

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
ORCID: 0000-0003-4673-0490 | amirandag1100@alumno.ipn.mx

²Instituto Politécnico Nacional, MÉXICO
ORCID: 0000-0001-8436-3025 | arosales@ipn.mx

³Tecnológico Nacional de México, MÉXICO
ORCID: 0000-0001-8665-4096 | dante.mv@cenidet.tecnm.mx

⁴Instituto Politécnico Nacional, MÉXICO
ORCID: 0000-0003-3772-3651 | pescamillaa@ipn.mx

⁵Instituto Politécnico Nacional, MÉXICO
ORCID: 0000-0001-5237-3050 | jkinani@ipn.mx

⁶Instituto Politécnico Nacional, MÉXICO
ORCID: 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).

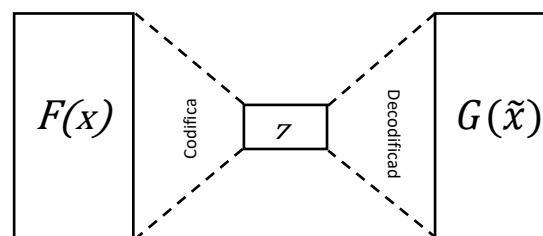


Fig. 1. Arquitectura Autoencoder [6].

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.

B. Resultados

Para la fase de resultados se realizó una comparativa con tres diversas redes neuronales especializadas para la supresión de ruido como lo son: Denoising Convolutional Neural Network (*DnCNN*) [16], Nonlinear Activation Free Network for Image Restoration (*NAFNET*) [17] y Efficient Transformer for High-Resolution Image Restoration (*Restormer*) [18], Resnet [19] e Inception [20]. La comparativa consiste en una inspección visual de los resultados y el uso de las métricas previamente mencionadas.



Fig. 3. Imágenes de prueba.

Para el primer caso se realizó el análisis con cinco diferentes imágenes corrompidas con ruido gaussiano, los resultados de *PSNR* y *SSIM* se muestran a continuación (véase Tabla 1).

TABLA 1
VALORES PSNR Y SSIM PARA CINCO IMÁGENES CORROMPIDAS POR DIFERENTES NIVELES DE RUIDO GAUSSIANO.

Imagen	σ	PSNR						SSIM					
		Autoencoder	DnCNN	Inception	Nafnet	Resnet	Restormer	Autoencoder	DnCNN	Inception	Nafnet	Resnet	Restormer
Camille Pissarro	0	25.060	43.722	19.065	28.682	17.757	31.371	0.879	0.998	0.511	0.898	0.529	0.934
	0.10	23.112	19.433	18.690	18.382	18.642	18.116	0.794	0.804	0.466	0.771	0.496	0.764
	0.15	25.194	16.410	18.403	15.728	18.458	15.534	0.785	0.763	0.457	0.748	0.488	0.737
	0.20	24.785	14.199	17.795	13.642	18.030	13.575	0.751	0.717	0.435	0.707	0.468	0.696
	0.25	19.900	12.480	17.307	12.006	16.813	11.992	0.699	0.675	0.404	0.646	0.445	0.643
	0.30	18.905	11.088	16.534	10.749	15.698	10.710	0.668	0.623	0.374	0.603	0.416	0.589
	0.35	18.814	9.928	16.012	9.660	15.040	9.618	0.644	0.568	0.343	0.550	0.398	0.537
	0.40	21.155	8.935	15.321	8.700	14.607	8.676	0.734	0.518	0.314	0.502	0.375	0.489
	0.45	19.976	8.083	14.169	7.880	13.968	7.868	0.727	0.472	0.286	0.454	0.355	0.444
0.50	17.823	7.349	12.480	7.166	12.766	7.169	0.704	0.431	0.261	0.412	0.351	0.404	
Edgar Degas 1	0	32.364	59.757	18.823	34.779	16.356	36.521	0.986	1.000	0.759	0.985	0.738	0.990
	0.10	25.562	19.751	19.443	18.703	18.793	18.489	0.934	0.922	0.735	0.894	0.731	0.907
	0.15	28.963	16.544	18.882	15.928	19.351	15.735	0.934	0.894	0.726	0.872	0.731	0.882
	0.20	24.301	14.210	18.281	13.785	18.575	13.650	0.910	0.857	0.711	0.840	0.723	0.848
	0.25	19.071	12.444	16.353	12.151	17.822	12.037	0.860	0.815	0.683	0.802	0.705	0.809
	0.30	25.011	11.050	16.270	10.837	16.229	10.749	0.901	0.764	0.666	0.757	0.687	0.760
	0.35	22.635	9.923	15.411	9.858	15.543	9.700	0.889	0.705	0.627	0.697	0.654	0.700
	0.40	18.939	9.006	14.212	8.853	14.319	8.833	0.867	0.637	0.574	0.628	0.628	0.630
	0.45	18.102	8.249	13.193	8.128	14.195	8.119	0.854	0.560	0.509	0.552	0.583	0.553
0.50	15.474	7.625	11.931	7.524	13.224	7.524	0.815	0.481	0.431	0.474	0.543	0.475	
Edgar Degas 2	0	32.599	55.537	19.047	31.673	17.326	35.278	0.980	1.000	0.731	0.965	0.709	0.980
	0.10	25.664	19.682	18.881	18.556	18.451	18.279	0.903	0.880	0.682	0.839	0.681	0.854
	0.15	27.867	16.451	17.641	15.772	18.471	15.540	0.913	0.841	0.661	0.807	0.671	0.821
	0.20	26.407	14.122	16.061	13.662	17.906	13.478	0.902	0.799	0.633	0.771	0.657	0.785
	0.25	19.796	12.315	15.671	11.970	16.520	11.823	0.858	0.749	0.620	0.733	0.635	0.743
	0.30	18.440	10.875	15.424	10.611	15.967	10.505	0.814	0.703	0.604	0.686	0.610	0.697
	0.35	16.309	9.690	15.321	9.473	14.018	9.400	0.780	0.656	0.587	0.638	0.579	0.648
	0.40	13.857	8.678	13.760	8.495	13.016	8.449	0.699	0.608	0.548	0.586	0.551	0.599
	0.45	16.768	7.816	13.409	7.660	13.215	7.621	0.764	0.558	0.524	0.535	0.549	0.548
0.50	14.840	7.054	12.293	6.916	12.139	6.897	0.732	0.507	0.483	0.486	0.521	0.498	
Gustave Courbet	0	24.179	53.679	21.289	33.570	21.755	36.395	0.907	0.996	0.705	0.930	0.695	0.959
	0.10	29.090	19.484	17.690	18.208	19.719	17.902	0.803	0.644	0.473	0.589	0.487	0.601
	0.15	25.717	16.295	16.180	15.499	17.706	15.189	0.744	0.579	0.438	0.535	0.448	0.549
	0.20	21.429	13.922	15.512	13.385	15.808	13.131	0.686	0.524	0.421	0.492	0.420	0.505
	0.25	18.223	12.040	14.921	11.657	13.780	11.472	0.619	0.478	0.405	0.454	0.388	0.467
	0.30	17.756	10.517	14.752	10.227	13.109	10.065	0.600	0.437	0.393	0.416	0.376	0.429
	0.35	16.064	9.240	13.153	9.006	11.368	8.874	0.549	0.394	0.353	0.381	0.342	0.391
	0.40	14.504	8.144	12.000	7.955	11.118	7.843	0.497	0.357	0.314	0.341	0.326	0.351
	0.45	13.620	7.179	10.652	7.022	11.204	6.933	0.459	0.320	0.272	0.301	0.312	0.312
0.50	13.468	6.326	9.625	6.193	9.114	6.130	0.440	0.287	0.236	0.269	0.266	0.280	
Henri Rousseau	0	29.633	57.999	18.127	30.060	17.056	34.139	0.943	1.000	0.548	0.935	0.527	0.971
	0.10	24.489	19.493	17.979	18.324	17.994	18.195	0.871	0.858	0.511	0.812	0.488	0.817
	0.15	24.175	16.370	17.738	15.644	17.925	15.488	0.866	0.825	0.502	0.784	0.479	0.791
	0.20	24.730	14.056	16.594	13.557	17.409	13.427	0.865	0.786	0.481	0.749	0.467	0.758
	0.25	19.729	12.265	15.520	11.906	15.963	11.794	0.836	0.746	0.464	0.713	0.451	0.723
	0.30	19.201	10.819	14.920	10.541	14.947	10.453	0.822	0.698	0.442	0.670	0.428	0.682
	0.35	17.245	9.608	13.996	9.383	14.389	9.315	0.793	0.648	0.417	0.622	0.402	0.633
	0.40	17.671	8.591	14.142	8.410	12.730	8.356	0.609	0.591	0.395	0.572	0.376	0.579
	0.45	16.747	7.714	13.494	7.530	11.661	7.523	0.579	0.530	0.356	0.508	0.338	0.519
0.50	15.521	6.965	12.041	7.494	12.249	6.807	0.555	0.469	0.312	0.055	0.316	0.455	

