



Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y
Tecnología

Máster Universitario en Inteligencia Artificial

**Deep Learning como alternativa a la
deconvolución de imágenes de galaxias
capturadas con el telescopio espacial
Hubble**

Trabajo fin de estudio presentado por:	Ramírez Peláez, Luis Enrique
Tipo de trabajo:	Comparativa de soluciones
Director/a:	Baena Gallé, Roberto Cervantes Rovira, Alejandro
Fecha:	01/02/2023

Resumen

Las anomalías y artefactos de las capturas de imágenes de galaxias reales provocan la presencia de ruido que dificulta la labor de observación e investigación de los astrónomos. El ruido de Poisson y la Función de Dispersión de Punto (PSF) representan dos casos típicos que, de forma habitual, son objeto de tratamiento para su atenuación o supresión. Los avances de la Inteligencia Artificial han permitido la construcción de modelos que pueden ser entrenados para reconstruir imágenes de galaxias mitigando las aberraciones inherentes en el proceso de adquisición. En este proyecto se van a utilizar imágenes reales del telescopio espacial Hubble para entrenar tres arquitecturas de redes neuronales, AEPP2 (basada en Autoencoders), U-Net y DIDN, y realizar una reconstrucción de estas, eliminando el efecto de la Función de Dispersión de Punto, conforme se haría con un algoritmo de deconvolución. Las pruebas realizadas aportan datos con los que se puede afirmar que el modelo de red U-Net es el que mejores resultados ofrece en este contexto.

Palabras clave: Aprendizaje profundo, COSMOS survey, deconvolución, DIDN, función de dispersión de punto, imágenes astronómicas, telescopio espacial Hubble, U-Net.

Abstract

The anomalies and artifacts of the real galaxy image captures cause the presence of noise that hinders the work of observation and investigation of astronomers. Poisson noise and the Point Spread Function (PSF) represent two typical cases that are usually treated for attenuation or suppression. Advances in Artificial Intelligence have enabled the construction of models that can be trained to reconstruct galaxy images, mitigating the aberrations inherent in the acquisition process. In this project, real images from the Hubble Space Telescope will be used to train three neural network architectures, AEPP2 (based on Autoencoders), U-Net and DIDN, and perform a reconstruction of them, eliminating the effect of the Point Spread Function, as would be done with a deconvolution algorithm. The tests carried out provide data with which it can be stated that the U-Net network model is the one that offers the best results in this context.

Keywords: Astronomical imaging, COSMOS survey, deconvolution, Deep Learning, DIDN, Hubble Space Telescope, point spread function, U-Net.

Índice de contenidos

1. Introducción	11
1.1. Motivación	11
1.2. Aberraciones en las imágenes astronómicas.....	12
1.3. Planteamiento del problema	13
1.4. Estructura de la memoria	14
2. Contexto y estado del arte	16
2.1. Deconvolución de imágenes a partir de su PSF	16
2.2. Técnicas de deconvolución	17
2.3. Reconstrucción de imágenes con aprendizaje profundo	18
3. Objetivos y metodología de trabajo.....	26
3.1. Objetivo principal.....	26
3.2. Objetivos secundarios.....	26
3.3. Metodología de trabajo	26
4. Planteamiento de la comparativa	29
4.1. Campo de trabajo	29
4.1.1. El telescopio espacial Hubble y el dataset COSMOS	29
4.1.2. Arquitecturas de redes neuronales	31
4.2. Evaluación de resultados	31
4.2.1. Métricas de evaluación.....	32
4.2.2. Comparativa visual	34
5. Desarrollo de la comparativa	36
5.1. Construcción del dataset	36
5.1.1. Tamaño de las imágenes	38

5.1.2.	Normalización de valores	40
5.1.3.	Procedimiento de deconvolución.....	41
5.2.	Arquitecturas empleadas.....	43
5.2.1.	Arquitectura AEPP2	43
5.2.2.	Arquitectura U-Net.....	45
5.2.3.	Arquitectura DIDN	46
5.3.	Hiperparámetros.....	47
5.4.	Entrenamiento y test	49
5.5.	Recursos hardware y software	49
6.	Discusión y análisis de resultados	51
6.1.	Configuración de hiperparámetros.....	51
6.2.	Evaluación de la arquitectura AEPP2	51
6.3.	Evaluación de la arquitectura U-Net.....	54
6.4.	Evaluación de la arquitectura DIDN	58
6.5.	Discusión	61
7.	Conclusiones y trabajo futuro	64
7.1.	Conclusiones	64
7.2.	Líneas de trabajo futuro	64
	Referencias bibliográficas.....	67
Anexo A.	Estadísticas de entrenamientos	72
Anexo B.	Artículo de investigación	85

Índice de figuras

Figura 1 – Disco de Airy. Fuente: https://upload.wikimedia.org/wikipedia/commons/4/4b/Diffraction_disc_calculated.png ..	13
Figura 2 – Efecto de la convolución de una imagen real (Objeto) y la PSF, obteniéndose la imagen resultante. Fuente: https://svi.nl/DeConvolucion	16
Figura 3 - Imagen original (izquierda) e imagen deconvolucionada (derecha).	17
Figura 4 - Arquitectura GAN para deconvolución de imágenes. Fuente: (Makarkin & Bratashov, 2021).....	20
Figura 5 - Arquitectura Autoencoder con skip-connections. Fuente: (Mao et al., 2016).....	21
Figura 6 - Arquitectura U-Net. Fuente: (Ronneberger et al., 2015).....	21
Figura 7 - Arquitectura SeeingGAN. A la izquierda, red generadora basada en ResNet y U-Net. A la derecha, red discriminadora basada en PatchGan. Fuente: (Gan et al., 2021)	23
Figura 8 - Arquitectura DIDN. Esquema general (a) y subprocesamiento interno (b) - Fuente: (Yu et al., 2019).....	24
Figura 9 - Diseño de los experimentos.	28
Figura 10 - De izquierda a derecha, imagen del cielo explorado por el Hubble https://esahubble.org/images/heic0701g/ y mosaico creado con las capturas individuales terrestres (Koekemoer et al., 2007)	30
Figura 11 - Diagrama del cálculo de SSIM. Fuente: (Wang et al., 2004).....	33
Figura 12 – Resultados de una prueba. Primera fila y de izquierda a derecha, imágenes originales, objetivo y predicha. Fila central y de izquierda a derecha, imágenes originales, objetivo y predicha en representación logarítmica. Fila inferior de izquierda a derecha, diferencias entre la imagen predicha y la imagen objetivo, con mapas de color escala de grises y “gist_stern “	35

Figura 13 - Ejemplo de predicción deficiente. De izquierda a derecha, imagen original, target, predicción, diferencia predicción-target en escala de grises y diferencia predicción-target en mapa de color "gist_stern"	36
Figura 14 - Proceso de creación del dataset de trabajo a partir de las imágenes capturadas y PSF del "COSMOS real galaxy dataset" y del resultado de la deconvolución.	38
Figura 15 - Proceso de recorte y rellenado para adaptar las imágenes a tamaño 128 x 128 píxeles.....	39
Figura 16 - Imagen junto a su histograma original e histograma con valores normalizados [0,1]	40
Figura 17 - Deconvolución con algoritmo Richardson-Lucy con distintos valores de regularización.	41
Figura 18 - Deconvolución con algoritmo unsupervised wiener con valores multiplicados por 1E1, 1E2, 1E3, 1E4, 1E5 y 1E6.....	42
Figura 19 - Arquitectura AEPP2 - (Hernández Afonso, 2022).....	43
Figura 20 – Arquitectura AEPP2. Gráficas de entrenamiento con batch size 64. A la izquierda entrenamiento con 100 epoch, a la derecha con 200 epoch.....	52
Figura 21 - Arquitectura AEPP2. Ejemplos de predicciones. a) reconstrucción óptima. b) Reconstrucción eficiente de la estructura de la galaxia. c) Reconstrucción deficiente de la estructura de la galaxia. d) Efecto cuadrícula. De izquierda a derecha, imagen original, target y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y target, en escala de grises y en mapa de color "gist_stern.	53
Figura 22 - Reconstrucción deficiente de la estructura de la galaxia	53
Figura 23 - Efecto cuadrícula debido a una anomalía en la predicción.	54
Figura 24 - Resultados de las predicciones con la arquitectura U-Net,	55
Figura 25 – Arquitectura U-Net. Gráficas de entrenamiento. A la izquierda ejemplo de convergencia rápida. A la derecha, ocurrencia de overfitting.	56

Figura 26 – Arquitectura U-Net. Ejemplos de predicciones. De izquierda a derecha, imagen original, target y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y target, en escala de grises y en mapa de color "gist_stern".....	57
Figura 27 – Arquitectura DIDN. Gráficas de entrenamiento. A la izquierda convergencia lenta hasta 100 epoch. A la derecha la misma prueba con 200 epoch con la tasa de pérdida estabilizada.....	59
Figura 28 – Arquitectura DIDN. Ejemplos de predicciones. De izquierda a derecha, imagen original, target y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y target, en escala de grises y en mapa de color "gist_stern".....	60
Figura 29 - Efecto cuadrícula debido a una anomalía en la predicción.	60
Figura 30 - Resultados de las predicciones de las tres arquitecturas.	62
Figura 31 - Ejemplos de predicciones realizadas con los mejores modelos de cada arquitectura. De izquierda a derecha, imagen capturada, target, predicción con AEPP2, Unet, DIDN e imagen residual en AEPP2, Unet y DIDN.....	63
Figura 32 – Imágenes de las mismas galaxias capturadas con el telescopio espacial James Webb (arriba) y el telescopio espacial Hubble (abajo) (Casey et al., 2022).....	65
Figura 33 - AEPP. A la izquierda, resultados de entrenamiento (LOSS y MAE). A la derecha resultados de predicción con las imágenes de test (SSIM y PSNR).....	72
Figura 34 - AEPP2. Gráficas de entrenamiento. De arriba a abajo, valores 32, 64, 128 y 256 batch size. Columna izquierda, valores para 100 epoch. Columna derecha, valores para 200 epoch.	73
Figura 35 - AEPP2. Muestra de predicciones. De izquierda a derecha, imagen original, target y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y target, en escala de grises y en mapa de color "gist_stern".	74
Figura 36 - AEPP2. Comparativa de la predicción de 2 imágenes con 4 entrenamientos distintos, de arriba abajo en cada bloque, batch size de 32, 64, 128 y 256.	75
Figura 37 - UNET. Resultados de entrenamiento (LOSS y MAE).	77

Figura 38 – UNET. Resultados de predicciones (SSIM y PSNR)	77
Figura 39 - Gráficas de entrenamiento de los modelos UNET.	78
Figura 40 - UNET. Muestra de predicciones. De izquierda a derecha, imagen original, target y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y target, en escala de grises y en mapa de color "gist_stern.	79
Figura 41 – UNET 64. Comparativa de la predicción de 2 imágenes con 4 entrenamientos distintos, de arriba abajo en cada bloque, batch size de 32, 64, 128 y 256.	80
Figura 42 - DIDN. A la izquierda, resultados de entrenamiento (LOSS y MAE). A la derecha resultados de predicción con las imágenes de test (SSIM y PSNR).....	82
Figura 43 - Gráficas de entrenamiento de los modelos DIDN.....	82
Figura 44 - DIDN. Muestra de predicciones. De izquierda a derecha, imagen original, target y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y target, en escala de grises y en mapa de color "gist_stern.	83
Figura 45 - DIDN16DUB2. Comparativa de la predicción de 2 imágenes con 4 entrenamientos distintos, de arriba abajo en cada bloque, batch size de 32, 64, 128 y 256.	84

Índice de tablas

Tabla 1 - Número de filtros usados en la construcción del modelo AEPP2.	44
Tabla 2 - Número de filtros usados en distintas construcciones de modelos U-Net.....	45
Tabla 3 - Número de filtros y parámetros de arquitectura DIDN con distinto número de bloques de sub procesado DUB.	47
Tabla 4 - Distribución de datos de entrenamiento, validación y test.	49
Tabla 5 - Resultados del modelo con mejores métricas de la arquitectura AEPP2.	52
Tabla 6 - Resultados del modelo con mejores métricas de la arquitectura U-Net.....	55
Tabla 7 - Resultados del modelo con mejores métricas de la arquitectura DIDN.	58
Tabla 8 - AEPP2. Hiperparámetros, métricas de entrenamiento y predicción.	72
Tabla 9 - UNET. Hiperparámetros, métricas de entrenamiento y predicción. Los modelos marcados con (*) se han entrenado en Google Colab.	76
Tabla 10 - DIDN. Hiperparámetros, métricas de entrenamiento y predicción. Los modelos marcados con (*) se han entrenado en Google Colab.	81

1. Introducción

1.1. Motivación

En la adquisición de imágenes astronómicas de galaxias se introducen anomalías provocadas por los instrumentos de observación y por la turbulencia atmosférica. Dichas anomalías, denominadas artefactos afectan a la correcta interpretación de los datos recibidos, provocando errores de apreciación que afectan en mayor o menor medida a los estudios científicos.

Uno de los aspectos que siempre se ha tenido en cuenta por parte de la comunidad científica es el tratamiento de estas anomalías, su atenuación o directamente su corrección. Ese es el motivo por el que históricamente se hayan estudiado y desarrollado distintas técnicas y mecanismos para conseguir, en última instancia, unas imágenes lo más reales y limpias posibles. En este sentido, los avances de la tecnología y en especial la ingeniería informática han posibilitado la aparición de múltiples líneas de investigación para abordar esta problemática.

En el caso de las imágenes astronómicas, los científicos han pretendido mejorar las observaciones situando directamente en el espacio exterior los instrumentos de captación, para evitar el efecto pernicioso de la atmósfera terrestre y tener así una observación mucho más nítida y detallada. Se ha pasado de telescopios en tierra como el Grantecan¹ que usa óptica adaptativa y un espejo primario de 10 metros, a telescopios en órbita con espejos más pequeños y a un coste económico superior, pero que incluyen las ventajas anteriormente comentadas. Existen ya diseños y proyectos de telescopios de 30 metros.

Hasta la reciente puesta en órbita del telescopio James Webb², ha sido sin duda el telescopio Hubble³ una de las herramientas científicas más importante para la comunidad astronómica, y durante tres décadas ha servido para profundizar en el conocimiento del cosmos y en el

¹ Grantecan, Gran Telescopio de Canarias. <http://grantecan.es/>

² Telescopio Espacial James Webb. <https://www.jwst.nasa.gov/>

³ Telescopio Espacial Hubble. <https://hubblesite.org/>

avance en nuevas teorías de diferente índole, como estudios de la formación y evolución de las galaxias, la búsqueda de planetas extrasolares o el conocimiento de la materia oscura.

Pero, a pesar de ser el Hubble un telescopio en órbita, libre de los errores de captación propios de la atmósfera terrestre, sigue siendo sensible a otros artefactos, como el ruido de Poisson y la Función de Dispersión de Punto (PSF), que provocan observaciones erróneas de los objetos estelares a los que se dirigen sus instrumentos. Las galaxias en concreto, situadas en su inmensa mayoría a distancias muy lejanas de la apreciación del ojo humano, hacen que las anomalías en las imágenes adquiridas sean especialmente críticas para una correcta observación e interpretación.

1.2. Aberraciones en las imágenes astronómicas

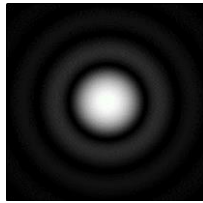
Principalmente existen dos aberraciones que generan ruido en las imágenes capturadas de galaxias: El ruido de Poisson y la función de dispersión de punto (PSF).

El ruido de Poisson (Hasinoff, 2014) es característico de los sensores electrónicos asociados a los instrumentos ópticos empleados en la adquisición de imágenes, provocando fluctuaciones en la captación. Es debido al movimiento aleatorio de los fotones que inciden en el sensor. Este valor de incertidumbre en la señal provoca un cierto nivel de ruido en la imagen final.

La función de dispersión de punto, en inglés PSF (*point spread function*) (Sacek, 2006) define la respuesta de un sensor óptico frente a la luz que incide en forma de impulsos, y es específica de cada instrumento de captación, manifestándose visualmente en un cierto grado de desenfoque. Este comportamiento está motivado por el componente ondulatorio de la luz, que hace que la incidencia de una fuente puntual de emisión, a través de la abertura del telescopio, no sea sobre un único elemento del sistema de captación (píxel) sino que afecta a los píxeles vecinos en lo que se denomina disco de Airy. La Figura 1 muestra los anillos concéntricos provocados por el disco de Airy.

Figura 1 – Disco de Airy. Fuente:

https://upload.wikimedia.org/wikipedia/commons/4/4b/Diffraction_disc_calculated.png



1.3. Planteamiento del problema

La corrección de los errores provocados por el ruido de Poisson y la PSF se ha afrontado desde distintos puntos de vista con diferentes resultados. Los avances de la ingeniería informática, y en concreto, de la Inteligencia Artificial, ha permitido la introducción de nuevas vías de trabajo, que están en continuo desarrollo y de la que se están obteniendo avances significativos.

