicados a diversas imágenes con un nivel de ruido moteado de 50%.

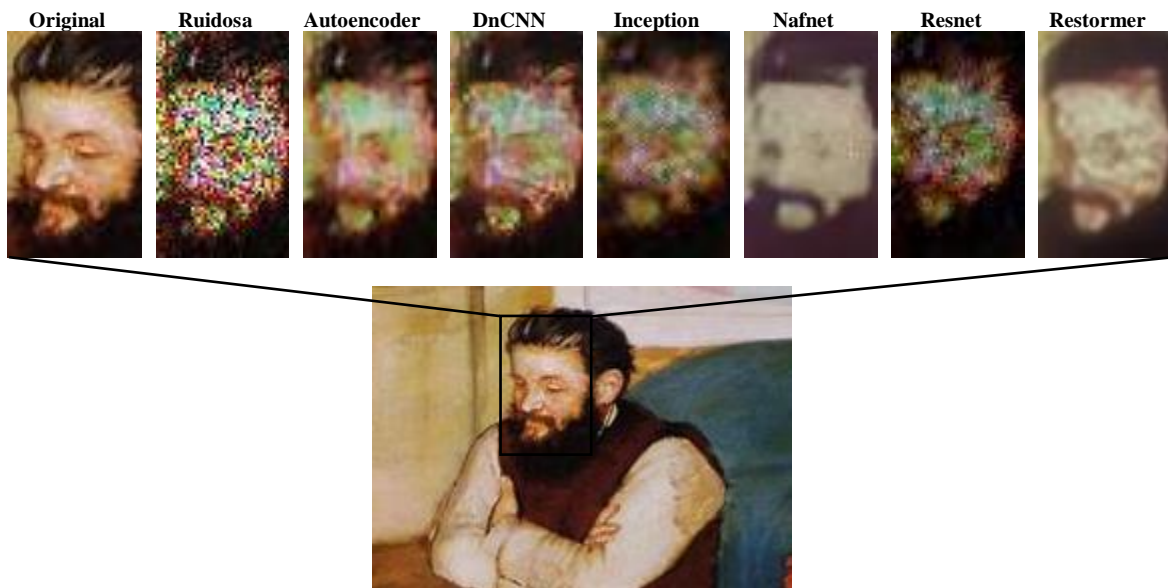


Fig. 8. Acercamiento de la imagen original contra imagen corrupta con ruido moteado e imágenes procesadas mediante diversos métodos.