En negritas se muestran los datos que obtuvieron mejores resultados para cada imagen y a su vez para cada densidad de ruido presente en la imagen, el DA cuando no hay ruido presente en la imagen tiene una ligera

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%.

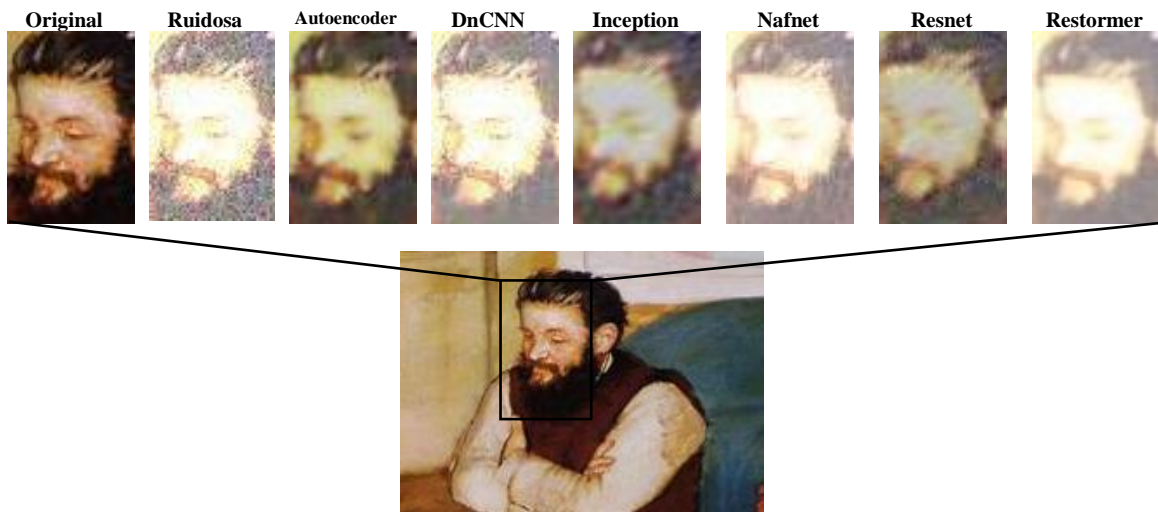


Fig. 5. Acercamiento de la imagen original contra imagen corrupta con ruido Gaussiano e imágenes procesadas mediante diversos métodos.