En este proyecto se va a abordar el uso de redes neuronales para la reconstrucción de imágenes de galaxias, pero, a diferencia de otros trabajos previos que utilizan simuladores de imágenes de galaxias para entrenar a las redes, se usarán imágenes reales captadas por el telescopio Hubble junto a los valores PSF correspondientes.

Como parte imprescindible del trabajo, se ha elaborado un *dataset* lo suficientemente grande y completo para realizar el entrenamiento, formado por imágenes reales, lo que permitirá utilizar las arquitecturas construidas en un entorno real, es decir, reconstruir nuevas imágenes que se capturen en el futuro.

Dicho *dataset* ha sido elaborado a partir del “*COSMOS real galaxy dataset*” (Mandelbaum et al., 2012), formado por 118.992 imágenes de galaxias capturadas por el telescopio Hubble en el programa *COSMOS (The Cosmic Evolution Survey)* (Scoville et al., 2007). El programa COSMOS es el mayor estudio sobre el universo para la investigación de la formación y evolución de las galaxias. Está dirigido por un nutrido grupo internacional de astrónomos que se sirven de las imágenes de las cámaras ACS y WFC del telescopio Hubble capturadas de una porción de cielo durante más de 1000 horas de exposición en 590 órbitas terrestres.

La primera tarea abordada ha sido realizar un proceso de armonización de las imágenes de las galaxias debido a que el *dataset* inicial contiene capturas a distintas resoluciones. Se ha optado por seleccionar un tamaño medio, en concreto 128x128 píxeles, descartando las más pequeñas y recortando las mayores a esa resolución. Las imágenes que se inyecten a la red neuronal serán de dimensiones similares, permitiendo así utilizar las mismas arquitecturas diseñadas sin cambios en la capa de entrada.

Por otra parte, el dataset de partida contiene los valores PSF de cada imagen, lo que permite realizar una deconvolución para reconstruir cómo sería la imagen o (Lucy, 1974; Richardson, 1972) y el filtro *Wiener* (Orieux et al., 2010) el cual utiliza un procedimiento que calcula automáticamente el parámetro de regularización.

Para la realización de las pruebas se han construido tres arquitecturas de redes neuronales:

- A. Arquitectura AEPP2 (Hernández Afonso, 2022) basada en un modelo con *autoencoders*.
- B. Arquitectura U-Net (Ronneberger et al., 2015), utilizando distintos niveles de profundidad.
- C. Arquitectura DIDN (Yu et al., 2019), basada en U-Net al que se añaden subprocesamientos de tipo iterativo.

1.4. Estructura de la memoria

La memoria redactada se ha estructurado en los siguientes capítulos:

- Capítulo 1 – Introducción: Descripción del área de trabajo y el planteamiento del problema.
- Capítulo 2 – Contexto y estado del arte: Se explican las principales vías de investigación para la reconstrucción de imágenes astronómicas usando redes neuronales, centrando el estudio en la restauración galaxias afectadas por la *point spread function*.
- Capítulo 3 – Objetivos y metodología de trabajo: Se exponen en detalles los objetivos perseguidos en este trabajo, así como la planificación y ejecución de todas las tareas identificadas.

- Capítulo 4 – Planteamiento de la comparativa: Se relata el origen del problema a solucionar, así como de las distintas alternativas que se van a llevar a cabo para obtener una solución. Igualmente se explican los criterios de evaluación empleados para validar el éxito de las soluciones propuestas.
- Capítulo 5 – Desarrollo de la comparativa: Se especifica detalladamente el proceso de elaboración del dataset de imágenes de galaxias, así como de las características de las arquitecturas empleadas para el estudio comparativo.
- Capítulo 6 – Discusión y análisis de resultados: Se analizan los resultados obtenidos incidiendo en los problemas encontrados y en las modificaciones realizadas para corregirlos.
- Capítulo 7 – Conclusiones y trabajo futuro: Se ofrecen unas conclusiones de todo el estudio realizado, abordando las posibles mejoras para trabajos futuros.
- Bibliografía: Relación de citas bibliográficas utilizadas en el proceso de investigación y desarrollo.
- Anexo A: Estadísticas de entrenamientos y resultados numéricos y visuales.
- Anexo B: Artículo de investigación: Documento que resume en formato académico el proyecto.

2. Contexto y estado del arte

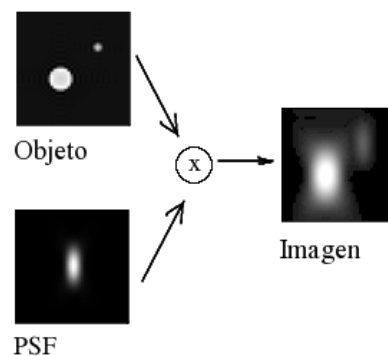
Este proyecto focaliza su campo de actuación en imágenes astronómicas, para lo cual ha sido imprescindible tener acceso a una colección lo suficientemente amplia, rica y representativa de fotografías. Más aún, se ha concretado el uso de imágenes de galaxias a fin de construir modelos de procesamiento que pretenden lograr una mejoría en las características visuales de las mismas, y así facilitar la tarea de los astrónomos y científicos.

En la página web *Nasa/IPAC Infrared Science Archive (IRSA)* ⁴ hay disponibles un considerable listado de conjuntos de imágenes de diferentes características y origen.

2.1. Deconvolución de imágenes a partir de su PSF

Las imágenes reales capturadas con un telescopio espacial son distorsionadas por el efecto de la PSF. Si esta PSF se considera invariable respecto a la posición en el plano del objeto, la imagen final se puede modelar mediante la convolución discreta de la real y la PSF.

Figura 2 – Efecto de la convolución de una imagen real (Objeto) y la PSF, obteniéndose la imagen resultante. Fuente: <https://svi.nl/DeConvolution>



La convolución discreta es un operador matemático que a partir de dos funciones f y g genera una tercera función h que es la superposición de f y la traslación e inversión de g :

$$f * g = h$$

⁴ Nasa/IPAC Infrared Science Archive (IRSA) <https://irsa.ipac.caltech.edu/frontpage/>

En el caso de imágenes astronómicas la convolución opera sobre la imagen real X y la matriz H correspondiente a la PSF, generando la imagen observada Y . Al proceso de convolución se le suma además ruido de tipo Gaussiano, representado como N (Sureau et al., 2020)

$$Y = H * X + N$$

Este operador es la representación matemática de la dispersión de la luz en el elemento de captura.

Por tanto, puesto que las imágenes capturadas contienen las aberraciones producidas por el ruido y la PSF, si lo que se pretende es invertir la dispersión de la luz reubicándola en su lugar original, lo que se hace matemáticamente es la operación inversa, llamada deconvolución (Sparacino et al., 2014) donde la clave es conocer la PSF correspondiente, cuyo cálculo, sin embargo, no es nada trivial.

En el contexto de este proyecto, las PSF de las imágenes capturadas por el telescopio Hubble han sido estimadas con el software *Tiny Tim*.

Figura 3 - Imagen original (izquierda) e imagen deconvolucionada (derecha).

Fuente: (Molina et al., 2001)



2.2. Técnicas de deconvolución

Las técnicas de deconvolución clásicas como las descritas en (Starck et al., 2002) se basan en el post procesamiento de la imagen capturada utilizando la PSF conocida.

- El método *CLEAN* descompone la imagen en un conjunto de funciones buscando el punto con mayor brillo absoluto y restando la PSF. El proceso se repite iterativamente hasta llegar a un límite prefijado. Finalmente, la convolución de las funciones obtenidas y la PSF genera la imagen reconstruida. Este método se emplea

principalmente en imágenes de radioastronomía donde la PSF en el dominio de Fourier está incompleta por el propio proceso de captación de los datos. En su diseño original, CLEAN asume que el objeto está formado solo por fuentes puntuales, por lo que no es válido para la reconstrucción de imágenes de galaxias adquiridas con HST.

- El algoritmo *Richardson-Lucy* es un proceso iterativo basado en la maximización de expectativas, es decir, maximiza la verosimilitud entre la solución propuesta y los datos en presencia de ruido de Poisson. El proceso necesita de una buena estimación de la PSF y requiere de un cierto número de iteraciones para converger de forma satisfactoria, a riesgo de amplificar el ruido existente.
- El filtro *Wiener* opera en el dominio de la frecuencia calculando una estimación estadística de la imagen real emborronada por una cierta PSF, asumiendo una estadística de ruido de tipo gaussiano.
- Los algoritmos anteriores pueden generalizarse mediante una descripción Bayesianas, que se basan en el cálculo de la probabilidad condicionada de tener la imagen real dada la observada.

Otras técnicas y variaciones de las anteriores son descritas, en un contexto Bayesiano, en (Molina et al., 2001) como regularizaciones de máxima entropía, uso de gradientes, algoritmo R-L-EM, algoritmo FMAPE, uso de transformadas multiescalares, etc.

Sin embargo, no siempre es posible tener la imagen PSF o su estimación resulta compleja. En estos casos se puede hacer una deconvolución ciega o *blind deconvolution* (Jefferies et al., 1993) donde, sin conocer la PSF, se hace una estimación de esta basándose en el uso de ciertas suposiciones, lo cual puede no generar resultados óptimos. Frente a esta circunstancia han surgido nuevas técnicas que utilizan el aprendizaje automático profundo para, partiendo de un conjunto de pares *imagen capturada – imagen objetivo*, reconstruir otras imágenes de galaxias sin conocer su PSF.

2.3. Reconstrucción de imágenes con aprendizaje profundo

Las técnicas de aprendizaje profundo han supuesto una nueva perspectiva en la resolución de la problemática de la captura de imágenes astronómicas. El entrenamiento de arquitecturas de redes neuronales con un suficiente número de pares *imagen capturada - imagen objetivo*

(*target*) permite la predicción de cómo será una imagen real dada su captura, sin necesidad de conocer su PSF, ni real ni estimada, lo cual ahorra precisamente la parte más compleja de la reconstrucción de imágenes.

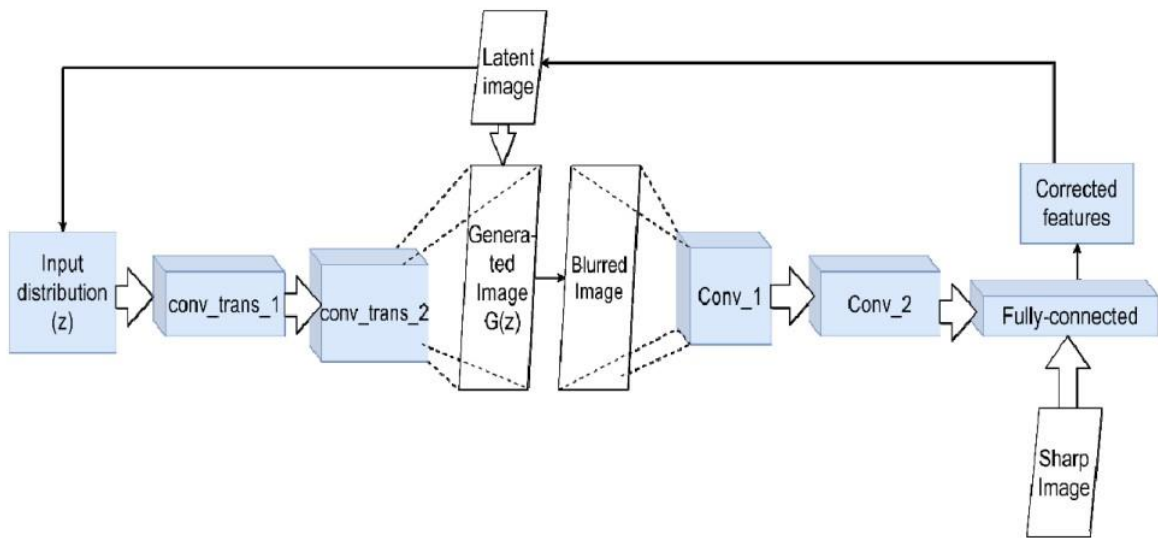
En los últimos años se ha avanzado decididamente en la construcción de arquitecturas de redes de neuronas que sean capaces de realizar el proceso de eliminación de aberraciones, comportándose en la práctica como un proceso de deconvolución ciega.

En (Makarkin & Bratashov, 2021) se hace una descripción de las distintas arquitecturas de redes neuronales que se emplean comúnmente para problemas de deconvolución de imágenes. Una opción son el uso de CNN clásicas (*Convolutional Neural Networks*) (O'Shea & Nash, 2015) que emplean alternativamente capas de convolución con un tamaño determinado de *kernel*, y capas de *max pooling* para obtener características singulares. Aunque inicialmente podrían considerarse como válidas para la obtención de imágenes sin aberraciones, en la práctica se ha comprobado que no son tan exitosas, principalmente por el uso de *kernel* pequeños. Arquitecturas de tipo CNN como *AlexNet* o *VGG* incurren además en resultados pobres cuando el fondo de las imágenes no es uniforme, añadiendo áreas borrosas.

Otro tipo de arquitecturas son las redes GAN (Goodfellow et al., 2020) que implican un planteamiento completamente distinto. Empleadas para deconvolución, la *red generadora* crea una imagen falsa que se parezca lo más posible a la original sin aberraciones, y la *red discriminadora* se entrena para distinguir las imágenes con y sin anomalías. El proceso termina cuando el *generador* es capaz de construir una imagen que el *discriminador* considere libre de aberraciones. La mayor dificultad es aprender la naturaleza del ruido existente. En la propuesta de (Lee et al., 2017) se utiliza una CNN como *red generadora* para la GAN consiguiendo mejorar la predicción.

Otro inconveniente habitual de las redes GAN, y que afecta a este escenario, es su dificultad para el entrenamiento.

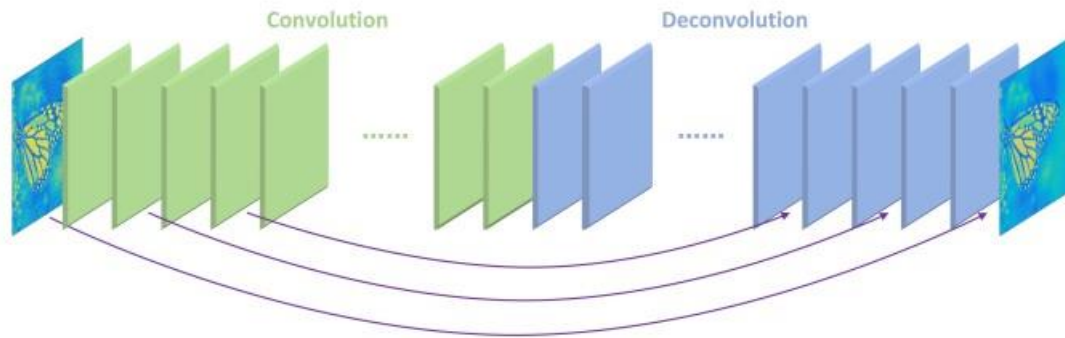
Figura 4 - Arquitectura GAN para deconvolución de imágenes. Fuente: (Makarkin & Bratashov, 2021)



El uso de arquitecturas de *Autoencoders* también permite abordar el proceso de deconvolución de imágenes, donde la primera fase de *encoder* transforma la imagen de entrada a una representación compacta intentando la eliminación del ruido. Posteriormente en la fase de *decoder* se recompone la imagen original sin las características eliminadas. El mayor problema que tienen es que cuanto mayor sea la profundidad de la red, menor será la capacidad de reconstruir la imagen original. La dificultad radica en encontrar un equilibrio entre la profundidad de la red y los resultados en la reconstrucción de la imagen original.

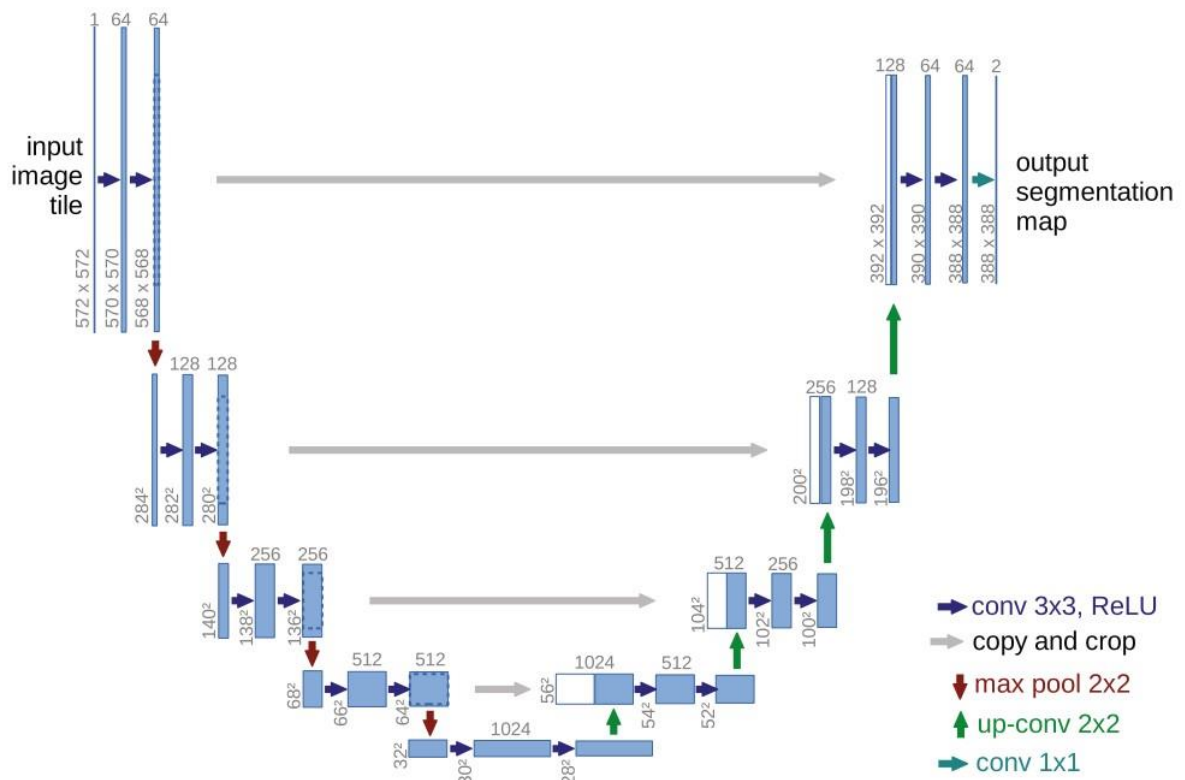
Una modificación al uso de *Autoencoders* que mejora su problemática es propuesta por (Mao et al., 2016). Se trata de una red convolucional muy profunda de tipo *Autoencoders* que consta de una etapa de codificación y otra de decodificación. La particularidad radica en el uso de una característica de las redes *ResNet*, la existencia de *skip-connections*, que conectan capas no adyacentes de ambas etapas, lo que propaga en ambos sentidos la información, evitando la pérdida de características de la imagen de entrada.