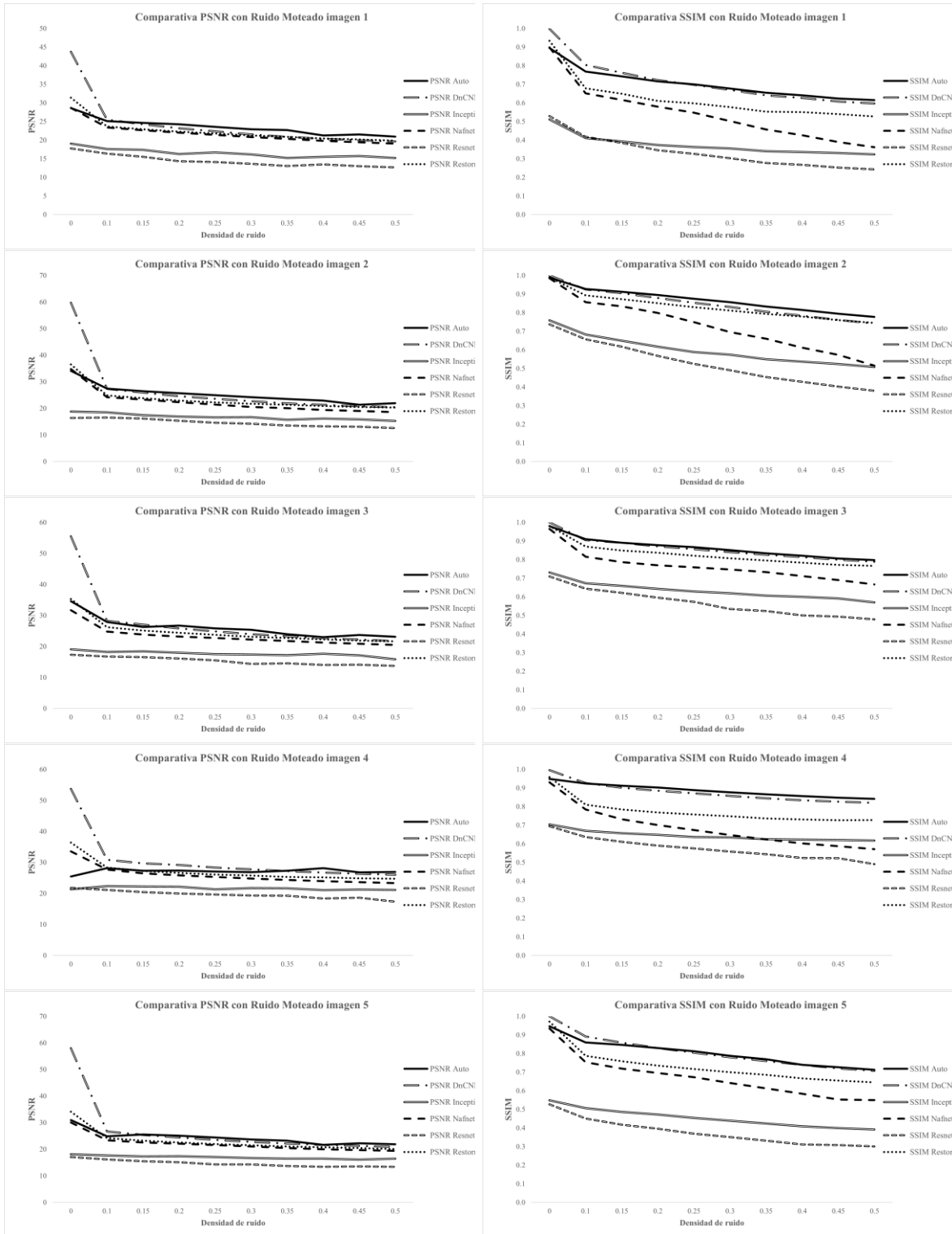


Fig. 9. Valores PSNR y SSIM para las imágenes analizadas con diversas densidades de ruido moteado.