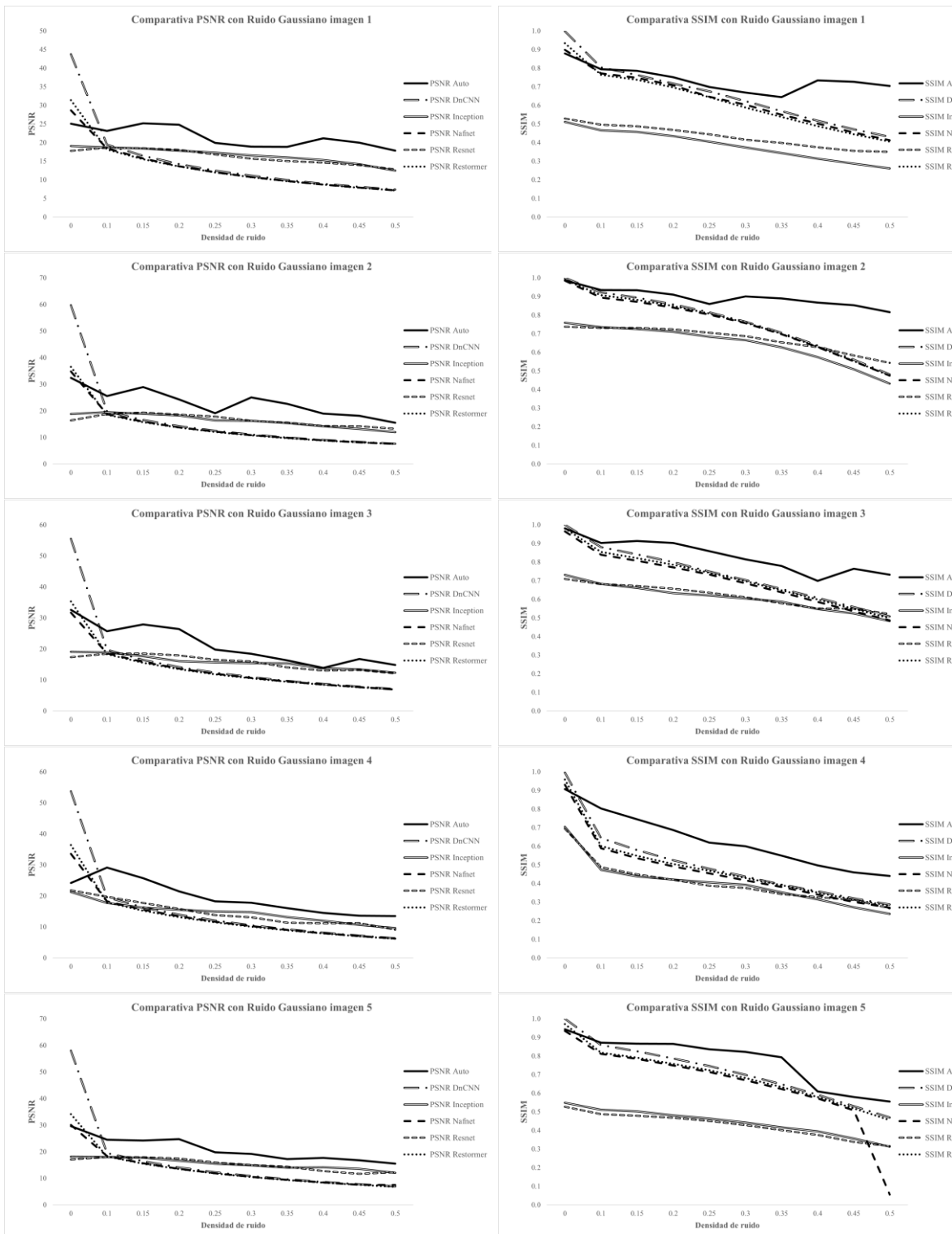


Fig. 6. Valores PSNR y SSIM para las imágenes analizadas con diversas densidades de ruido gaussiano.

Para el ruido moteado se realizó el mismo análisis mencionado anteriormente sobre las cinco imágenes propuestas para ejemplificar los resultados (véase Tabla 2). De igual manera el DA al no haber ruido presente en la imagen presenta una disminución en cuanto a su PSNR y SSIM pero una vez presente el ruido las métricas tienden a favor del algoritmo propuesto.

TABLA 2
VALORES PSNR Y SSIM PARA CINCO IMÁGENES CORROMPIDAS POR DIFERENTES NIVELES DE RUIDO MOTEADO.