Figura 5 - Arquitectura Autoencoder con skip-connections. Fuente: (Mao et al., 2016)



Otras arquitecturas con una filosofía similar a *Autoencoders* son las *U-net*, que están adaptadas a los procesos de deconvolución y tienen su origen en tareas de segmentación de imágenes médicas (Ronneberger et al., 2015).

Figura 6 - Arquitectura U-Net. Fuente: (Ronneberger et al., 2015)



Constan de una primera etapa de codificación en el que se extraen características propias de la imagen, y una segunda etapa de decodificación, simétrica a la primera. En ambas fases hay

una comunicación entre capas simétricas, transfiriéndose información, y evitando que se pierdan características en la codificación. Un aspecto importante es que no necesitan de un dataset de entrenamiento muy grande, y en su lugar se emplea la técnica de *data augmentation* para entrenar de forma eficiente. Esta técnica, sin embargo, supone un importante hándicap para su empleo con datos astronómicos, ya que las modificaciones realizadas con el uso de *data augmentation* pueden modificar la naturaleza o propiedades astrofísicas de las imágenes.

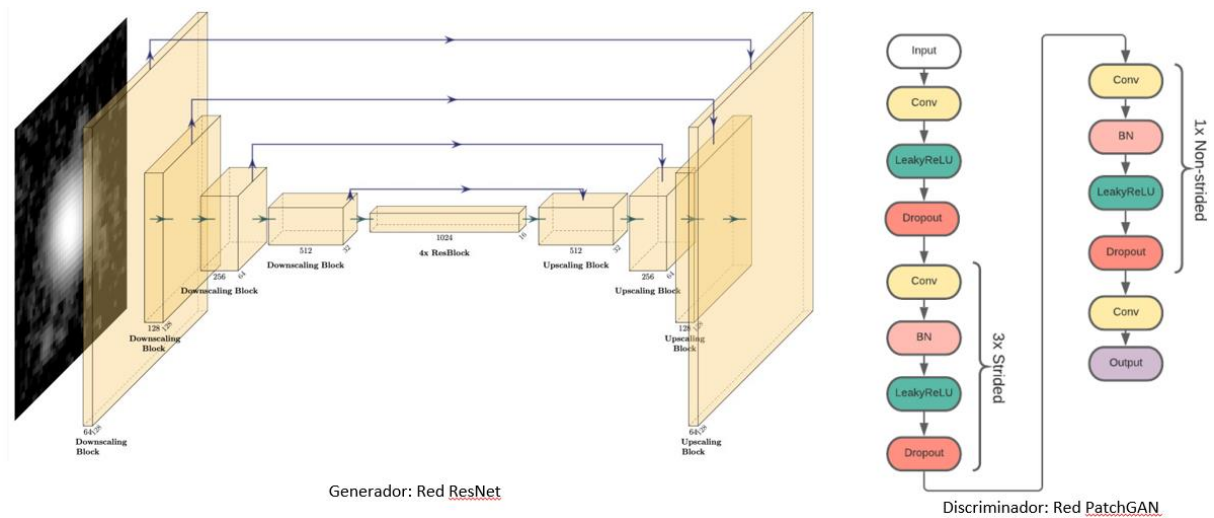
En (Sureau et al., 2020) se utiliza una arquitectura *U-Net* para eliminar aberraciones en imágenes de galaxias simuladas. Partiendo de la imagen simulada y de la PSF, se hace un proceso de deconvolución empleando dos algoritmos, *Tikhonov* y *ADMM*. Se genera así un dataset de entrenamiento con el que la red *U-Net* aprende los parámetros adecuados para, en la fase de predicción, ser capaz de eliminar la aberración de una imagen de prueba. El empleo del algoritmo *Tikhonov* y la red *U-Net* es denominado red *Tikhonet*.

Una evolución y mejora de la arquitectura *Tikhonet* es la propuesta de (Nammour et al., 2022) llamada *ShapeNet* que tiene en cuenta la forma de la estructura de las galaxias, añadiendo a la función de pérdida un nuevo término relacionado con el grado de elipticidad de estas.

En otros trabajos se construyen arquitecturas híbridas como *SeeingGAN* (Gan et al., 2021) donde se combinan *ResNet* y *U-Net* implementando además algunos cambios adicionales. Por otra parte, se modifica la función de pérdida usando una combinación de pérdida *Wasserstein* (Arjovsky et al., 2017) y pérdida perceptual (pérdida sobre los mapas de características de las capas convolucionales). También se añade como métrica de evaluación la CW-SSIM que es más robusta a pequeños cambios en la geometría de las imágenes de entrada y salida.

Figura 7 - Arquitectura SeeingGAN. A la izquierda, red generadora basada en ResNet y U-Net.

A la derecha, red discriminadora basada en PatchGAN. Fuente: (Gan et al., 2021)

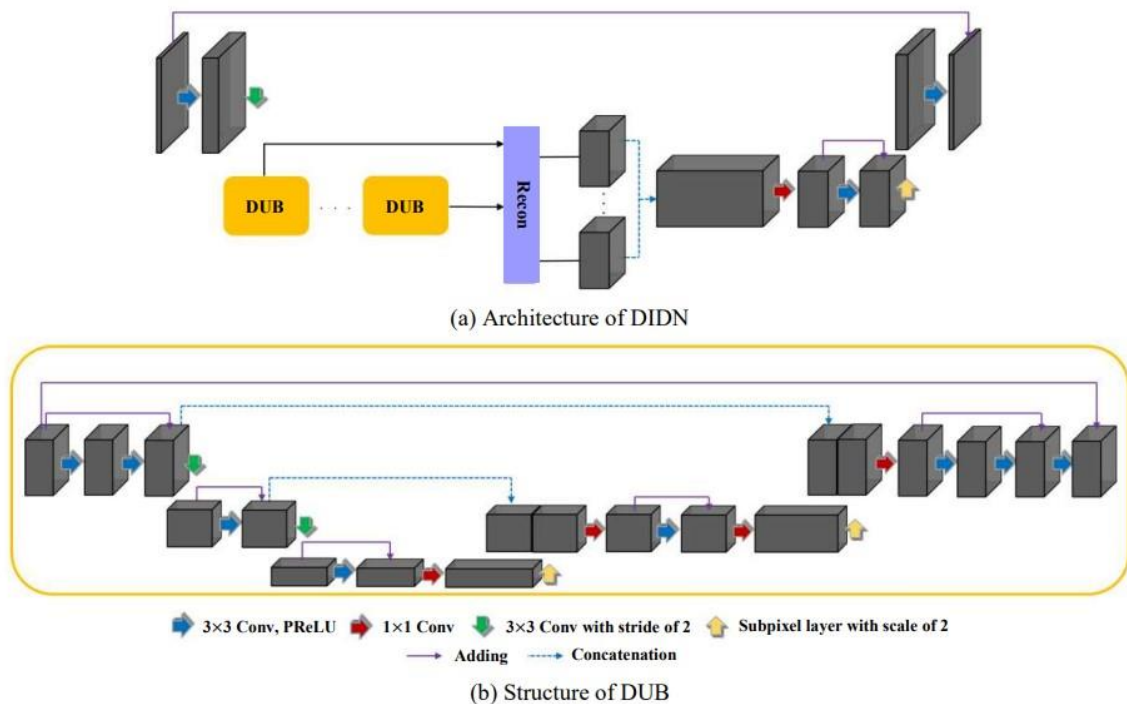


Las arquitecturas *U-Net* se han posicionado como muy eficaces en situaciones donde hay escasez de datos, consiguiendo buenos resultados en dataset reducidos. Esta cualidad es también explorada con la arquitectura *Learnlet* (Ramzi et al., 2023) que se basa en el uso de la funcionalidad del descenso de gradiente para mejorar el poder expresivo de las *wavelets*. En este *paper* se compara con *U-Nets* tradicionales con distinto grado de profundidad. Estas arquitecturas son también objeto de estudio para la deconvolución de imágenes de galaxias, como el trabajo realizado por (Akhaury et al., 2022) en el que se contrastan los resultados con *Tikhonet*, *Lernet* y *U-Net64*, y se introducen modificaciones para evaluar las métricas, basadas en el uso de una ventana en el espacio de la imagen donde se realizan los cálculos de error. Los autores concluyen que los dos últimos modelos y especialmente *U-Net*, obtienen resultados superiores al ser capaces de generalizar mejor que *Tikhonet*.

La gran cantidad de trabajos llevados a cabo con distintas arquitecturas han hecho de la búsqueda de mejoras en las operaciones de deconvolución y eliminación de ruido en imágenes un campo de estudio con gran cantidad de investigadores dedicados. Arquitecturas como *DIDN* (Yu et al., 2019) amplían la complejidad de los modelos con unos resultados prometedores. En este caso, la red consta de 4 etapas: extracción de características, reducción y escalado iterativo (subprocesamiento DUB), reconstrucción y mejora. El proceso iterativo

realizado con la estructura DUB (Figura 8.b) es en esencia una red U-Net, y su iteración consigue mejorar la eliminación de ruido en las imágenes. Con esta arquitectura se consiguió el segundo y tercer puesto en los desafíos *Image Denoising Challenge track 1 y track 2* respectivamente, en el NTIRE 2019 ⁵.

Figura 8 - Arquitectura DIDN. Esquema general (a) y subprocesamiento interno (b) - Fuente: (Yu et al., 2019)



El estado del arte respecto a las técnicas de deconvolución de imágenes astronómicas es un área en continua evolución en la que se hacen estudios y contribuciones tanto con metodología clásicas, basadas en algoritmos y transformaciones matemáticas, como con el uso de aprendizaje automático, siendo el *Deep Learning* su máximo exponente. Desde 2019 las contribuciones a la cosmología de estas técnicas no han hecho sino crecer significativamente (Huertas-Company & Lanusse, 2023).

Las arquitecturas basadas en *U-Net*, tanto en su configuración original como con modificaciones que intentan mejorar sus puntos flacos o simplemente pretenden

⁵ NTIRE 2019. New Trends in Image Restoration and Enhancement workshop and challenges on image and video restoration and enhancement: <https://data.vision.ee.ethz.ch/cvl/ntire19/>

Deep Learning como alternativa a la deconvolución de imágenes de galaxias capturadas con el telescopio espacial Hubble
experimental para mejorar los resultados, están demostrando su valía ofreciendo unos resultados muy destacables, contribuyendo significativamente al estudio del cosmos y su evolución en el tiempo.

3. Objetivos y metodología de trabajo

3.1. Objetivo principal

Mejora de la resolución espacial de imágenes de galaxias capturadas con el telescopio Hubble mediante la corrección de la función de dispersión de punto usando redes convolucionales.

3.2. Objetivos secundarios

En la consecución del objetivo principal de este proyecto es necesario el cumplimiento de una serie de hitos alcanzables que son determinantes para la obtención de resultados.

1. Construir un dataset de imágenes homogéneo y suficientemente extenso para el proceso de aprendizaje, validación y test.
2. Conseguir un mecanismo eficiente para el tratamiento y almacenamiento de las imágenes, incluidas las operaciones de ajuste de tamaño y normalización de valores sin pérdida de información.
3. Identificar y estudiar las técnicas de deconvolución existentes y sus parametrizaciones para aplicar a las imágenes capturadas y las PSF asociadas, y así obtener la imagen objetivo o *target* con el que entrenar.
4. Estudiar las arquitecturas de redes neuronales susceptibles de ser empleadas con éxito en el contexto de trabajo, y elección de las que se emplearán para el proyecto.
5. Elegir la mejor parametrización de las redes empleadas, o un rango con el que ofrecer resultados comparables.
6. Determinar las métricas de error que mejor se adapten a este tipo de estudios.
7. Identificar la red neuronal más apropiada para este ámbito.

3.3. Metodología de trabajo

Las características de este proyecto, su ámbito, los datos a analizar y los métodos con los que experimentar han permitido su desarrollo mediante una metodología empírica en la que, a partir de observaciones, se han realizado experimentos para demostrar una hipótesis, que es el objetivo principal del proyecto.

En concreto se ha seguido una metodología empírica cuantitativa caracterizada por los siguientes elementos:

- Planteamiento del problema: Se parte de un contexto bien definido del que se dispone de información y datos suficientemente representativos, debidamente estructurados y con posibilidad de operación sobre ellos. Con esta premisa es posible plantear hipótesis que se pueden demostrar con experimentos, utilizando herramientas y mecanismos usados en proyectos similares en el estado del arte.
- Formulación de hipótesis: Se constata la existencia de un problema sobre el que se plantea un modelo para llegar a una solución factible, utilizando experimentos limitados en el tiempo, reproducibles y medibles.
- Análisis de datos: La ejecución de los experimentos genera resultados que pueden ser evaluados con distintas métricas, definidas acorde a la naturaleza de los datos.
- Conclusiones: Los resultados obtenidos pueden servir para determinar conclusiones basadas en un análisis objetivo de dichos resultados.

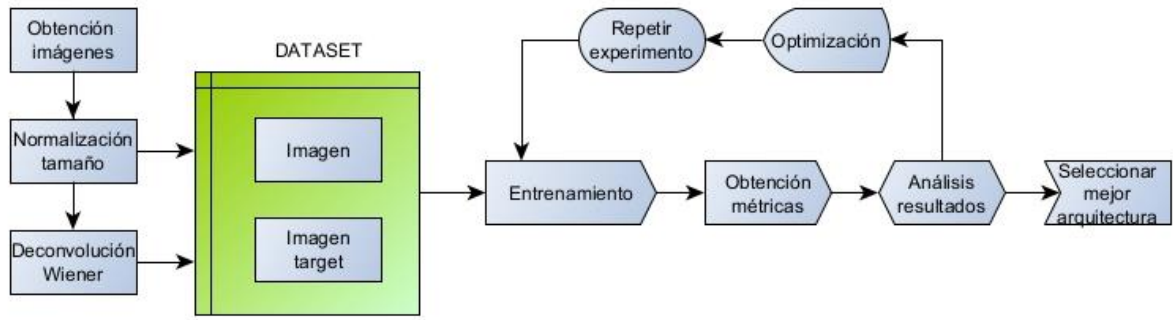
Los experimentos se han diseñado atendiendo a los siguientes requisitos:

- Definir los datos, las variables y las métricas de evaluación.
- Determinar la relación entre variables.
- Garantizar la repetibilidad de los experimentos.
- Utilizar datos de test para evaluar el modelo.

Particularizando para este ámbito, los experimentos han incluido los siguientes aspectos:

- Obtención de imágenes iniciales.
- Normalización de tamaño.
- Deconvolución de imágenes para obtener el *target*.
- Entrenamiento.
- Optimización de parámetros.
- Análisis de resultados.
- Selección de la mejor arquitectura.

Figura 9 - Diseño de los experimentos.



4. Planteamiento de la comparativa

Atendiendo a la definición de los objetivos principal y secundarios, se ha propuesto una comparativa de arquitectura válidas para la reconstrucción de imágenes de galaxias. Se ha buscado un conjunto de capturas de las que se dispone su PSF, con la que se realizará una deconvolución, que será la imagen objetivo a predecir por la red neuronal. En términos prácticos la red neuronal realizará un proceso similar a una deconvolución ciega.

Mediante la evaluación de métricas apropiadas, se determinará qué arquitectura es capaz de reconstruir con mayor eficacia las imágenes reales de las galaxias.

4.1. Campo de trabajo

La realización de las pruebas requiere de dos elementos fundamentales.