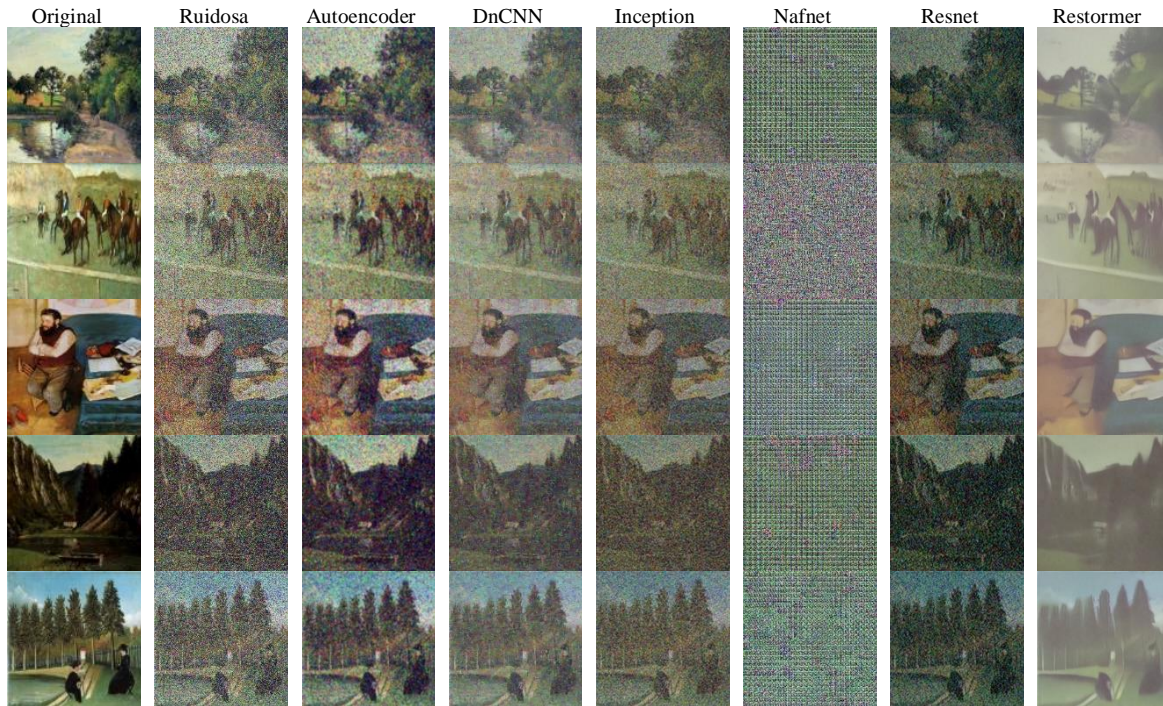


Fig. 10. Imágenes filtradas resultantes por el método analizado y de comparativo aplicados a diversas imágenes con un nivel de ruido sal y pimienta de 50%.

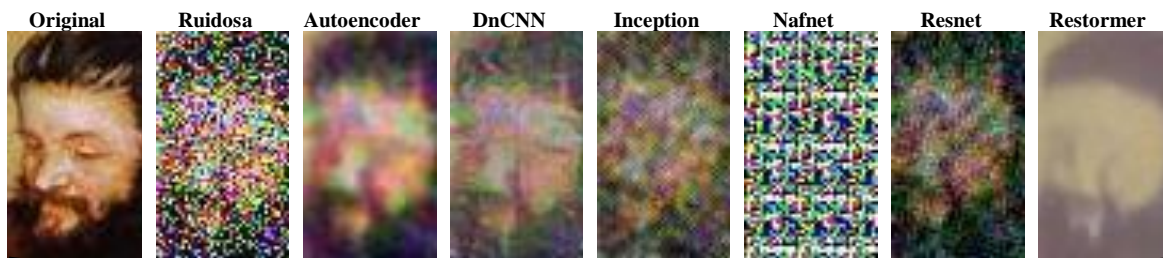


Fig. 11. Acercamiento de la imagen original contra imagen corrupta con ruido sal y pimienta e imágenes procesadas mediante diversos métodos.