Imagen	σ	PSNR						SSIM					
		Autoencoder	DnCNN	Inception	Nafnet	Resnet	Restormer	Autoencoder	DnCNN	Inception	Nafnet	Resnet	Restormer
Camille Pissarro	0	28.602	43.722	19.065	28.682	17.757	31.371	0.895	0.998	0.511	0.898	0.529	0.934
	0.10	25.120	25.472	17.600	23.384	16.367	23.641	0.769	0.803	0.411	0.651	0.417	0.678
	0.15	24.642	24.266	17.414	22.630	15.520	22.926	0.743	0.762	0.392	0.616	0.386	0.649
	0.20	24.270	23.186	16.279	21.978	14.339	22.259	0.716	0.722	0.374	0.580	0.346	0.610
	0.25	23.554	22.358	16.684	21.406	14.141	21.763	0.700	0.697	0.363	0.547	0.326	0.598
	0.30	22.890	21.539	16.170	20.802	13.614	21.227	0.677	0.670	0.355	0.503	0.303	0.578
	0.35	22.681	20.886	15.179	20.323	13.051	20.850	0.654	0.640	0.340	0.457	0.277	0.552
	0.40	21.232	20.325	15.530	19.802	13.460	20.467	0.640	0.627	0.336	0.426	0.267	0.551
	0.45	21.524	19.941	15.747	19.450	13.002	20.190	0.623	0.607	0.330	0.389	0.252	0.540
	0.50	20.959	19.634	15.184	19.060	12.673	19.867	0.615	0.597	0.323	0.362	0.243	0.527
Edgar Degas 1	0	34.095	59.757	18.823	34.779	16.356	36.521	0.989	1.000	0.759	0.985	0.738	0.990
	0.10	27.427	27.349	18.462	24.206	16.529	24.876	0.927	0.925	0.682	0.856	0.657	0.893
	0.15	26.473	25.929	17.419	23.305	16.140	23.840	0.913	0.905	0.650	0.834	0.618	0.873
	0.20	25.682	24.606	16.947	22.337	15.369	22.952	0.895	0.879	0.617	0.798	0.567	0.851
	0.25	24.963	23.631	16.634	21.389	14.589	22.268	0.875	0.853	0.589	0.749	0.525	0.830
	0.30	24.231	22.709	16.653	20.518	14.185	21.744	0.857	0.831	0.574	0.696	0.490	0.812
	0.35	23.496	21.871	15.704	20.054	13.516	21.311	0.833	0.805	0.549	0.660	0.454	0.794
	0.40	22.904	21.326	16.142	19.388	13.249	20.903	0.814	0.783	0.536	0.611	0.428	0.779
	0.45	21.367	20.760	15.826	19.018	13.047	20.506	0.794	0.761	0.523	0.573	0.403	0.759
	0.50	21.938	20.458	15.302	18.592	12.582	20.311	0.777	0.741	0.508	0.515	0.380	0.745
Edgar Degas 2	0	34.565	55.537	19.047	31.673	17.326	35.278	0.980	1.000	0.731	0.965	0.709	0.980
	0.10	27.905	28.188	18.235	24.780	16.731	26.218	0.910	0.904	0.672	0.815	0.644	0.870
	0.15	26.285	27.029	18.445	23.815	16.571	25.120	0.891	0.890	0.659	0.786	0.621	0.849
	0.20	26.717	25.844	17.941	23.188	16.105	24.390	0.878	0.872	0.643	0.769	0.595	0.837
	0.25	25.798	24.850	17.528	22.734	15.509	23.734	0.867	0.857	0.629	0.759	0.573	0.821
	0.30	25.351	23.967	17.421	22.216	14.334	23.099	0.851	0.841	0.620	0.747	0.534	0.807
	0.35	23.984	23.337	17.175	21.770	14.539	22.703	0.834	0.826	0.606	0.733	0.524	0.795
	0.40	23.011	22.693	17.617	21.269	14.016	22.284	0.821	0.813	0.600	0.712	0.500	0.783
	0.45	23.696	22.202	17.163	20.862	14.081	21.911	0.806	0.799	0.591	0.690	0.494	0.772
	0.50	23.130	21.744	15.863	20.513	13.756	21.673	0.798	0.789	0.570	0.667	0.479	0.766
Gustave Courbet	0	25.449	53.679	21.289	33.570	21.755	36.395	0.949	0.996	0.705	0.930	0.695	0.959
	0.10	28.142	30.848	22.378	27.713	21.140	28.361	0.924	0.926	0.670	0.784	0.637	0.810
	0.15	27.282	29.716	22.226	26.469	20.424	27.241	0.912	0.901	0.657	0.732	0.611	0.785
	0.20	27.377	29.195	22.146	25.793	19.941	26.585	0.902	0.886	0.648	0.701	0.591	0.768
	0.25	27.034	28.357	21.309	25.278	19.637	26.100	0.889	0.871	0.636	0.674	0.575	0.758
	0.30	26.868	27.766	21.732	24.793	19.349	25.773	0.877	0.857	0.634	0.647	0.558	0.748
	0.35	27.293	27.172	21.661	24.350	19.228	25.337	0.866	0.844	0.625	0.623	0.544	0.736
	0.40	28.114	26.742	21.053	23.986	18.369	25.178	0.856	0.834	0.622	0.602	0.524	0.731
	0.45	26.812	26.358	21.234	23.639	18.625	24.873	0.847	0.826	0.621	0.587	0.522	0.726
	0.50	26.941	26.007	21.110	23.329	17.368	24.743	0.841	0.820	0.618	0.571	0.490	0.728
Henri Rousseau	0	30.994	57.999	18.127	30.060	17.056	34.139	0.947	1.000	0.548	0.935	0.527	0.971
	0.10	24.887	26.692	17.665	23.336	16.127	24.199	0.859	0.891	0.506	0.753	0.450	0.788
	0.15	25.583	25.399	17.258	22.561	15.463	23.217	0.846	0.858	0.486	0.719	0.418	0.759
	0.20	25.104	24.453	17.386	22.091	15.121	22.535	0.830	0.829	0.472	0.696	0.396	0.735
	0.25	24.455	23.513	17.070	21.553	14.316	21.961	0.813	0.804	0.453	0.673	0.370	0.716
	0.30	23.656	22.794	16.685	21.024	14.290	21.482	0.788	0.780	0.439	0.642	0.351	0.700
	0.35	23.198	22.159	16.488	20.450	13.715	21.044	0.769	0.760	0.424	0.612	0.331	0.685
	0.40	21.660	21.634	16.433	20.013	13.367	20.704	0.740	0.740	0.409	0.584	0.312	0.667
	0.45	22.223	21.250	16.130	19.655	13.532	20.417	0.727	0.720	0.399	0.553	0.308	0.654
	0.50	21.882	20.913	16.490	19.418	13.394	20.094	0.713	0.708	0.392	0.550	0.301	0.646

En la Fig. 7 la densidad utilizada para ejemplificar la inspección es igual a 0.5, en la comparativa se percibe que el DA mantiene los detalles presentes en la imagen sin cambiar las tonalidades originales de los píxeles. En la Fig. 8 se observa que las redes neuronales Nafnet y Restormer suavizan demasiado la imagen provocando que se pierdan los detalles del rostro, la red DnCNN si preserva los detalles al igual que el DA pero contiene mayor cantidad de píxeles aún corruptos.

La Fig. 9 corresponde a las gráficas obtenidas de los resultados para cada una de las densidades de ruido utilizadas sobre las imágenes.

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%.