1. Un dataset de imágenes astronómicas.
2. Una arquitectura de red neuronal.

Existen distintos *dataset* de imágenes astronómicas usados por la comunidad científica, obtenidos a partir de campañas de observación con distintos telescopios espaciales. En la página web *Nasa/IPAC Infrared Science Archive (IRSA)* ⁶ hay disponibles un considerable listado de conjuntos de imágenes de diferentes características y origen.

Respecto a las arquitecturas de redes neuronales, el requisito previo es que estén adaptadas a operar con las imágenes de partida, o en su defecto puedan ser adaptadas y parametrizadas para su uso libre de errores.

4.1.1. El telescopio espacial Hubble y el dataset COSMOS

Para el desarrollo de presente trabajo se ha empleado el dataset “*COSMOS real galaxy dataset*” (Mandelbaum et al., 2012) formado por 118.992 imágenes de galaxias capturadas con el telescopio Hubble como parte del programa COSMOS, *The Cosmic Evolution Survey* (Scoville et al., 2007)

⁶ <https://irsa.ipac.caltech.edu/frontpage/>

El programa COSMOS es el mayor estudio sobre el universo en el ámbito de la investigación de la formación y evolución de las galaxias. Está dirigido por un nutrido grupo internacional de astrónomos que se sirven de las imágenes de la cámara ACS - *Advanced Camera for Surveys* (Ryon et al., 2022) del telescopio Hubble, capturadas de una porción de cielo durante más de 1.000 horas de exposición en 583 órbitas terrestres (Koekemoer et al., 2007).

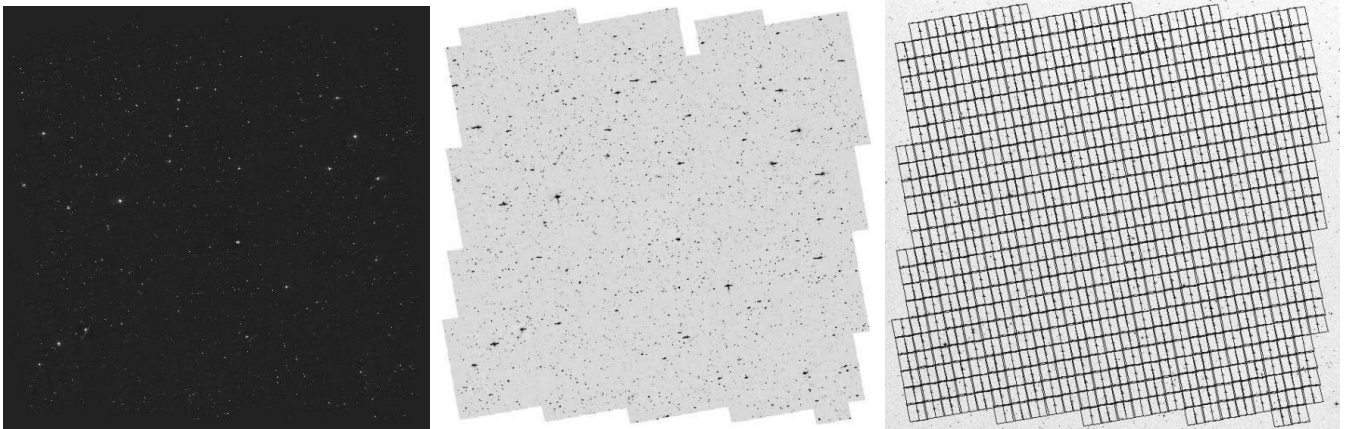
En la captura con la cámara ACS se ha utilizado uno de sus 3 canales, el WFC (*Wide Field Channel*) en la banda de frecuencia F814W, cercana a la luz infrarroja. El sensor de adquisición emplea un detector CCD de 4096 x 4096 píxeles, con un tamaño de píxel de 15 μ m.

Para el programa COSMOS se capturó un mosaico de imágenes de una porción del cielo (Figura 10), cada una de las cuales se sometió a una exposición de 2.028 segundos, en 4 etapas de 504 segundos para mejorar el efecto de la PSF.

A partir de toda esta información se construyó el dataset *COSMOS real galaxy dataset* formado por imágenes individuales de 118.992 galaxias.

Figura 10 - De izquierda a derecha, imagen del cielo explorado por el Hubble

<https://esahubble.org/images/heic0701g/> y mosaico creado con las capturas individuales terrestres (Koekemoer et al., 2007)



En el dataset, para cada imagen de galaxia se dispone de la PSF con la que se puede realizar un proceso de deconvolución, y así corregir la aberración producida por la función de dispersión de punto.

Los valores de la PSF de cada imagen se crearon usando el software de simulación *Tiny Tim* (Krist et al., 2011). Dicho software es una evolución del software original *TIM*, usado específicamente para el telescopio espacial Hubble, cuya importancia y criticidad fue determinante tras la detección del defecto del espejo principal del telescopio una vez se puso en órbita. La complejidad del uso de *TIM* derivó en la creación de un nuevo software, *Tiny Tim*, más fácil de utilizar y con nuevas funcionalidades.

4.1.2. Arquitecturas de redes neuronales

En el estado del arte se relatan numerosas arquitecturas de redes neuronales que pueden ser factibles para entrenar modelos que simulen la deconvolución de imágenes astronómicas. Como se ha explicado en la sección 2.3, a lo largo de los últimos años los trabajos realizados en este ámbito científico han sido numerosos, empleando distintas arquitecturas de red.

Para este proyecto se han seleccionado dos modelos que están dando muy buenos resultados en las operaciones de deconvolución, y otra arquitectura probada en un trabajo reciente que se usó con imágenes simuladas y se obtuvieron resultados preliminares con datos reales. En concreto, se han utilizado tres arquitecturas para evaluar el proceso de deconvolución.

- A. Arquitectura AEPP2 (Hernández Afonso, 2022)
- B. Arquitectura U-Net (Ronneberger et al., 2015)
- C. Arquitectura DIDN (Yu et al., 2019)

En la sección 5.2 se describen en detalles todas ellas y las modificaciones realizadas para las pruebas ejecutadas en este proyecto.

4.2. Evaluación de resultados

El contexto de trabajo en el que se desarrolla este proyecto hace necesario el uso de unos criterios de evaluación adecuados. Las imágenes de partida, provenientes del telescopio espacial Hubble, no deben ser alteradas de forma que se modifique su fisionomía o se altere la información descriptiva, pues la deconvolución con la PSF obtendría resultados inesperados. Por ello se descarta el uso de técnicas de *data augmentation* para aumentar la extensión del *dataset*. El dataset está compuesto por las imágenes capturadas con el Hubble

y la imagen deconvolucionada con su PSF, que será la imagen objetivo o *target*. La red deberá aprender a “simular” el proceso de deconvolución, por lo que se deberá de evaluar el error cometido en estos cálculos.

Teniendo en cuenta esta particularidad, se van a emplear las siguientes métricas:

4.2.1. Métricas de evaluación

MAE: *Mean absolute error* o error absoluto medio, utilizado en el entrenamiento y definido por:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - x_i| \quad y_i: \text{Imagen calculada } x_i: \text{Imagen objetivo}$$

MSE: *Mean squared error* o error cuadrático medio, definido por:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2 \quad y_i: \text{Imagen calculada } x_i: \text{Imagen objetivo}$$

RMSE: Raíz de error cuadrático medio, definido por:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2} \quad y_i: \text{Imagen calculada } x_i: \text{Imagen objetivo}$$

MSE y RMSE penalizan si la predicción es muy mala y por tanto es muy sensible a errores altos, MAE no penaliza tanto los valores muy distintos.

PSNR: *Peak Signal-to-Noise Ratio*, definido por

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

$MAXi$ = Máximo valor de los píxeles, representados con B bits por muestra

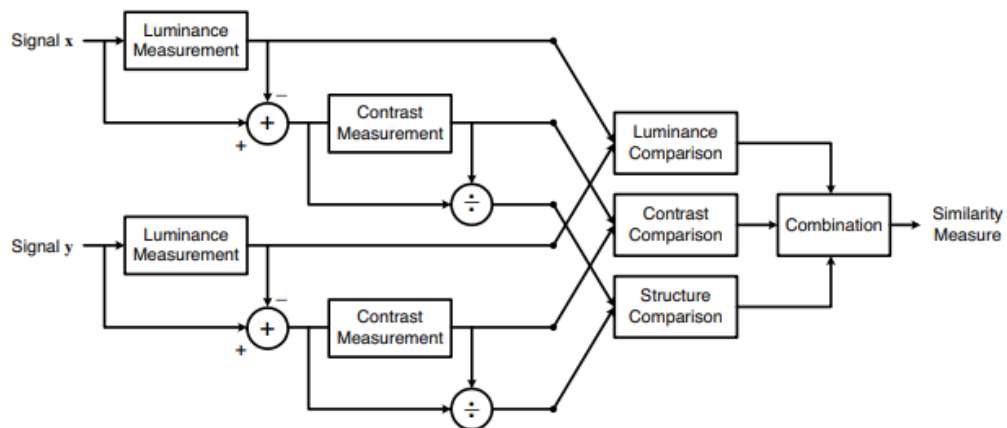
$$MAXi = 2^B - 1$$

La PSNR mide la relación entre la máxima energía posible de una señal y el ruido que le afecta. Se expresa en escala logarítmica en unidades de decibelios, y en su cálculo se emplea el MSE. Los valores normales oscilan entre 30 dB y 50 dB, siendo mejores cuanto mayor sean.

El valor PSNR se utiliza, por ejemplo, para evaluar de forma cuantitativa la calidad de algoritmos de compresión de imágenes y vídeo como MPG. De la misma forma puede ser utilizada esta métrica para evaluar la calidad del proceso de deconvolución de imágenes de galaxias, pues es esencialmente el mismo caso de uso.

SSIM: Structural Similarity Index. Esta medida propuesta por (Wang et al., 2004) se aproxima a lo que la visión humana realiza para comparar dos imágenes. En lugar de utilizar los datos en bruto de los valores de la imagen, emplea una combinación de luminancia, contraste y estructura con el objetivo de evaluar la similitud estructural entre los píxeles vecinos.

Figura 11 - Diagrama del cálculo de SSIM. Fuente: (Wang et al., 2004)



Cada uno de estos tres elementos se calcula de la siguiente manera:

$$\text{Luminancia}(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$\text{Contraste}(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$\text{Estructuras}(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

donde

μ_x es la media de los píxeles de la imagen x

μ_y es la media de los píxeles de la imagen y

σ_x^2 es la varianza de la imagen x

σ_y^2 es la varianza de la imagen y

σ_{xy} es la covarianza

C_1, C_2 y C_3 son coeficientes de estabilización para las divisiones

De esta forma, el cálculo de SSIM se expresa de la siguiente manera:

$$SSIM(x, y) = l(x, y)^\alpha \cdot c(x, y)^\beta s(x, y)^\gamma$$

α, β, γ son pesos para ponderar cada uno de los componentes

El valor de SSIM oscila entre 0 y 1, siendo 1 cuando dos imágenes son idénticas. Aunque depende de cada caso, un valor superior a 0,97 indica que las imágenes son muy similares.

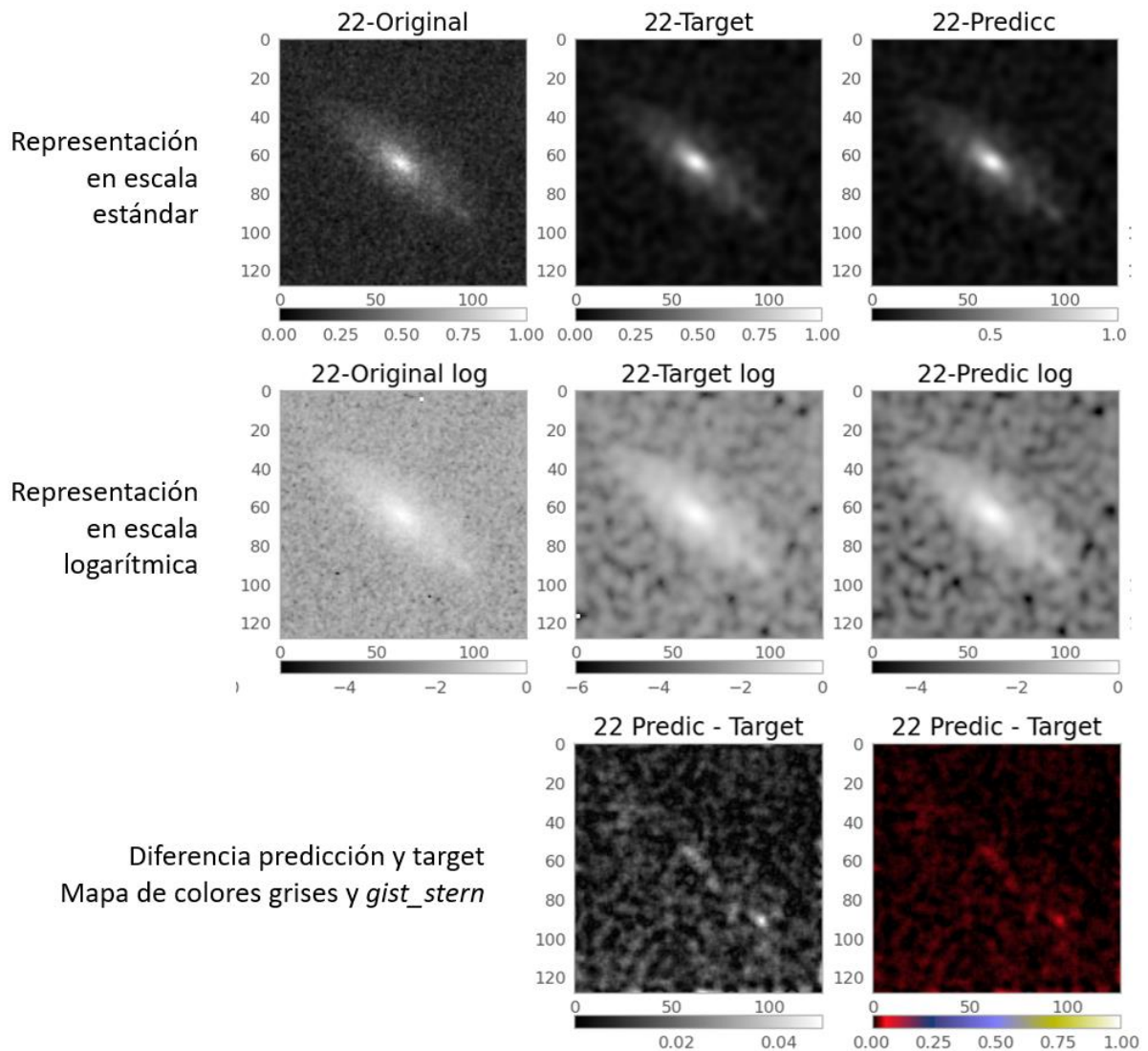
Esta métrica es muy sensible en el sentido de que un pequeño cambio en el valor puede suponer una importante mejora o empeoramiento. Por ejemplo, pasar de 0,94 a 0,95 puede ser muy significativo en el resultado final, por lo que los esfuerzos en optimizar alguna décima o incluso centésimas merecen la pena ser tenidos en cuenta.

Para el ámbito de este proyecto, el uso de SSIM permite evaluar el proceso de deconvolución simulado en la red neuronal, es decir, determinar si la imagen calculada por la red a partir de la original es suficientemente similar al *target*.

4.2.2. Comparativa visual

Se va a proporcionar un mecanismo para una evaluación visual de los resultados, comparado la imagen de calculada con la imagen objetivo. En las pruebas preliminares y sobre todo durante el estudio del contexto de trabajo, se ha observado que los valores de las imágenes capturadas con el telescopio Hubble comprenden un rango muy pequeño, también tras el proceso de normalización al rango [0,1] al que se someten, y que se detalla en el apartado 5.1.2. Esta circunstancia requiere que la evaluación visual se haga modificando la escala de representación. Por una parte, se ha considerado oportuno mostrar el resultado de la ejecución del modelo sobre las imágenes de test en escala logarítmica (figura 12), lo cual permite apreciar de mejor manera algunas estructuras de las galaxias. Por otra parte, se ha realizado una resta de valores de la imagen *objetivo* y la predicha por el modelo (valor residual), mostrándose en un mapa de color visualmente significativo.

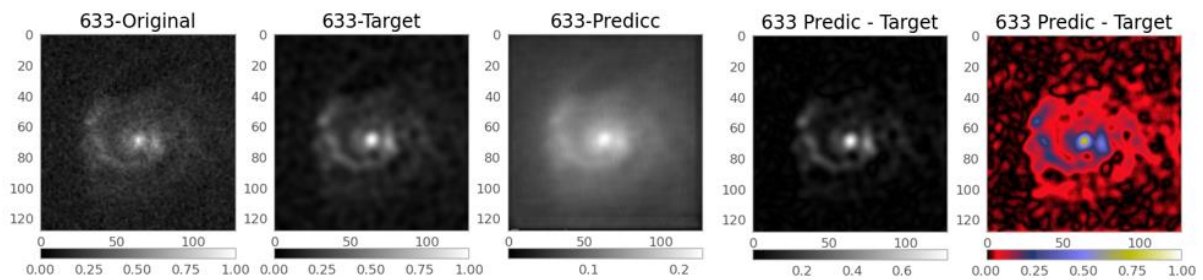
Figura 12 – Resultados de una prueba. Primera fila y de izquierda a derecha, imágenes originales, objetivo y predicha. Fila central y de izquierda a derecha, imágenes originales, objetivo y predicha en representación logarítmica. Fila inferior de izquierda a derecha, diferencias entre la imagen predicha y la imagen objetivo, con mapas de color escala de grises y “gist_stern”.