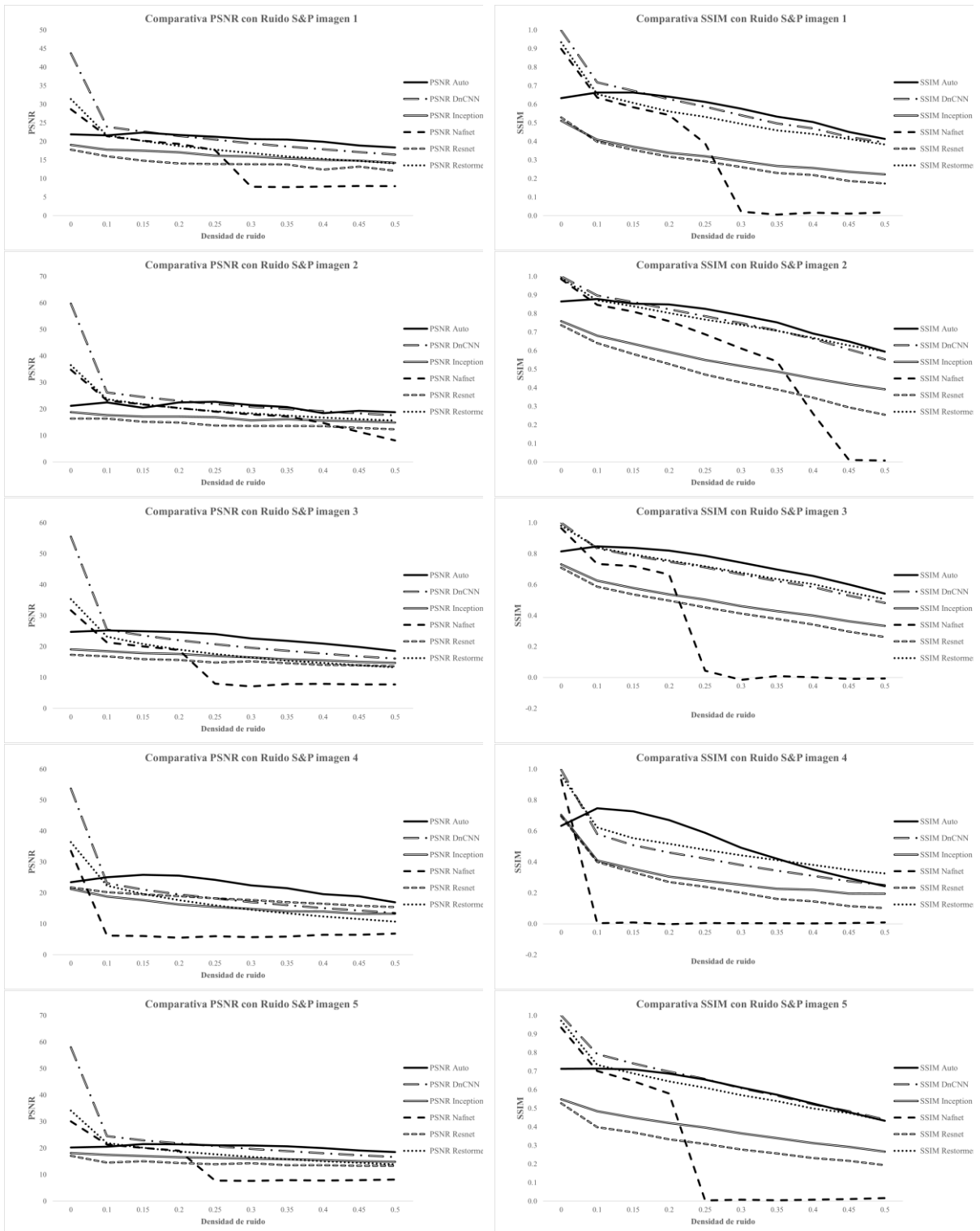


Fig. 12. Valores PSNR y SSIM para las imágenes analizadas con diversas densidades de ruido sal y pimienta.

Para el caso del ruido poisson los resultados se ven reflejados a continuación (véase Tabla 4). En la Fig. 13 se observa que las cuatro redes trabajan de manera efectiva sobre este tipo de ruido sin presentar cambios importantes que afecten el resultado final del procesamiento.

TABLA 4
VALORES PSNR Y SSIM PARA CINCO IMÁGENES CORROMPIDAS POR RUIDO POISSON.

Imagen	PSNR						SSIM					
	Autoencoder	DnCNN	Inception	Nafnet	Resnet	Restormer	Autoencoder	DnCNN	Inception	Nafnet	Resnet	Restormer
1	30.323	32.332	19.012	28.761	17.643	29.865	0.934	0.945	0.501	0.893	0.519	0.904
2	29.853	31.386	18.842	32.048	16.138	33.397	0.976	0.962	0.755	0.972	0.730	0.976
3	33.755	33.059	18.908	30.755	17.334	33.115	0.968	0.957	0.723	0.950	0.704	0.963
4	26.701	36.948	21.542	33.221	21.702	34.815	0.960	0.972	0.701	0.921	0.689	0.945
5	31.017	32.863	18.107	29.537	17.087	31.790	0.966	0.965	0.544	0.927	0.521	0.951



Fig. 13. Imágenes filtradas resultantes por el método analizado y de comparativo aplicados a diversas imágenes con ruido poisson.

En la Fig. 14 mediante el acercamiento se percibe que las redes Nafnet y Restormer suavizan ligeramente el área del rostro continuando la tendencia de perder detalles en la imagen cuando se presenta ruido en ella, en el caso de la red DnCNN se perciben mayor cantidad de píxeles corruptos con respecto al resultado obtenido con la red DA.

En la Fig. 15 se muestran las gráficas comparativas de las cinco imágenes con la misma densidad de ruido poisson presente.