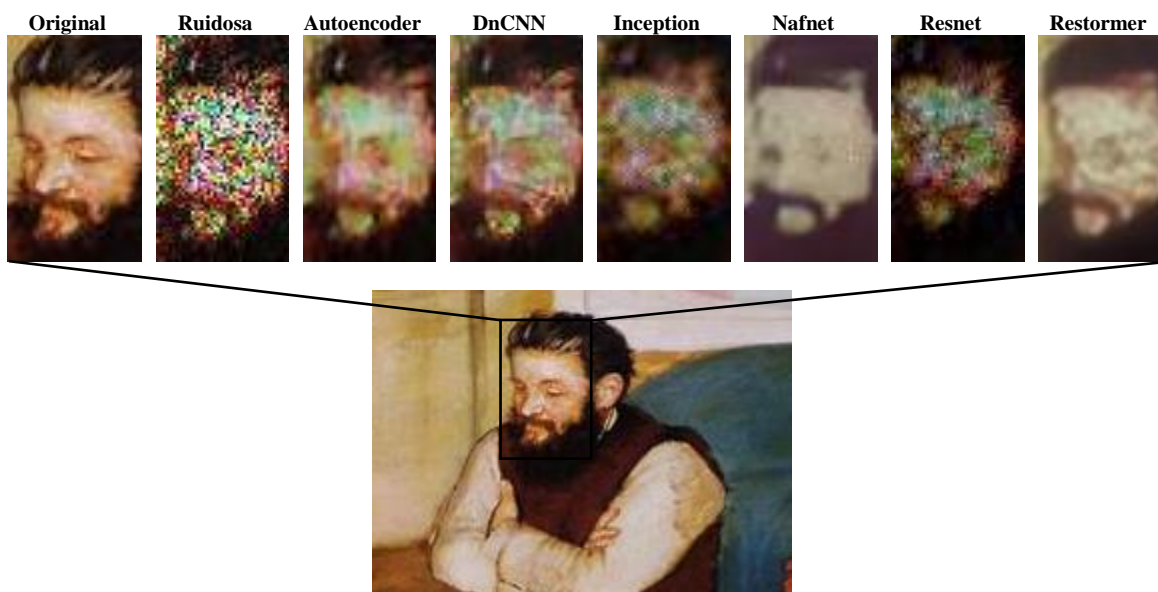


Fig. 8. Acercamiento de la imagen original contra imagen corrupta con ruido moteado e imágenes procesadas mediante diversos métodos.

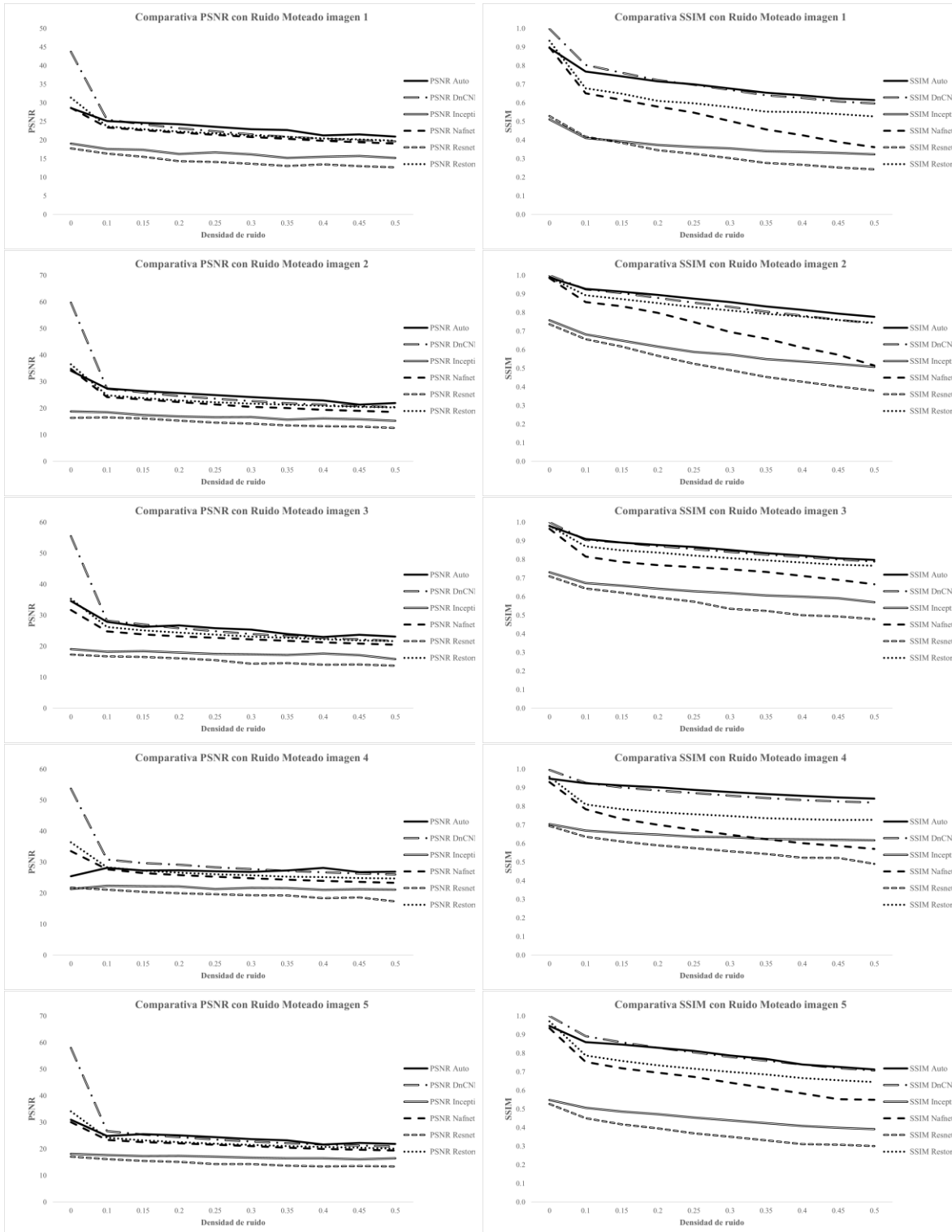


Fig. 9. Valores PSNR y SSIM para las imágenes analizadas con diversas densidades de ruido moteado.

El ruido sal y pimienta es de los ruidos más complejos que existen para ser suprimidos debido a su comportamiento aleatorio de impulsos es por ello por lo que los resultados reflejan esta misma naturaleza del ruido (véase Tabla 3), a pesar de ello el DA obtuvo mejores resultados en densidades altas.