La visualización de la imagen residual con el mapa de color *gist_stern* permite apreciar mejor a simple vista el comportamiento del modelo. Una predicción muy buena, como la mostrada en la figura 12, tendrá una imagen residual con tonos negros. Las predicciones peores irán apareciendo en tonos rojizos pasando a azulados y amarillentos si estas son muy incorrectas.

En la figura 13, con el uso del mapa de color *gist_stern* se puede apreciar claramente las deficiencias en la predicción realizada.

Figura 13 - Ejemplo de predicción deficiente. De izquierda a derecha, imagen original, target, predicción, diferencia predicción-target en escala de grises y diferencia predicción-target en mapa de color “*gist_stern*”



5. Desarrollo de la comparativa

Una vez establecido el ámbito de estudio, la problemática a tratar y el mecanismo para evaluar los resultados, se describen a continuación los detalles de los datos que se han empleado, los elementos usados en la construcción de las redes neuronales y el nivel de complejidad de estas. También se describen los hiperparámetros empleados en las pruebas y los datos estadísticos que se calcularán para formalizar la evaluación de los modelos.

5.1. Construcción del dataset

Como se ha indicado en la sección 4.2.1, las imágenes de partida para este proyecto se han obtenido del dataset “*COSMOS real galaxy dataset*” formado por 118.992 imágenes de galaxias capturadas con el telescopio espacial Hubble (*HTS*) como parte del programa *COSMOS, The Cosmic Evolution Survey*. Las imágenes están divididas en 2 grupos en función de su magnitud aparente de la galaxia, menor a 23.5 y menor a 25.2. Para las pruebas se han seleccionado imágenes de ambos grupos evitando la ocurrencia de repeticiones ya que todas las capturas del grupo 25.2 están incluidas en el grupo 23.5. Esta distinción ha sido posible ya que cada imagen de galaxia está identificada con un número único.

El conjunto de imágenes incluye los valores de la PSF individuales de cada una de estas capturas del HTS, los cuales fueron estimados usando el software de simulación *Tiny Tim* (Krist et al., 2011).

Todas las imágenes se almacenan en ficheros de tipo FITS (*Flexible Image Transport System*) (Chiappetti et al., 2018) que es el estándar habitual para el intercambio de información en el ámbito astronómico.

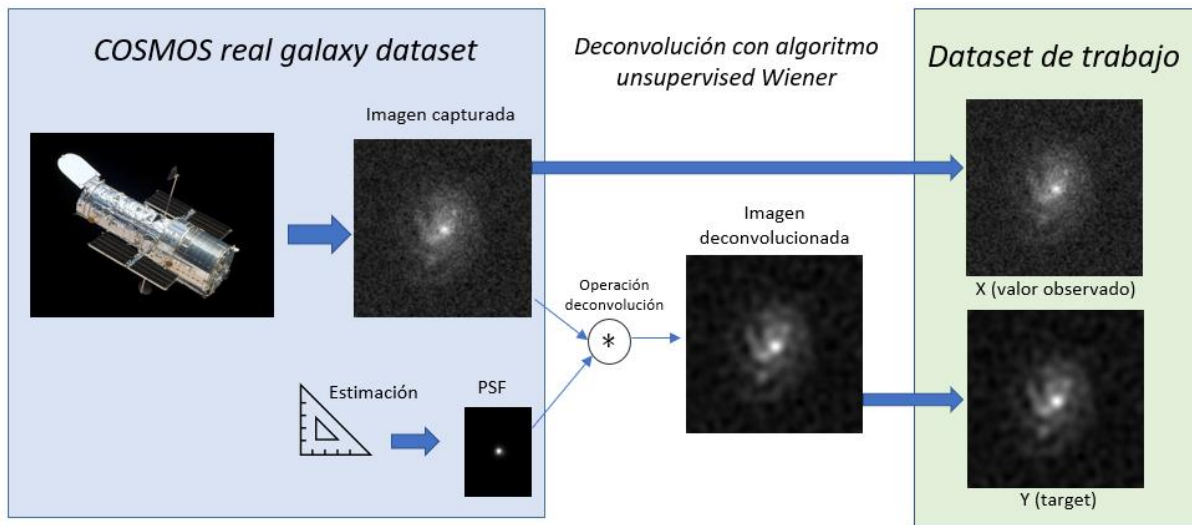
En este dataset se utilizan archivos *fits* con 1.000 imágenes cada uno, para un total de 145 capturas (118.992 imágenes de galaxias distintas). Para cada fichero existe también el asociado a las PSF calculadas para cada imagen, con una correspondencia uno a uno entre el número de fichero y la posición en el mismo. Por tanto, también hay 118.992 imágenes PSF.

El manejo de los ficheros FITS se ha realizado con la librería de *Python Astropy*⁷, de uso habitual entre la comunidad astronómica y mantenida por esta, que proporciona funciones para el acceso, lectura y escritura de este tipo de fichero, y que maneja el formato interno de los datos estructurados que contienen.

La construcción del dataset de trabajo se ha realizado creando los pares X (valor observado), Y (*target*), siendo X la imagen capturada con el telescopio *Hubble* e Y el resultado de la deconvolución de dicha imagen con su PSF.

⁷ The Astropy Project - <https://www.astropy.org/>

Figura 14 - *Proceso de creación del dataset de trabajo a partir de las imágenes capturadas y PSF del "COSMOS real galaxy dataset" y del resultado de la deconvolución.*



5.1.1. Tamaño de las imágenes

Las redes neuronales empleadas requieren que las imágenes que se les inyecten tengan un tamaño único, pues las operaciones de convolución que se hacen en las capas sucesivas operan con las dimensiones y están pensadas para unas cifras concretas. En estudios similares al del presente trabajo es habitual trabajar con imágenes de 64x64 píxeles, aunque no siempre es así y es posible encontrar valores diferentes.

Para este tipo de proyectos, partiendo de un tamaño de imagen, se diseña o se adapta la red neuronal a esos valores de entrada, con objeto de conseguir un funcionamiento libre de errores. En caso contrario se trabaja con otras dimensiones de partida.

También hay que tener en cuenta el coste computacional y de memoria, que se ven incrementados con el aumento de los datos a procesar.

Para la elección del tamaño de las imágenes se hicieron pruebas preliminares con las arquitecturas elegidas, con distintos tamaños de imagen, para comprobar que se ejecutaban, compilaban y se entrenaban correctamente.

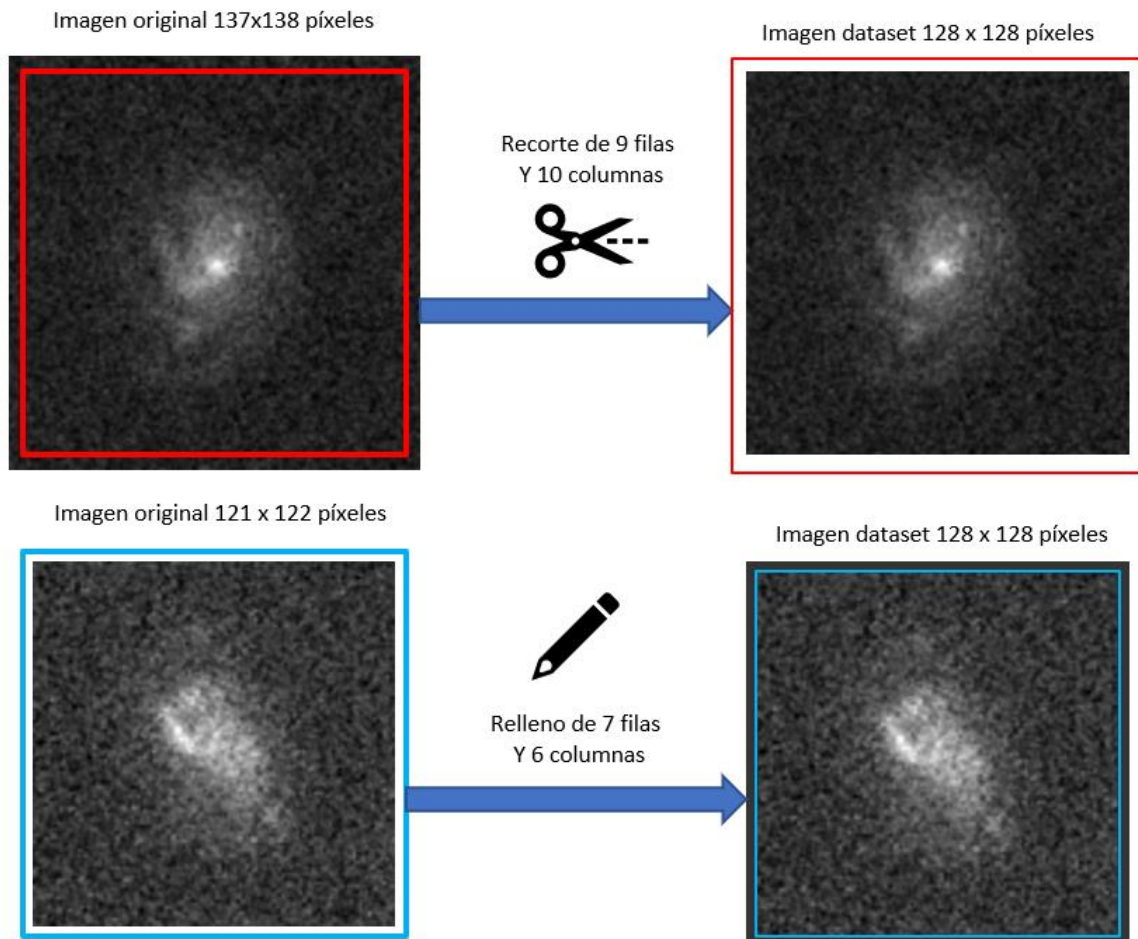
Otra circunstancia es que el dataset original *"COSMOS real galaxy dataset"* no contiene imágenes uniformes en tamaño, y las 118.992 capturas van desde unas dimensiones de 9x9

Deep Learning como alternativa a la deconvolución de imágenes de galaxias capturadas con el telescopio espacial Hubble hasta 962x962 píxeles. Sin embargo, la inmensa mayoría de las imágenes se concentran entre los 40x40 píxeles y los 160x160 píxeles.

Con estas 2 premisas se determinó el uso de imágenes de 128x128 píxeles, cifra verificada como válida para emplear en las arquitecturas de redes seleccionadas.

A consecuencia de la variabilidad de los tamaños de las imágenes capturadas con el telescopio Hubble, la cantidad de aquellas de dimensiones 128x128 es insuficiente, por lo que se ha determinado utilizar algunas más grandes y otras más pequeñas, recortándolas o rellenándolas en filas y columnas.

Figura 15 - *Proceso de recorte y rellenado para adaptar las imágenes a tamaño 128 x 128 píxeles*



Esta tarea debe respetar la información de la imagen, es decir, los valores de cada píxel deben permanecer invariables pues es una información fundamental para el desarrollo del proyecto.

Por este motivo, no se realiza ningún tipo de redimensionado al valor 128x128 que pueda alterar los datos. También se debe de tener en cuenta la morfología de las imágenes. Al tratarse de capturas de galaxias, estas se ubican en el centro de la imagen, sin llegar a tocar los márgenes, por lo que es factible eliminar algunas filas y columnas sin perder información valiosa, y de la misma forma es posible añadirlas sin afectar a la información válida. En este último caso, a los píxeles de las filas y columnas añadidas se les asigna el valor 0.

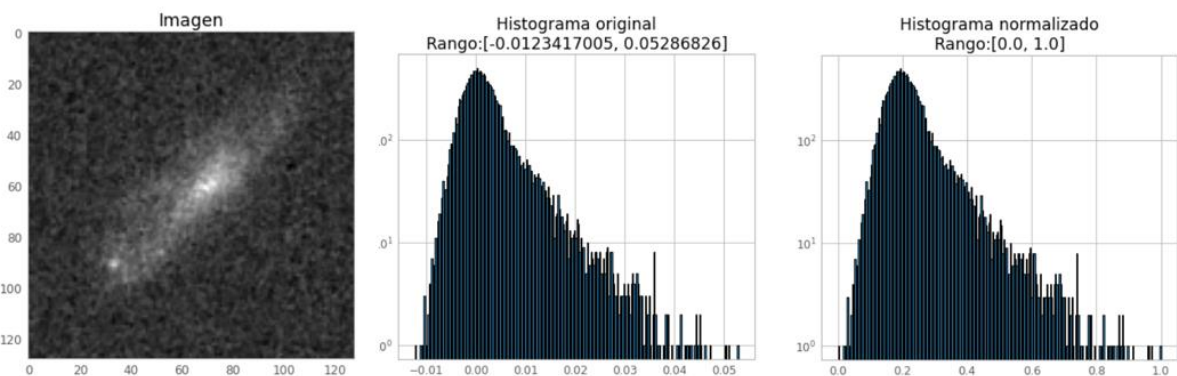
El proceso de recorte y rellenado se hace alternativamente a izquierda y derecha, y por arriba y por abajo para que la imagen quede centrada. Se ha aplicado a las imágenes cuyas dimensiones se encuentran entre 110x110 píxeles y 140x140 píxeles.

Tras estas operaciones de armonización de información, se ha construido un dataset de 9.625 imágenes de 128 x 128 píxeles, divididos en datos de entrenamiento, datos de validación y datos de test, en unos porcentajes de 70%, 17% y 13% respectivamente.

5.1.2. Normalización de valores

Los píxeles de las imágenes originales contienen valores positivos y negativos, pero dentro de un rango muy reducido. Esta circunstancia puede provocar que los cálculos realizados en el entrenamiento no sean apropiados, teniendo en cuenta que lo habitual en redes neuronales es utilizar valores entre 0 y 1. Por ello, en el proceso de carga de datos en memoria antes del entrenamiento, los valores de las imágenes son normalizados al rango [0,1] sin que por ello se vea afectada la imagen.

Figura 16 - Imagen junto a su histograma original e histograma con valores normalizados [0,1]



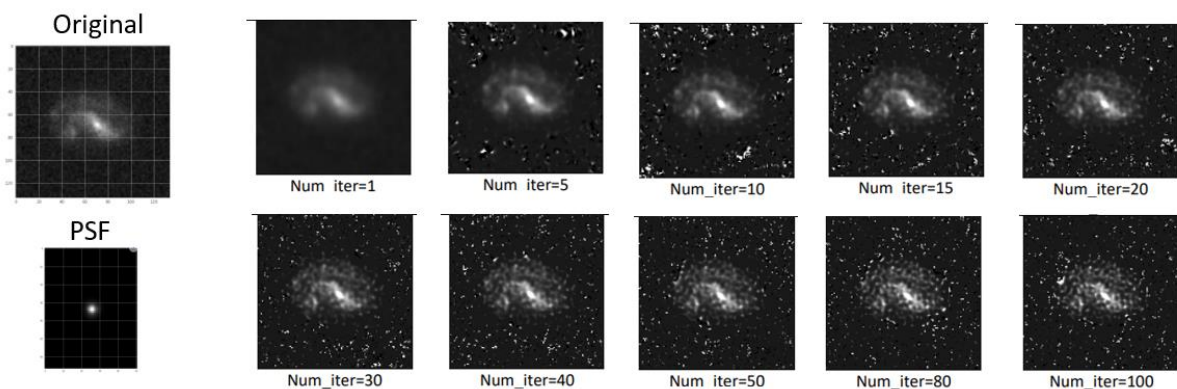
5.1.3. Procedimiento de deconvolución

El planteamiento de este proyecto se basa en la construcción de una red neuronal que sea entrenada para simular un proceso de deconvolución de una imagen de una galaxia. Para ello el entrenamiento parte de la imagen original capturada con el telescopio Hubble y la deconvolucionada a partir de su PSF. Esta última forma el *target*, que es el objetivo de los cálculos de la red.

Para el proceso de deconvolución existen distintos algoritmos matemáticos de los cuales se han probado dos: *Richardson-Lucy* y *Wiener*, utilizando las implementaciones existentes en la librería *Scikit*⁸

Con *Richardson-Lucy* se realizaron pruebas variando el parámetro *num_iter* desde 1 hasta 100, en intervalos de 5. Esta variable hace el papel de regularización. El número de iteraciones permite mejorar la reconstrucción en cuanto a su resolución espacial, pero provoca un aumento del ruido. Por ello es necesario elegir una cifra que pondere ambas características y permita obtener unos resultados óptimos.

Figura 17 - Deconvolución con algoritmo *Richardson-Lucy* con distintos valores de regularización.