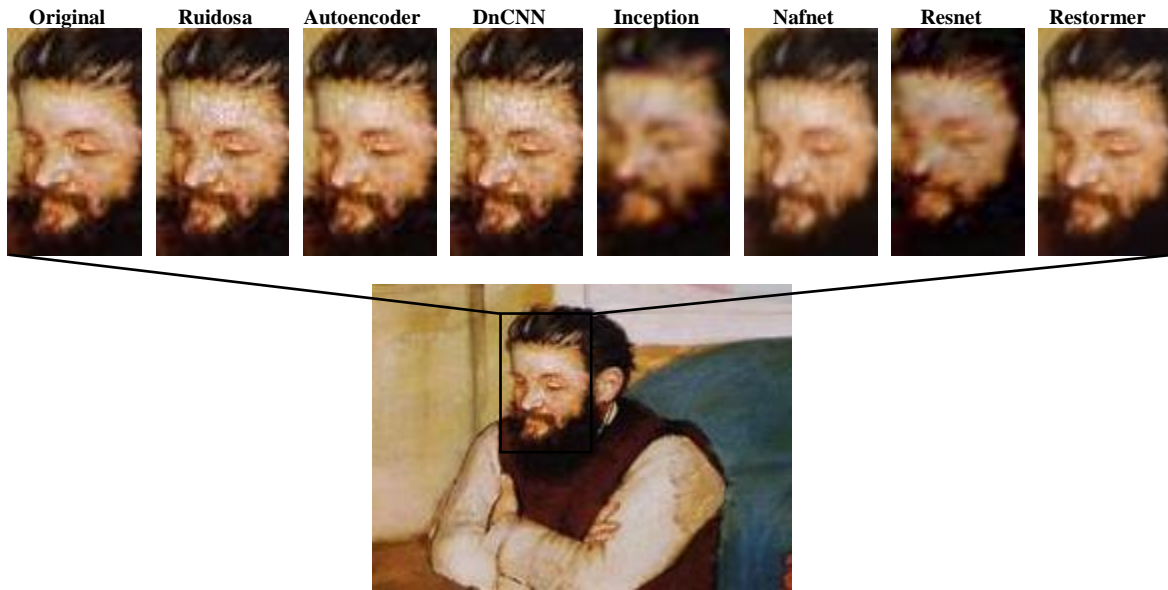


Fig. 14. Acercamiento de la imagen original contra imagen corrupta con ruido poisson e imágenes procesadas mediante diversos métodos.

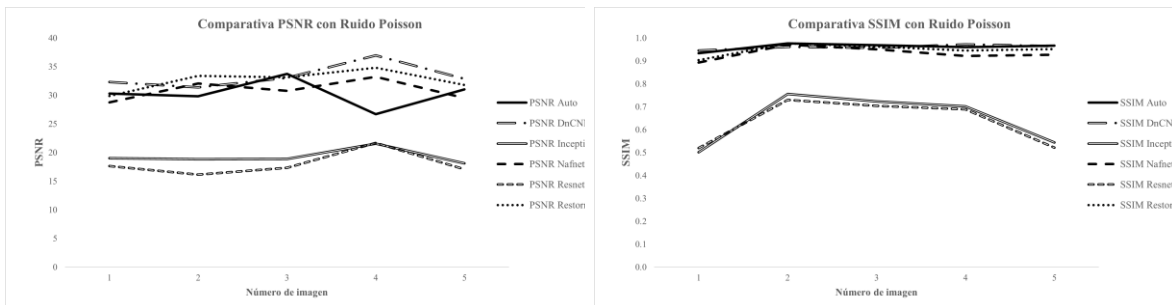


Fig. 15. Valores PSNR y SSIM para las imágenes analizadas con ruido poisson.

IV. CONCLUSIONES

Al llevar a cabo el análisis de los resultados mostrados, se identifica que, a partir del uso del DA, es posible suprimir de forma eficiente el ruido de imágenes a través de las técnicas de aprendizaje profundo independientes del tipo de ruido que corrompa la imagen, lo cual lleva a no tener que determinar un tipo de filtro en particular que pueda considerarse adecuado para situaciones específicas. Los resultados del DA muestran mejores resultados en la supresión de ruido comparado con los algoritmos DnCNN, NAFNET y Restormer para los ruidos gaussiano, moteado y sal y pimienta. En el caso del ruido poisson se observaron resultados similares en las cuatro redes analizadas, sin embargo, en la inspección visual el DA se percibe con menor cantidad de píxeles corruptos preservando los detalles de la imagen en cuestión.