11 TABLA 3 VALORES PSNR Y SSIM PARA CINCO IMÁGENES CORROMPIDAS POR DIFERENTES NIVELES DE RUIDO S&P.

Imagen	σ	PSNR						SSIM					
		Autoencoder	DnCNN	Inception	Nafnet	Resnet	Restormer	Autoencoder	DnCNN	Inception	Nafnet	Resnet	Restormer
Camille Pisarro	0	21.872	43.722	19.065	28.682	17.757	31.371	0.633	0.998	0.511	0.898	0.529	0.934
	0.10	21.608	23.914	17.785	21.407	16.022	21.647	0.663	0.718	0.409	0.636	0.399	0.654
	0.15	22.396	22.663	17.471	20.174	14.795	20.117	0.663	0.673	0.372	0.584	0.355	0.607
	0.20	21.733	21.464	17.064	19.300	14.064	18.768	0.640	0.628	0.338	0.542	0.318	0.561
	0.25	21.245	20.507	16.158	17.642	13.904	17.764	0.613	0.587	0.319	0.390	0.293	0.532
	0.30	20.582	19.493	15.946	7.792	13.865	16.832	0.577	0.541	0.293	0.021	0.262	0.495
	0.35	20.499	18.618	15.392	7.677	13.798	15.955	0.533	0.497	0.267	0.006	0.230	0.459
	0.40	19.934	17.867	15.032	7.857	12.395	15.292	0.504	0.470	0.257	0.017	0.219	0.440
	0.45	18.904	17.095	14.814	8.020	13.204	14.719	0.451	0.424	0.236	0.011	0.187	0.413
	0.50	18.349	16.414	14.245	7.961	12.091	14.096	0.413	0.394	0.223	0.018	0.173	0.384
Edgar Degas 1	0	21.154	59.757	18.823	34.779	16.356	36.521	0.865	1.000	0.759	0.985	0.738	0.990
	0.10	22.486	26.213	17.653	23.214	16.387	23.643	0.878	0.897	0.680	0.847	0.641	0.871
	0.15	20.445	24.471	17.152	21.682	15.176	21.772	0.853	0.861	0.637	0.810	0.581	0.839
	0.20	22.479	23.096	17.158	20.347	14.911	20.317	0.848	0.823	0.593	0.759	0.528	0.802
	0.25	22.680	21.875	16.874	19.032	13.738	19.195	0.825	0.786	0.550	0.688	0.471	0.767
	0.30	21.511	20.850	15.690	17.930	13.630	18.312	0.789	0.748	0.517	0.612	0.429	0.739
	0.35	20.726	19.967	16.148	17.202	13.642	17.543	0.753	0.708	0.487	0.542	0.391	0.707
	0.40	18.451	19.140	15.604	14.743	13.540	16.797	0.692	0.664	0.450	0.261	0.346	0.668
	0.45	19.337	18.342	15.343	11.381	12.861	16.184	0.649	0.606	0.419	0.011	0.294	0.628
	0.50	18.806	17.578	14.875	8.115	12.356	15.671	0.595	0.553	0.392	0.008	0.254	0.596
Edgar Degas 2	0	24.698	55.537	19.047	31.673	17.326	35.278	0.814	1.000	0.731	0.965	0.709	0.980
	0.10	25.190	25.427	18.487	21.328	16.807	23.179	0.847	0.834	0.625	0.734	0.587	0.839
	0.15	24.898	23.565	17.838	20.050	15.912	20.804	0.837	0.788	0.577	0.720	0.536	0.795
	0.20	24.646	22.026	17.555	18.856	15.654	18.980	0.819	0.748	0.536	0.666	0.496	0.755
	0.25	23.990	20.746	16.964	7.984	14.810	17.595	0.786	0.712	0.503	0.044	0.453	0.718
	0.30	22.630	19.603	16.474	7.124	15.219	16.395	0.743	0.667	0.460	-0.015	0.413	0.677
	0.35	21.857	18.587	15.820	7.857	14.603	15.492	0.697	0.623	0.428	0.008	0.378	0.635
	0.40	20.908	17.704	15.509	7.944	14.044	14.687	0.657	0.584	0.400	0.001	0.342	0.604
	0.45	19.821	16.836	14.981	7.747	13.920	13.885	0.600	0.529	0.363	-0.009	0.295	0.550
	0.50	18.578	16.142	14.686	7.752	13.630	13.309	0.542	0.482	0.334	-0.007	0.262	0.507
Gustave Courbet	0	23.461	53.679	21.289	33.570	21.755	36.395	0.634	0.996	0.705	0.930	0.695	0.959
	0.10	25.050	23.150	18.787	6.191	20.243	22.348	0.746	0.581	0.409	0.004	0.400	0.623
	0.15	25.841	21.075	17.630	6.034	19.528	19.617	0.727	0.509	0.355	0.008	0.333	0.553
	0.20	25.584	19.519	16.235	5.526	18.956	17.656	0.671	0.461	0.305	-0.003	0.270	0.518
	0.25	24.205	18.226	15.437	5.996	18.253	16.001	0.588	0.422	0.277	0.005	0.239	0.479
	0.30	22.439	17.028	14.786	5.650	17.709	14.543	0.491	0.379	0.252	0.004	0.201	0.442
	0.35	21.505	16.019	14.018	5.871	17.012	13.405	0.421	0.343	0.226	0.004	0.161	0.411
	0.40	19.579	15.067	13.984	6.465	16.522	12.383	0.351	0.310	0.220	0.003	0.146	0.381
	0.45	18.895	14.271	13.239	6.476	15.862	11.521	0.294	0.275	0.195	0.005	0.114	0.349
	0.50	16.955	13.501	13.286	6.814	15.414	10.695	0.244	0.250	0.194	0.008	0.102	0.326
Henri Rousseau	0	20.246	57.999	18.127	30.060	17.056	34.139	0.713	1.000	0.548	0.935	0.527	0.971
	0.10	20.482	24.518	17.434	21.350	14.499	21.865	0.714	0.792	0.484	0.701	0.398	0.734
	0.15	21.524	22.931	16.964	20.067	15.000	20.114	0.709	0.742	0.450	0.646	0.370	0.688
	0.20	21.454	21.775	16.556	19.035	14.388	18.749	0.687	0.698	0.421	0.580	0.333	0.645
	0.25	21.046	20.735	16.320	7.747	13.891	17.698	0.654	0.658	0.396	0.004	0.308	0.611
	0.30	20.951	19.724	15.988	7.583	14.295	16.696	0.611	0.608	0.365	0.008	0.278	0.570
	0.35	20.693	18.853	15.809	7.924	13.538	15.812	0.573	0.568	0.339	0.004	0.256	0.538
	0.40	20.006	18.025	15.479	7.765	13.535	15.033	0.526	0.522	0.312	0.008	0.232	0.499
	0.45	19.119	17.314	15.066	7.887	13.328	14.418	0.478	0.484	0.292	0.011	0.217	0.474
	0.50	18.486	16.653	14.788	8.124	13.284	13.807	0.433	0.439	0.267	0.016	0.193	0.433

En la Fig. 10 se observa que la red Nafnet tiene problemas para procesar la imagen cuando es una alta densidad, la red Restormer suaviza todos los impulsos generados por el ruido, pero cambia las tonalidades originales de la imagen y a su vez pierde los detalles presentes; la red DnCNN y el DA contienen aún gran cantidad de impulsos, pero mantienen las tonalidades y detalles de la imagen. En la Fig. 11 se muestra un acercamiento para apreciar con mayor detalle las observaciones previamente realizadas.