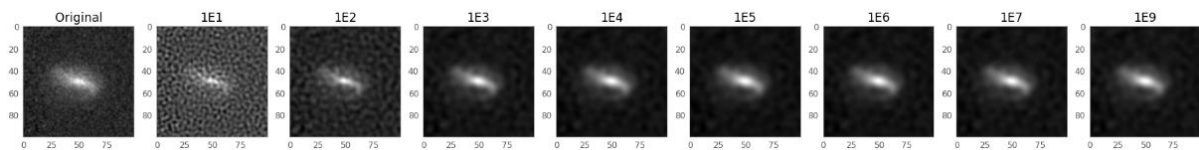
Como muestra la Figura 17 el algoritmo *Richardson-Lucy* introduce alteraciones en las imágenes, aunque en distinto grado dependiendo de cada caso concreto.

⁸ Librería Scikit que incluye el módulo *restoration*, el cual implementa los algoritmos Richardson-Lucy y Wiener. <https://scikit-image.org/docs/dev/api/skimimage.restoration.html#module-skimage.restoration>

El algoritmo *Wiener* en *scikit* tiene dos implementaciones, *Wiener* y *Unsupervised_wiener*. La primera utiliza un parámetro de regularización *balance* que intenta equilibrar la restauración de la imagen original, pero que depende de dicha imagen, por lo que es necesario especificarlo para cada caso. La segunda hace una estimación automática de esta regularización con lo que no es necesario adaptar a cada imagen. Esta opción no supervisada es la que se ha empleado para hacer pruebas de deconvolución ya que permite automatizar el proceso independizándolo del criterio visual del científico.

Debido a la formulación matemática empleada por el algoritmo *Wiener*, los valores tan bajos de los píxeles de las imágenes capturadas hacen que el cálculo sea deficiente. Por este motivo, se han hecho pruebas multiplicado previamente dichos valores por 10, 100, 1000, 10000, 100000 y 1000000 (1E1, 1E2, 1E3, 1E4, 1E5, 1E6) para determinar qué factor de aumento funciona mejor con el algoritmo.

Figura 18 - Deconvolución con algoritmo *unsupervised wiener* con valores multiplicados por 1E1, 1E2, 1E3, 1E4, 1E5 y 1E6.



El multiplicador más apropiado también depende del tipo de imagen y sus características. Así, por ejemplo, para galaxias lenticulares que se muestran muy pequeñas, la deconvolución con un factor de multiplicación concreto produce oscilaciones de Gibbs a su alrededor, algo que con galaxias espirales no es tan visible.

Con las pruebas realizadas se concluyó que una deconvolución exitosa para cada imagen requiere de parametrización específica para cada una de ellas, algo que en este escenario no es viable. Por ello, aunando las distintas posibilidades para maximizar el éxito del proceso, se determinó utilizar el algoritmo *Unsupervised Wiener*, multiplicando previamente los valores de las imágenes por 1E3. Tras la deconvolución, los valores son devueltos a su escala original dividiendo por ese mismo factor de multiplicación.

La deconvolución con parámetros únicos para todas las imágenes hace que el éxito de su ejecución no sea uniforme en todas las imágenes, produciendo reconstrucciones satisfactorias en algunos casos y no tan buenas en otros. Sin embargo, para el objetivo final de este proyecto, esta circunstancia no es determinante, pues la red neuronal se entrena para reconstruir imágenes y compararlas con la imagen objetivo, independientemente de que ese *target* sea mejor o peor.

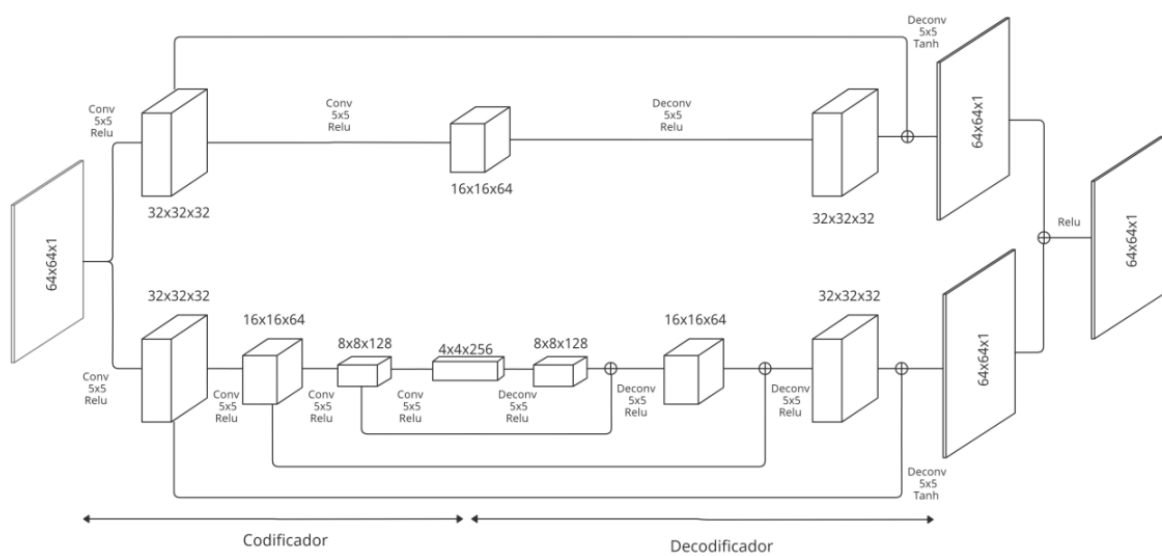
5.2. Arquitecturas empleadas

Como se ha anticipado en el apartado 4.1.2, se han empleado 3 arquitectura de redes neuronales para realizar pruebas.

5.2.1. Arquitectura AEPP2

Se trata de un modelo basado en la propuesta de (Liu & Lam, 2018) que utiliza *autoencoders* en dos ramas, en las que se realizan operaciones de convolución y deconvolución. La arquitectura AEPP2 fue empleada por (Hernández Afonso, 2022) considerando algunas modificaciones para una mejor adaptación a la eliminación de ruido.

Figura 19 - Arquitectura AEPP2 - (Hernández Afonso, 2022)



Este modelo consta de 2 ramas, la superior incluye 2 operaciones de convolución (codificación o compresión) y otras 2 de deconvolución (decodificación o descompresión); la inferior consta de 3 operaciones de cada tipo.

La idea es transformar la imagen original en una representación compacta intentando eliminar el ruido (compresión) y posteriormente recomponer la imagen original sin las características eliminadas (descompresión). La existencia de 2 ramas se justifica en poder incluir en el mismo proceso dos niveles de eliminación de ruido. Así, la rama inferior produce una mayor compresión, eliminando más cantidad de ruido, pero recordando menos detalles. La rama superior, de menor capacidad de compresión, elimina menor ruido, pero permite recuperar mejor la imagen original.

Para evitar la pérdida de información motivada por el uso de *autoencoders*, se incluyen *skip-connections* que son conexiones entre capas del inicio y del final para conservar las características de las imágenes. Este problema tiene su origen en el uso de demasiada profundidad en el procesamiento, lo que puede provocar un desvanecimiento de gradiente. De hecho, el modelo de partida de (Liu & Lam, 2018) limita esta profundidad

En el trabajo indicado se experimenta con esta red para eliminar el ruido en imágenes de galaxias generadas artificialmente y con una estimación de PSF única para todos los casos. Como mejoras a futuro se indicaba la posibilidad de utilizar esta arquitectura con imágenes reales de galaxias. Por este motivo se ha elegido para el desarrollo del presente trabajo.

En la Figura 19 se muestra la arquitectura que mejores resultados ha dado en las pruebas realizadas en la referencia mencionada, y que es el modelo que se ha empleado en el presente proyecto.

La tabla 1 contiene los detalles de construcción de la red y su nivel de complejidad

Tabla 1 - Número de filtros usados en la construcción del modelo AEPP2.

Modelo	Nº filtros sucesivos	Nº parámetros totales
AEPP2	Rama superior: 32, 64, 32 Rama inferior: 32, 64, 128, 256, 128, 64, 32	1.444.674

5.2.2. Arquitectura U-Net

Este modelo utiliza una filosofía parecida a los *Autoencoders* ejecutando dos etapas, una de codificación donde se extraen las características propias de la imagen, y una segunda etapa de decodificación, simétrica a la primera, y donde se reconstruye la imagen original. Lo más significativo de este planteamiento es la comunicación entre capas simétricas, lo cual transmite la información de la entrada a la salida, evitando que se pierdan demasiadas características.

Otro elemento fundamental es la ejecución tras cada etapa de un *maxpooling*, que reduce la dimensionalidad sin perder información importante.

El origen de esta arquitectura hay que buscarlo en la mejora de imágenes médicas (Ronneberger et al., 2015) pero hay numerosos estudios que la han empleado para procesos de deconvolución, y en concreto para imágenes astronómicas.

La arquitectura original mostrada en la Figura 6 contempla un nivel de profundidad de 64, que viene dado por los filtros existentes en la primera capa. En las capas sucesivas el número de filtros se van duplicando hasta llegar a 1.024, momento en el que se pasa a la etapa de decodificación. Este modelo contempla un total de 31.031.685 parámetros, lo cual implica un coste computacional nada despreciable.

Para este proyecto se han construido otros modelos de U-Net con distintos niveles de profundidad para probar los resultados en cada uno de ellos. Se ha partido de una implementación base en *Keras* realizada por (Roy, 2021/2022) que simplifica la arquitectura original dejándola en un nivel de profundidad de 16, que da lugar al uso sucesivo de 16, 32, 64, 128, 256, 128, 64, 32, 16 filtros en cada etapa, en la codificación y decodificación.

En la Tabla 2 se muestran los valores empleados en cada uno de los modelos construidos:

Tabla 2 - Número de filtros usados en distintas construcciones de modelos U-Net.

Modelo	Nº filtros sucesivos	Nº parámetros totales
U-Net 4	4, 8, 16, 32, 64, 32, 16, 8	121.725
U-Net 8	8, 16, 32, 64, 128, 64, 32, 16, 8	485.813

U-Net 16	16, 32, 64, 128, 256, 128, 64, 32, 16	1.941.093
U-Net 32	32, 64, 128, 256, 512, 256, 128, 64, 32	7.760.069
U-Net 64	64, 128, 256, 512, 1024, 512, 256, 64	31.031.685
U-Net 128	128, 256, 512, 1024, 2048, 1024, 512, 256, 128	124.109.573

5.2.3. Arquitectura DIDN

Este modelo propuesto en (Yu et al., 2019) parte de la estructura de una U-Net pero añade un subprocesamiento iterativo del proceso de convolución (DUB), afinando y mejorando la eliminación de ruido.

La red DIDN mostrada en la Figura 8 del capítulo 2.3 incluye 4 etapas de procesado:

- Extracción de características a través de *kernel 3x3* y *stride 2*.
- Reducción y escalado iterativo (subprocesamiento DUB). El proceso iterativo realizado con la estructura DUB (Figura 8.b) es en esencia una red U-Net, y su iteración consigue mejorar la eliminación de ruido en las imágenes.
- Reconstrucción: Las salidas de los subprocesamientos se procesan en 9 etapas de convolución
- Mejora: En la etapa final se realiza un ampliado para obtener la imagen final sin ruido.

La iteración sucesiva del subprocesamiento implica una elevada complejidad en el diseño y en las necesidades de cómputo, aspecto que resulta limitante en cuanto a las pruebas a realizar.

Se ha utilizado la implementación realizada en *PyTorch* por (Yu, 2019/2022). Sin embargo, el código se ha migrado a *Keras* para unificar el método de trabajo y las pruebas con el resto de las arquitecturas empleadas en este proyecto.

El modelo original incluye 256 filtros en la primera etapa lo cual crea una red neuronal muy extensa y compleja. Por este motivo se ha construido otra red simplificada con 16 filtros en la entrada.

Tabla 3 - Número de filtros y parámetros de arquitectura DIDN con distinto número de bloques de sub procesado DUB.

Modelo	Nº filtros sucesivos	Nº parámetros totales
DIDN16 1DUB	16, (16, 32, 64, 128, 32, 64, 16, 16), 16	1.507.360
DIDN16 2DUB	16, (16, 32, 64, 128, 32, 64, 16, 16) x2, 16	2.005.792

Solo se han entrenado dos variaciones con 1 y 2 bloques DUB, ya que, aunque se han construido modelos con hasta 10 DUB, los entrenamientos con más de tres unidades de subproceso no convergían con ninguna combinación de valores de *epoch*, *batch size* y *learning rate*. Particularmente, cada bloque DUB añade al modelo aproximadamente 500.000 nuevos parámetros.

5.3. Hiperparámetros

La elección de los hiperparámetros es determinante para la ejecución exitosa de los entrenamientos y para evaluar el rendimiento de las arquitecturas. Los valores seleccionados se han empleado en todos los modelos para poder comparar en las mismas condiciones de ejecución y así determinar la efectividad final de cada arquitectura.

Optimizador (*optimizer*)

El proceso de optimización intenta mejorar el comportamiento de la tasa de aprendizaje, modificando su valor en función de determinadas condiciones. Para todas las pruebas se ha empleado el optimizador *Adam* por ofrecer buenos resultados, con valores de *learning rate* de 1.0E-3, 1.0E-4, 1.0E-5 y 1.0E-6 según las particularidades de cada entrenamiento.

Función de pérdida (*loss*)

En el apartado 4.2.1 se indicó que el resultado de los test tras el entrenamiento se iba a evaluar, entre otras, con la métrica SSIM que es una medida muy apropiada de comparación de imágenes. Parece lógico por tanto que se use esta medida como función de pérdida en el entrenamiento para ir comparando la ejecución del modelo con el target.

El valor de SSIM oscila entre 0 y 1, siendo 1 la situación en que dos imágenes son idénticas. Para usar esta métrica como función de pérdida, hay que invertir su valor, restándolo de 1 (el objetivo de una función de pérdida es llegar al valor 0).

Para este proyecto se ha empleado la implementación SSIM que hace la librería *TensorFlow*⁹

Métrica (metrics)

La métrica empleada en el entrenamiento es *Mean Absolute Error* (MAE) por ser una opción de uso habitual en entrenamientos de redes neuronales. Esta función permite valorar el aprendizaje de la red, algo que ocurrirá cuando este valor vaya descendiendo paulatinamente.

Early Stop

En el apartado 4.2.1 se indicó que un valor de 0.97 para SSIM se podría considerar como correcto. Por tanto, se ha tomado esta cifra como el objetivo a conseguir en el entrenamiento, que, como se ha dicho anteriormente, se traduce en un valor de *loss* de

$$loss_{ideal} = 1 - 0,97 = 0,03$$

En la ejecución de las pruebas se ha implementado la técnica de *early_stop* que detiene el entrenamiento cuando el valor de *loss* < 0,03

En realidad, tras algunas pruebas se observó que este objetivo se podía alcanzar por lo que se modificó su valor y se fijó en 0,02. Valores inferiores son ya difíciles de alcanzar y no reportan mejoras significativas.

$$loss_{real} = 1 - 0,98 = 0,02$$

Ciclos de entrenamiento (epoch)

Inicialmente se ha planificado un ciclo de entrenamiento de 50 *epoch*, pero tras vislumbrar la posibilidad de alcanzar un valor de 0,98 en el SSIM de entrenamiento, se aumentó esta cifra paulatinamente a 100, 150 y 200 *epoch*, permitiendo además la detención con *early_stop*.

⁹ Implementación de SSIM en Tensorflow: https://www.tensorflow.org/api_docs/python/tf/image/ssim

Tamaño de entrenamiento (*batch size*)

El valor de este hiperparámetro hace que el comportamiento del entrenamiento pueda variar a mejor o a peor. Por ese motivo es necesario repetir las pruebas con distintos valores de *epoch*. Para todas las arquitecturas se han realizado entrenamientos con tamaños de *batch size* de 32, 63, 128 y 256.

5.4. Entrenamiento y test

El dataset original está formado por 9.625 imágenes de 128 x 128 píxeles, divididas en las proporciones mostradas en la tabla 4.

Tabla 4 - Distribución de datos de entrenamiento, validación y test.

Imágenes	Número	Porcentaje
Datos entrenamiento	6.780	70%
Datos validación	1.695	17%
Datos de test	1.150	13%

Todos los modelos se han entrenado con las mismas imágenes de entrenamiento y validación, aunque en cada ejecución se han mezclado al inicio utilizando el parámetro *shuffle* de *Keras*. Igualmente, todas las arquitecturas se han testeado con las mismas imágenes y así poder comparar eficientemente los resultados.

5.5. Recursos hardware y software

Las pruebas de entrenamiento se han ejecutado en dos entornos: local y online.

- Entorno local: Ordenador con CPU AMD RYZEN 5 5600X y GPU NVIDIA RTX 3060Ti 8GB
 - Cuda 12.0, nvidia-smi: 527.56
- Entorno online: Google Colab. GPU NVIDIA A100-SXM4 40GB
 - Cuda 11.6, nvidia-smi: 512.47.03

Toda la parte de programación se ha realizado en Python v3.9.12 usando las siguientes librerías:

- TensorFlow v2.10.0¹⁰ : Construcción y gestión de redes neuronales.
- Keras v2.10.0 ¹¹: Construcción y gestión de redes neuronales.
- Scikit-image v0.19.2 ¹²: Manejo de imágenes.
- Numpy v1.21.5 ¹³: Funciones varias para gestión de datos.
- Matplotlib v3.5.1 ¹⁴: Generación de gráficas.
- Astropy 5.0.4¹⁵: Gestión de archivos FITS.