AGRADECIMIENTOS

Los autores agradecen al Instituto Politécnico Nacional y al Consejo Nacional de Ciencia y Tecnología de México por el apoyo en la realización de este trabajo de investigación.

REFERENCIAS

- [1] A. Limshuenchuey, R. Duangsoithong, M. Saejia, "Comparison of Image Denoising using Traditional Filter and Deep Learning Methods", *17th International Conference on Electrical Engineering/Electronics*, 2020.
- [2] V. Pnomaryov, A.J. Rosales-Silva, F. Gallegos-Funes, "Fuzzy directional filter to remove impulsive noise from colour images", *IEICE Trans. Fundamentals*, pp. 570-572, 2010.
- [3] S. Schluter, A. Sheppard, Kendra Brown, *Image processing of multi-phase images obtained via x-ray microtomography: a review*, American Geophysical Union, 2014.
- [4] S. Kaur, "Noise types and various removal techniques", *International Journal of Advanced Research in Electronics and Communication Engineering*, vol. 4, pp. 226-230, 2015.
- [5] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, 4a ed., Pearson, 2018.
- [6] T. Remez, R. Giryes, A. M. Bronstein, "Class-Aware Fully Convolutional Gaussian and Poisson Denoising", *IEEE Transactions on Image Processing*, vol. 27, n.º 11, 2018.
- [7] S. Agarwal, A. Agarwal, M. Deshmukh, *Denoising Images with Varying Noises Using Autoencoders*, Springer Nature Singapore, 2020.
- [8] S. Yu, J. Príncipe, "Understanding autoencoders with information theoretic concepts", *Neural Networks*, vol. 117, pp. 104-123, 2019.
- [9] A. Pawar, "Noise reduction in images using autoencoders", *Proceedings of the Third International Conference on Intelligent Sustainable Systems*, 2020.
- [10] J. Dawani, *Hands-On Mathematics for Deep Learning*, Packt, 2020.
- [11] T. Ye, T. Wang, K. McGuinness, *Learning Multiple Views with Orthogonal Denoising Autoencoders*, Springer International, pp. 313-324, 2016.
- [12] Best Artworks of all Time, *Kaggle: Your Machine Learning and Data Science Community*, <https://www.kaggle.com/datasets/ikarus777/best-artworks-of-all-time> (acc. dec. 2022).
- [13] B. Kumar-Boyatm, "A review paper: Noise models in digital image processing", *Signal and Image Processing: An International Journal (SIPIJ)*, vol. 6, pp. 63-75, 2015.
- [14] A. Patil, A. Pramod, K. Singh, "An Approach to Image Denoising Using Autoencoders and Spatial Filters for Gaussian Noise", *11th International Conference on Cloud Computing, Data Science & Engineering*, pp. 454-458, 2021
- [15] K. Zhang, W. Ren, W. Luo, "Deep Image Deblurring: A Survey", *International Journal of Computer Vision*, 2022.
- [16] K. Zhang, W. Zuo, Y. Chen, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising", *IEEE Transactions on Image Processing*, vol. 26, no. 7, 2017.
- [17] C. Liangyu, C. Xiaojie, Z. Xiangyu, "Simple Baselines for Image Restoration", *Computer Vision and Pattern Recognition*, 2022.
- [18] S. W. Zamir, A. Arora, S. Khan, "Restormer: Efficient Transformer for High-Resolution Image Restoration", *Computer Vision and Pattern Recognition*, 2021.
- [19] C. Wickramasinghe, D. Marino, M. Manic, *ResNet Autoencoders for Unsupervised Feature Learning From High-Dimensional Data: Deep Models Resistant to Performance Degradation*, IEEE, vol. 9, 2021.
- [20] D. Wang, W. Gan, C. Yan, *Inception Model of Convolutional Auto-encoder for Image Denoising*, Springer Nature Switzerland, pp. 174-186, 2020.