La Fig. 12 contiene las gráficas obtenidas de los resultados para cada una de las densidades utilizadas con sobre las imágenes.

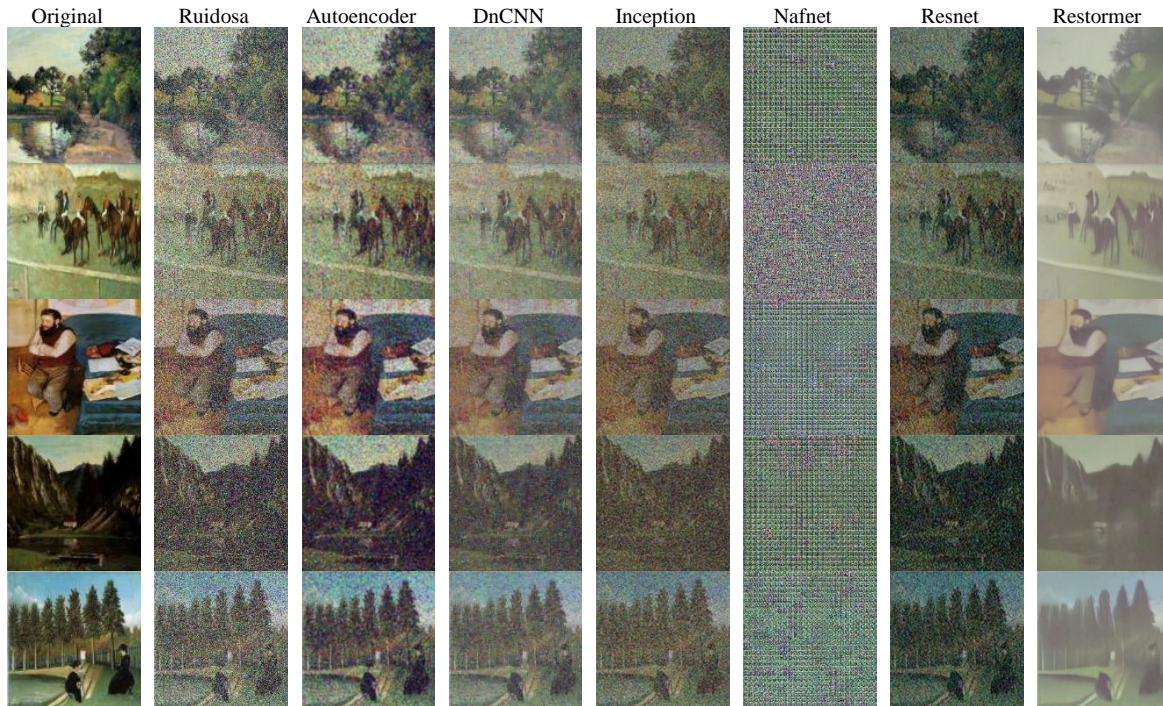


Fig. 10. Imágenes filtradas resultantes por el método analizado y de comparativo aplicados a diversas imágenes con un nivel de ruido sal y pimienta de 50%.

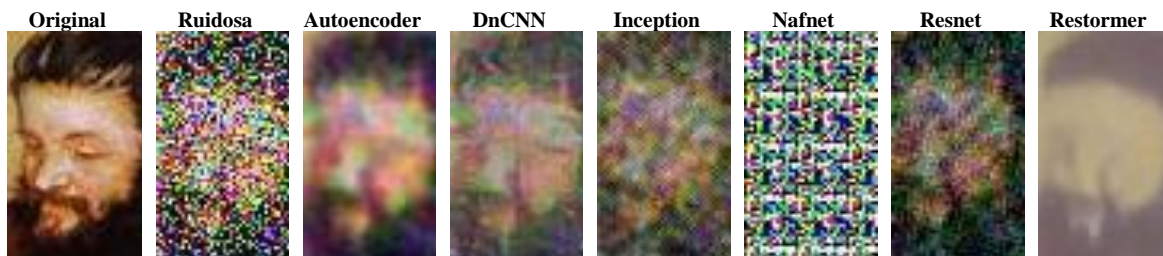


Fig. 11. Acercamiento de la imagen original contra imagen corrupta con ruido sal y pimienta e imágenes procesadas mediante diversos métodos.