¹⁰ TensorFlow - <https://www.tensorflow.org/>

¹¹ Keras - <https://keras.io/>

¹² Scikit-image - <https://scikit-image.org/>

¹³ Numpy - <https://numpy.org/doc/stable/index.html>

¹⁴ Matplotlib - <https://matplotlib.org/>

¹⁵ Astropy - <https://www.astropy.org/>

6. Discusión y análisis de resultados

A continuación, se muestran los resultados de las pruebas realizadas con las tres arquitecturas elegidas y sus modificaciones. En primer lugar, se hace una valoración del comportamiento de cada modelo por separado y posteriormente una evaluación conjunta.

En el Anexo A se incluyen los datos estadísticos y resultados completos de los entrenamientos.

6.1. Configuración de hiperparámetros

En todas las pruebas se han utilizado los mismos hiperparámetros, salvo la *learning rate* que en algunos casos se ha modificado para conseguir la convergencia del modelo.

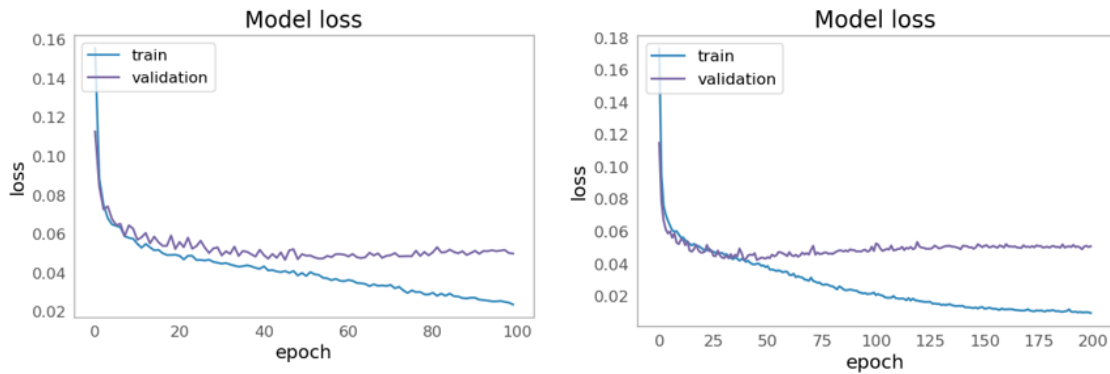
- Optimizador: Adam.
- *Learning rate*: 1.0E-3, 1.0E-4, 1.0E-5, 1.0E-6
- Función de pérdida loss: SSIM
- *Early Stop*: SSIM < 0.2
- Métricas de entrenamiento: *Mean absolute error*
- *Batch size*: 32, 64, 128 y 256

6.2. Evaluación de la arquitectura AEPP2

Se han realizado 8 experimentos con cada valor de *batch size*: 32, 64, 128 y 256 en dos tandas de 100 y 200 *batch*. La ejecución con 200 *batch* se justifica en la pretensión de obtener mejores resultados, cosa que finalmente no sucedió. Incluso observando las gráficas del entrenamiento (Figura 20) se puede afirmar que a partir de la *epoch* 40 la convergencia prácticamente se alcanza, no mejorándose casi los resultados.

También se constata que con tamaños más pequeños de *batch size*, el modelo produce cierto *overfitting*, aspecto que se ve corregido con tamaños superiores (Figura 34 – Anexo A).

Figura 20 – Arquitectura AEPP2. Gráficas de entrenamiento con batch size 64. A la izquierda entrenamiento con 100 epoch, a la derecha con 200 epoch.



Se realizaron los entrenamientos con un *learning rate* de 1.0E-3, aunque se probaron valores inferiores sin obtener mejores resultados, por lo que se consideró este como el mejor.

En todas las pruebas los resultados son muy similares, variando el valor de SSIM en solo algunas milésimas. El modelo con mejor comportamiento es el ejecutado con tamaño de *batch size* igual a 128 ya que ofrece además las mejores cifras para PSNR y MSE.

Tabla 5 - Resultados del modelo con mejores métricas de la arquitectura AEPP2.

Modelo AEPP2. 1.444.674 parámetros					Evaluación entrenamiento			Predicciones			
MODELO	EPOCH	BATCH SIZE	OPT	LR	LOSS (1-SSIM)	ME-TRIC (MAE)	T (seg)	MSE	RMSE	PSNR	SSIM
AEPP2	100	128	Adam	1.0E-3	0,0571	0,0154	819	37,8199	6,1498	32,3536	0,9540

En el Anexo A están reflejados los resultados de todas las pruebas realizadas.

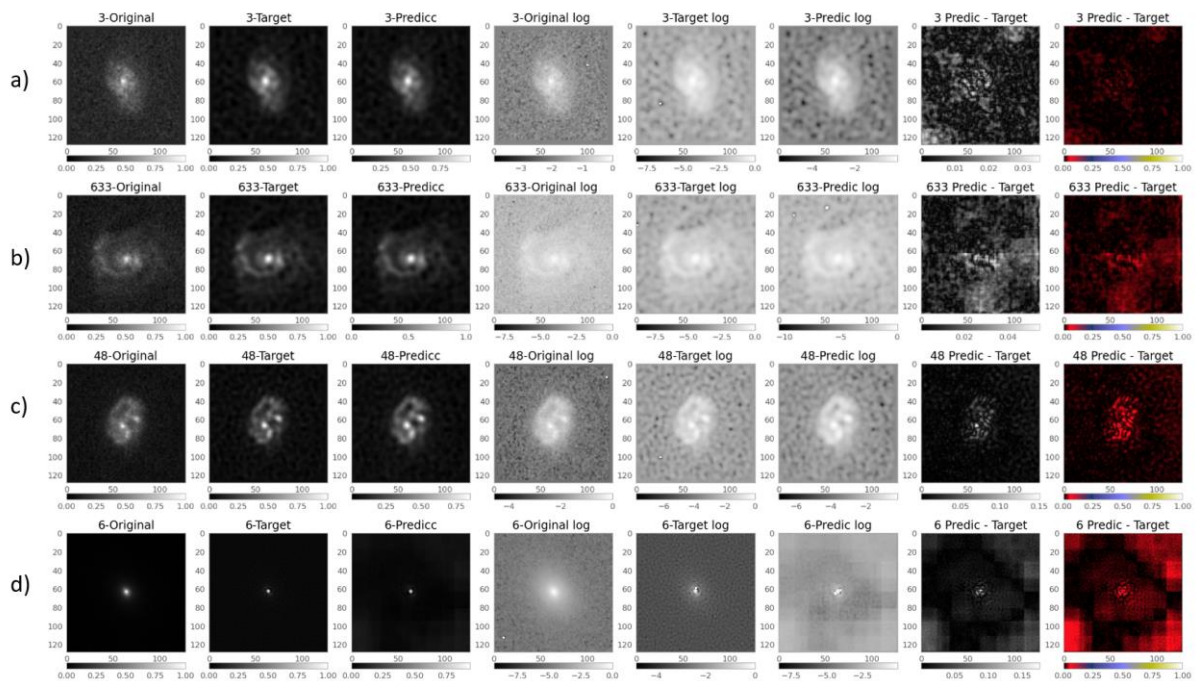
Respecto al resultado visual y en consonancia a los valores de SSIM obtenidos, las reconstrucciones realizadas son generalmente buenas, aunque con diferencias según el tipo de galaxia en cuestión. Sin embargo, no es trivial encontrar un patrón para mejores o peores resultados en función de la fisionomía de la galaxia.

En la Figura 21 se pueden ver algunos ejemplos del comportamiento del modelo. En 21.a) se realiza una reconstrucción muy buena. El residuo que queda al hacer la diferencia entre la

Deep Learning como alternativa a la deconvolución de imágenes de galaxias capturadas con el telescopio espacial Hubble

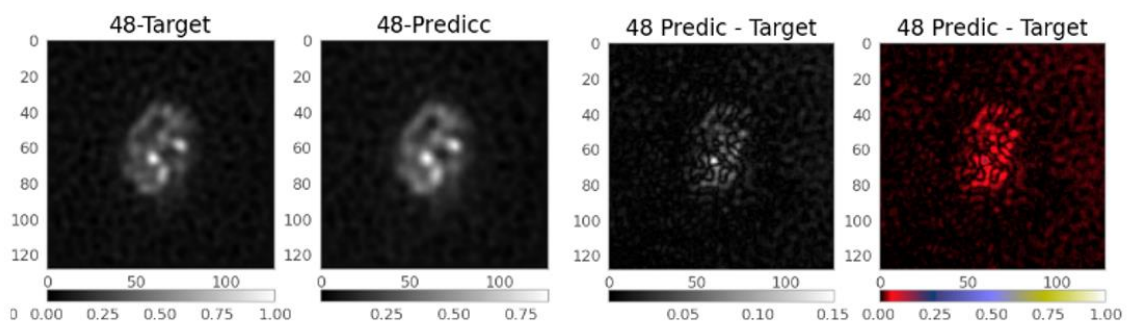
imagen predicha y la imagen objetivo muestra valores bajos, sin apreciarse estructuras resaltadas de la galaxia.

Figura 21 - Arquitectura AEPP2. Ejemplos de predicciones. a) reconstrucción óptima. b) Reconstrucción eficiente de la estructura de la galaxia. c) Reconstrucción deficiente de la estructura de la galaxia. d) Efecto cuadrícula. De izquierda a derecha, imagen original, target y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y target, en escala de grises y en mapa de color "gist_stern".



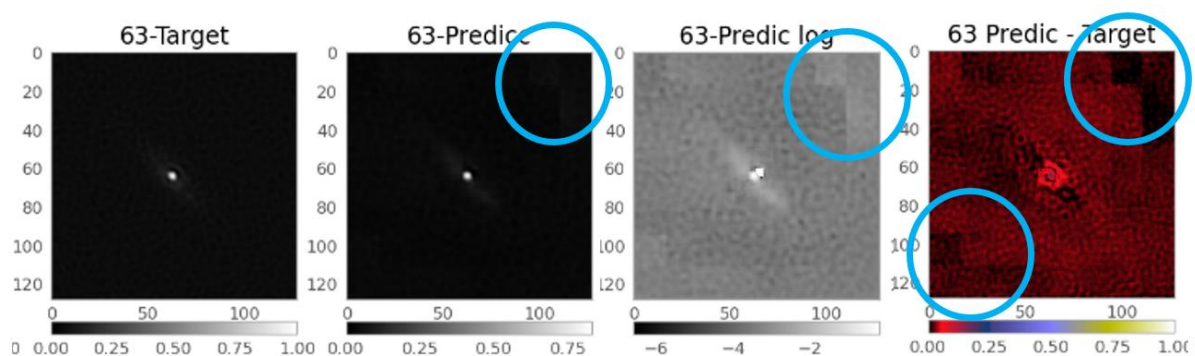
En 21.b) la reconstrucción es peor al quedar un residuo con más información. En 21.c) la estructura de la galaxia se ve afectada negativamente por la predicción, algo que se aprecia a simple vista observando ambas imágenes y se ratifica con el valor residual (Figura 22).

Figura 22 - Reconstrucción deficiente de la estructura de la galaxia



En 21.d) se observa una anomalía en la reconstrucción que se ha denominado “efecto cuadrícula”. Como puede verse ampliado en la Figura 23, la predicción genera una cuadrícula por toda la imagen, que se aprecia mejor en la escala logarítmica y en la imagen residual. Este efecto ocurre en gran parte de las predicciones, aunque suele ser muy atenuado y solo haciendo una ampliación abultada puede llegar a apreciarse. Esta cuadrícula la forman elementos de 16x16 píxeles, en total 8x8 mini bloques. Se da la circunstancia que la arquitectura emplea *kernel* paulatinamente hasta llegar a un nivel mínimo de imagen de 16x16, que parece ser el motivo de esta irregularidad.

Figura 23 - Efecto cuadrícula debido a una anomalía en la predicción.



Este comportamiento fruto de la arquitectura de la red neuronal hace que los datos relativamente buenos de las métricas se vean ensombrecidos, llevando a la conclusión de que este modelo no es adecuado al contexto de trabajo.

6.3. Evaluación de la arquitectura U-Net

Se han realizado 35 pruebas con las distintas variaciones de la arquitectura inicial según se ha explicado en el apartado 5.2.2. Los 6 modelos se han entrenado con la configuración de hiperparámetros común, pero eligiendo determinados valores en función de los resultados preliminares, ya que en algunas ocasiones el modelo no convergía y se ha tenido que variar el plan de entrenamiento. Por lo general, a mayor complejidad de la red neuronal, menor debe de ser la *learning rate* para conseguir que el entrenamiento se realizase sin errores. También se ha observado que en los modelos más complejos (Unet32, Unet64 y Unet128), entrenados en *Google Colab*, la ejecución de 200 *epoch* hacía caer al modelo en *overfitting*, por lo que las pruebas se han realizado solamente con 100 *epoch*.

De forma general los resultados son buenos, superándose en casi todos los casos la cifra de 0.95 para el SSIM, llegando a 0.96 en numerosas ocasiones y alcanzando 4 veces un valor de 0.97, en las que además se obtienen cifras entre 30 y 33 para la PSNR. En términos absolutos el mejor modelo se corresponde a Unet-64 con *batch size* igual a 32, cuyos resultados se muestran en la tabla 6.

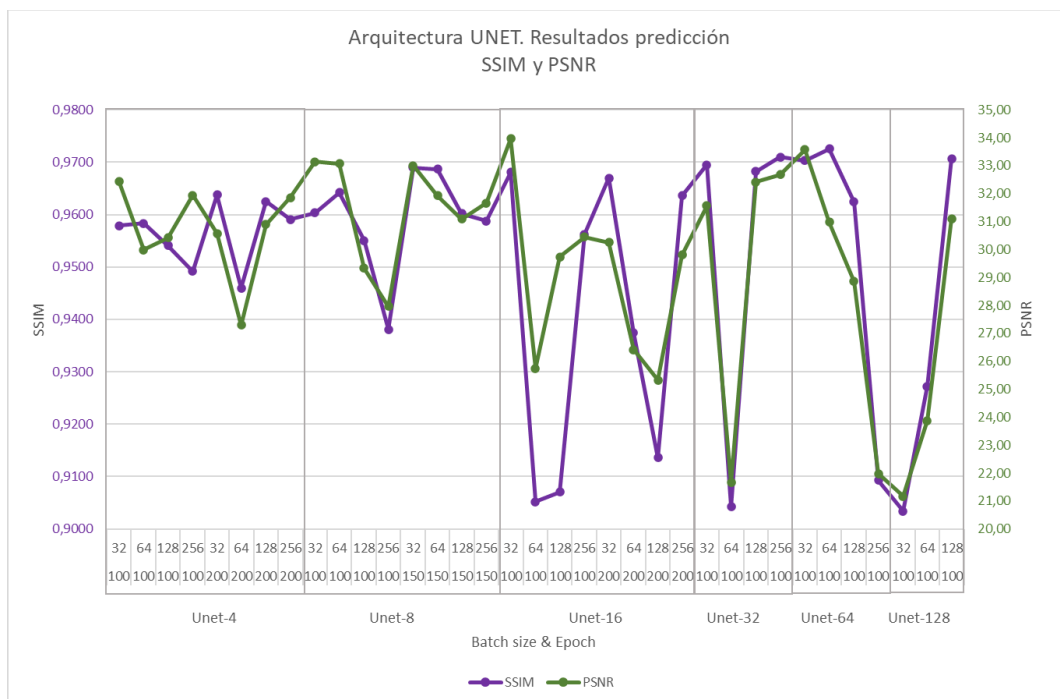
Tabla 6 - Resultados del modelo con mejores métricas de la arquitectura U-Net.

Modelo UNET64. 31.031.685 parámetros					Evaluación entrena- miento			Predicciones			
MODELO	EPOCH	BATCH SIZE	OPT	LR	LOSS (1-SSIM)	ME- TRIC (MAE)	T (seg)	MSE	RMSE	PSNR	SSIM
UNET64	100	32	Adam	1.0E-4	0,0369	0,0137	1666	28,5203	5,3404	33,5793	0,9703

En el Anexo A están reflejados los resultados de todas las pruebas realizadas.

La flexibilidad de esta arquitectura ha permitido hacer modificaciones de forma sencilla, obteniéndose modelos desde 121.000 parámetros (Unet-4) hasta más de 124 millones (Unet-128). Esta complejidad creciente, con el correspondiente coste computacional y en tiempo de entrenamiento, no refleja un beneficio acorde a este aumento, como se muestra en la Figura 24.