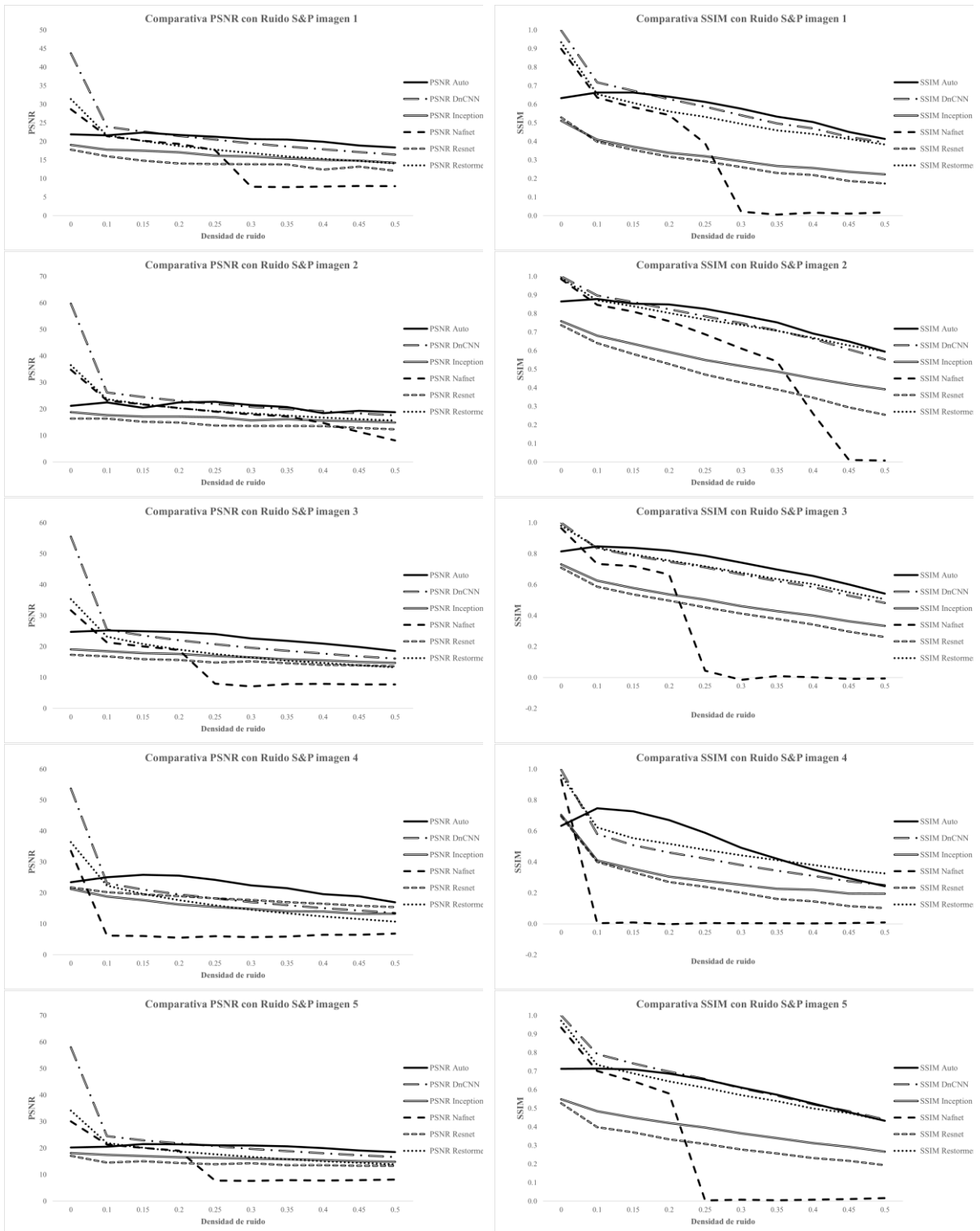


Fig. 12. Valores PSNR y SSIM para las imágenes analizadas con diversas densidades de ruido sal y pimienta.

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.

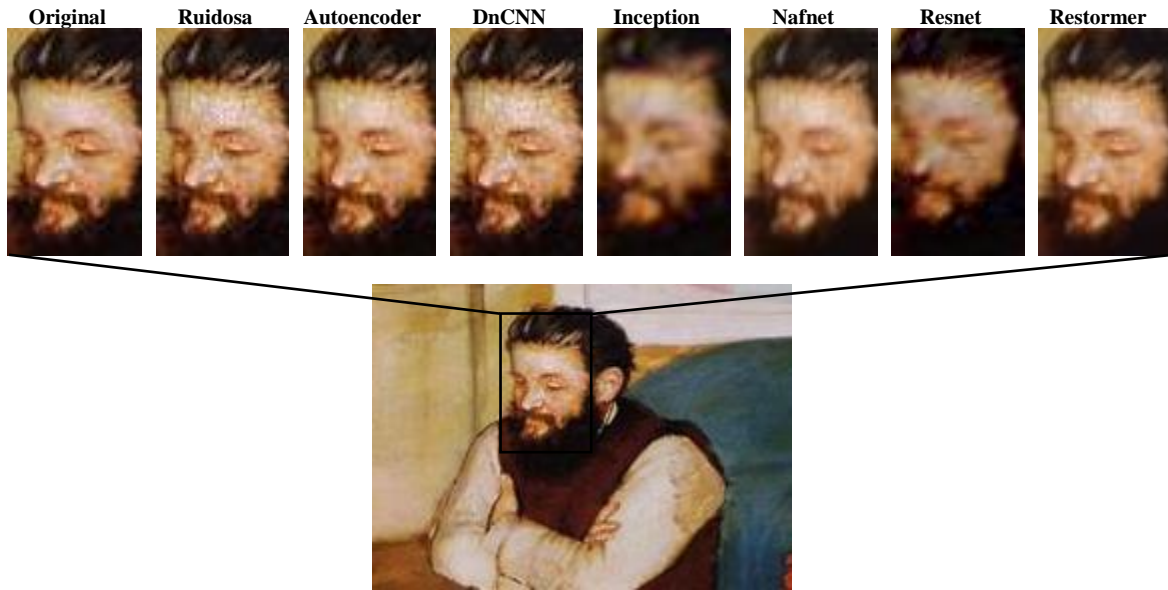


Fig. 14. Acercamiento de la imagen original contra imagen corrupta con ruido poisson e imágenes procesadas mediante diversos métodos.

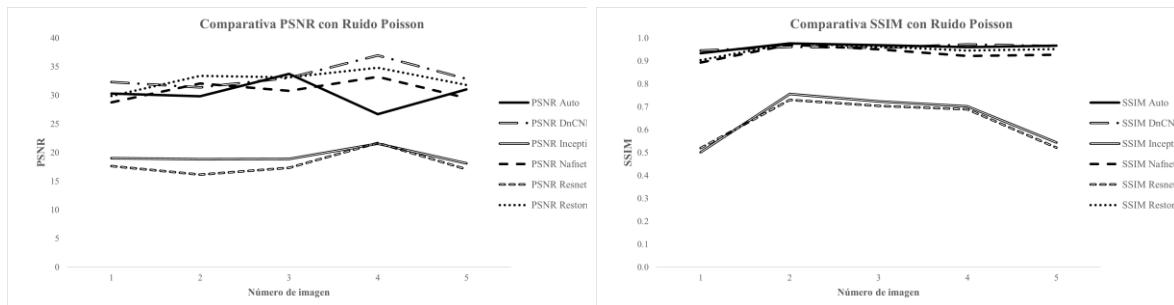


Fig. 15. Valores PSNR y SSIM para las imágenes analizadas con ruido poisson.

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.