Figura 24 - Resultados de las predicciones con la arquitectura U-Net,



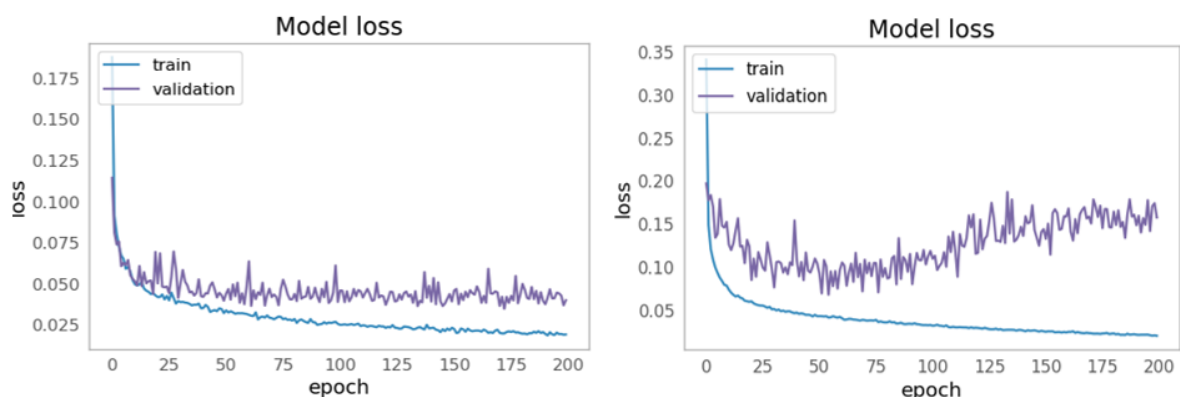
Existe una tendencia a obtener mejores resultados conforme se aumenta la complejidad de la arquitectura, pero, aunque el valor de SSIM es muy sensible a pequeños cambios y el objetivo de las pruebas es maximizarlo, el incremento del coste de entrenamiento asociado es significativamente superior a la ganancia en SSIM y PSNR.

Por otra parte, los modelos sencillos ofrecen resultados más homogéneos, mientras que los complejos obtienen métricas muy buenas, pero también inferiores incluso a los sencillos.

La evaluación en conjunto de los resultados y, sobre todo, la calidad en la reconstrucción de las imágenes de galaxias hace que el uso de las arquitecturas de elevada complejidad sea recomendable, con la salvedad de ser necesaria una buena parametrización para seleccionar la mejor opción.

En relación con la convergencia en el entrenamiento de los modelos, una característica de esta arquitectura es que dicha convergencia se alcanza de forma rápida. En la mayoría de los casos con poco más de 30 *epoch* se consiguen valores aceptables cercanos al óptimo, que ya son mejorados muy poco a poco. Un ejemplo se puede observar en la Figura 25.

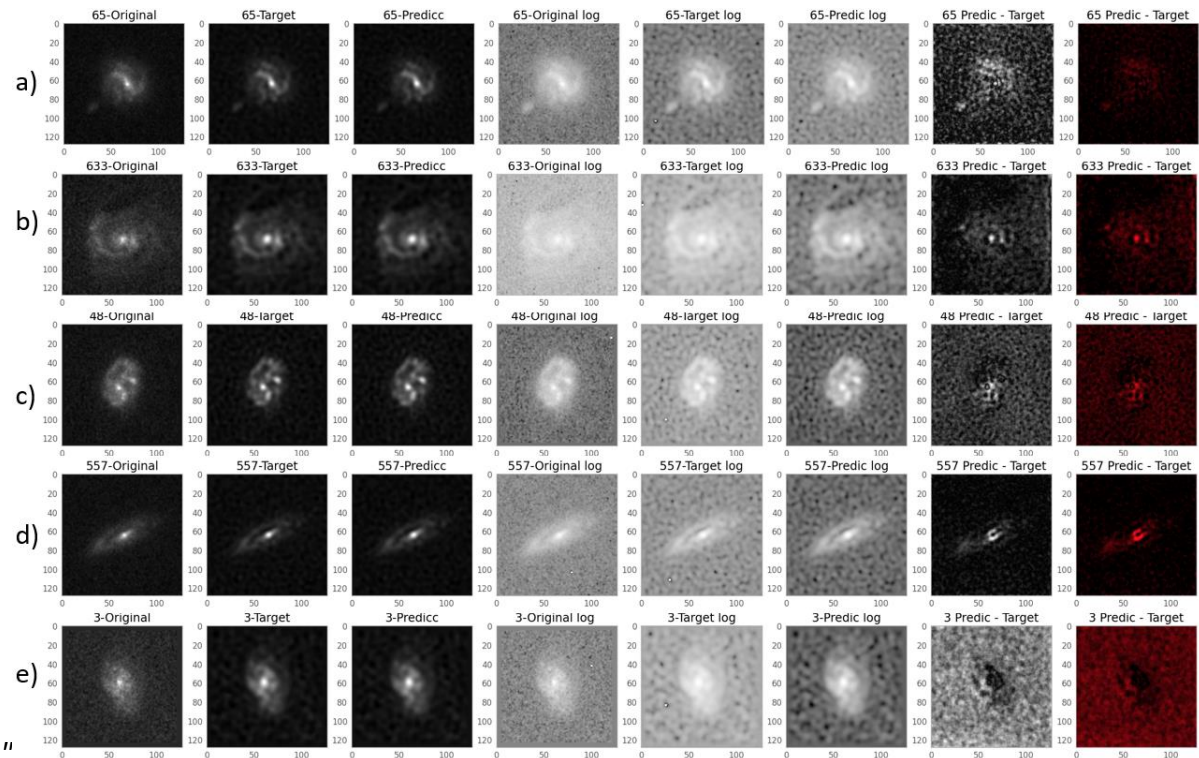
Figura 25 – Arquitectura U-Net. Gráficas de entrenamiento. A la izquierda ejemplo de convergencia rápida. A la derecha, ocurrencia de *overfitting*.



En algunos entrenamientos se ha producido *overfitting*, como el mostrado en la Figura 25. Esta circunstancia ha tenido lugar en todos los tipos de modelos independientemente de su complejidad, pero no siempre tan acusado como el ejemplo mostrado y sin ser la aparición del *overfitting* un comportamiento generalizado.

La evaluación visual de los resultados ofrece un balance positivo. En general las reconstrucciones realizadas por las distintas configuraciones de la arquitectura U-Net son buenas. En la figura 26 se muestran algunos ejemplos.

Figura 26 – Arquitectura U-Net. Ejemplos de predicciones. De izquierda a derecha, imagen original, target y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y target, en escala de grises y en mapa de color "gist_stern"



En 26.a) se consigue una reconstrucción óptima. En 26.b) y c) las diferencias son mínimas, respetándose la estructura de las galaxias. En 26.d) el proceso no es tan perfecto porque el cuerpo de la galaxia contiene algunas diferencias que se denotan por la imagen residual (diferencia entre la predicción y el *target*). En 26.e), aunque aparentemente el resultado es peor, la estructura de la galaxia se reconstruye bastante bien, mostrando la imagen residual tonos negros en el centro.

Comparando con la arquitectura AEPP2, con U-Net no se aprecia en ningún momento la ocurrencia del denominado "efecto cuadrícula". La ampliación de las imágenes no muestra patrones cuadrados ni de ninguna otra forma regular.

6.4. Evaluación de la arquitectura DIDN

Las pruebas con esta arquitectura se han realizado con los 2 modelos indicados en el apartado 5.2.3 y la combinación de hiperparámetros comúnmente establecida.

Los resultados no alcanzan una calidad aceptable, pues en ningún caso se supera el valor de 0.93 para SSIM. El modelo con 2 bloques de sub procesamiento DUB se comporta ligeramente mejor, siendo la prueba con *batch size* igual a 32 y entrenada con 150 *epoch* la que mejores resultados obtiene.

Tabla 7 - Resultados del modelo con mejores métricas de la arquitectura DIDN.

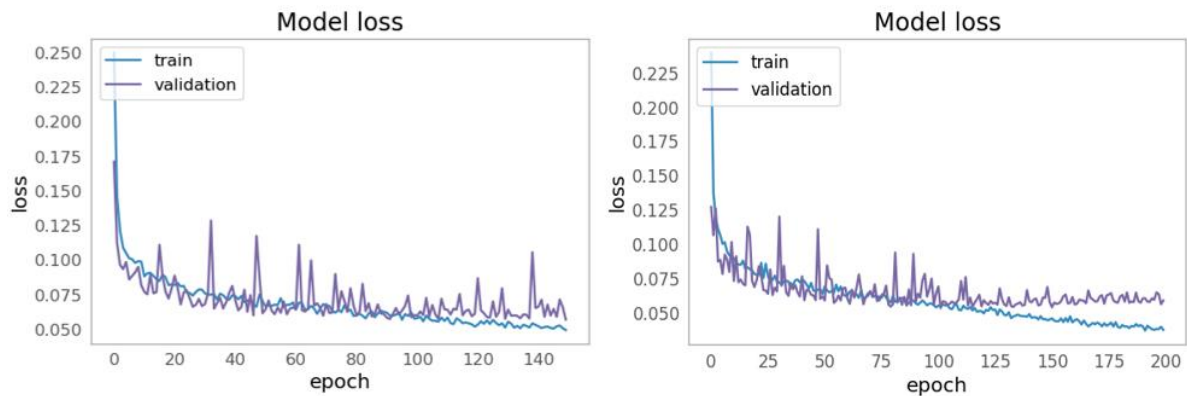
Modelo DIDN DUB2. 2.005.792 parámetros					Evaluación entrenamiento			Predicciones			
MODELO	EPOCH	BATCH SIZE	OPT	LR	LOSS (1-SSIM)	ME-TRIC (MAE)	T (seg)	MSE	RMSE	PSNR	SSIM
DIDN16DUB2	150	32	Adam	1.0E-3	0,0576	0,0170	2361	49,4050	7,0289	31,1931	0,9320

En el Anexo A están reflejados los resultados de todas las pruebas realizadas.

Una característica común a todos los entrenamientos es que la convergencia es algo lenta pero estable, y no se ha observado tendencia al *overfitting*. Los valores de la tasa de pérdida descienden paulatinamente hasta las 100 epoch aproximadamente, momento en el que dicha tasa deja prácticamente de mejorar.

Las pruebas iniciales con 150 *epoch* se repitieron con 200 *epoch* con objeto de conseguir alguna mejoría, algo que finalmente no se produjo y, por el contrario, provocó un ligero empeoramiento de la tasa de pérdida, como se aprecia en la Figura 27.

Figura 27 – Arquitectura DIDN. Gráficas de entrenamiento. A la izquierda convergencia lenta hasta 100 epoch. A la derecha la misma prueba con 200 epoch con la tasa de pérdida estabilizada.

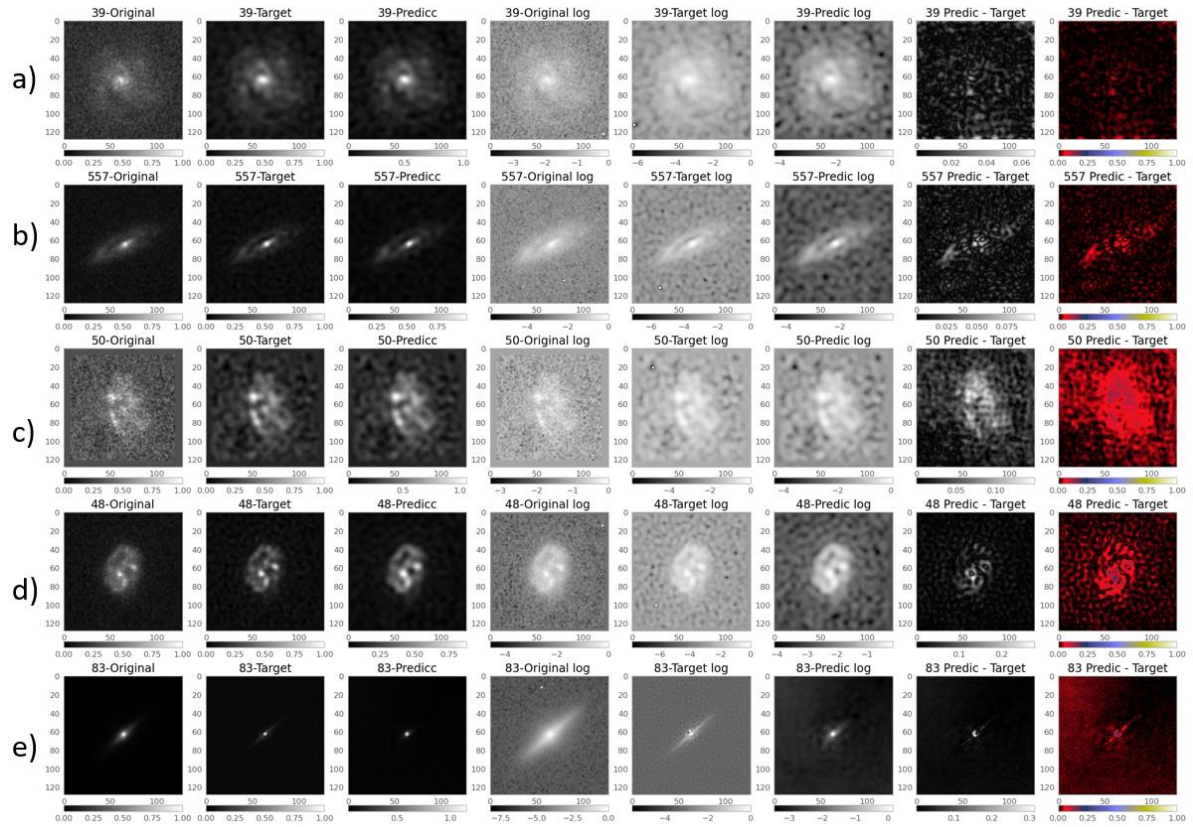


También se testaron valores para la *learning rate* distintos a $1.0E-3$ sin obtener mejoras en los resultados, por lo que las pruebas se ejecutaron finalmente con esta tasa de aprendizaje.

El análisis visual de las imágenes predichas por la arquitectura corrobora los valores obtenidos para SSIM y PSNR haciendo que las imágenes reconstruidas tengan de forma mayoritaria defectos en mayor o menor medida. Aun así, en algunas ocasiones estas predicciones son satisfactorias. En la Figura 28 se muestran algunos resultados.

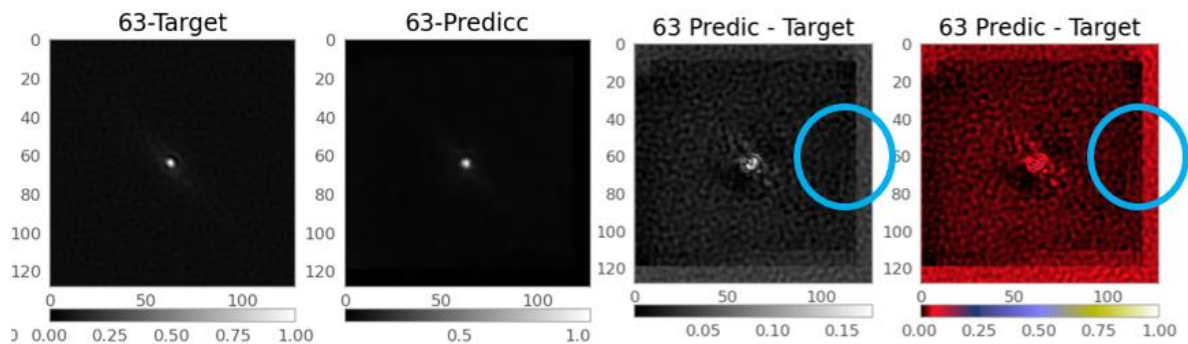
En la imagen 28.a) la predicción es óptima. En 28.b) la imagen residual (diferencia entre predicción y *target*) indica que la reconstrucción falla en la estructura de la galaxia. Este comportamiento se manifiesta a simple vista en 28.d). En 28.c) el valor residual indica bastante diferencias que afectan a toda la imagen. En 28.e) aparentemente no hay demasiado valor residual, sin embargo, el halo de la galaxia se ve afectado y se pierde de forma sustancial.

Figura 28 – Arquitectura DIDN. Ejemplos de predicciones. De izquierda a derecha, imagen original, target y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y target, en escala de grises y en mapa de color "gist_stern".



Para este modelo también se ha constatado la aparición del “efecto cuadrícula” observado con la arquitectura AEPP2. Como muestra la Figura 29, esta anomalía, difícil de ver en la imagen predicha, aparece claramente al calcular la imagen residual.

Figura 29 - Efecto cuadrícula debido a una anomalía en la predicción.



Este comportamiento no es muy frecuente y ha aparecido solo en el modelo con 1 bloque DUP. Los entrenamientos con el modelo de 2 DUB no muestra esta particularidad. Además, el efecto no es exactamente el mismo, porque en esta ocasión, la anomalía no la forman cuadrados claramente delimitados, sino una o dos franjas a modo de marco de la imagen.

6.5. Discusión

La arquitectura U-Net se alza como la mejor propuesta en esta comparativa. Los resultados cuantitativos obtenidos de las métricas SSIM y PSNR para algunas de sus configuraciones son los mejores, logrando además una cifra muy aceptable considerando la naturaleza de estas medidas de evaluación. El objetivo perseguido de deconvolucionar imágenes de galaxias capturadas con el telescopio espacial Hubble es posible con este modelo de red neuronal.

Hay que resaltar que la arquitectura U-Net respeta enormemente la estructura y fisionomía de las imágenes, en este caso galaxias, no introduciendo anomalías ni deformaciones que suponga un perjuicio desde el punto de vista científico. Los resultados en las predicciones que en ocasiones no son satisfactorias implican una merma en la calidad de la reconstrucción, pero no introducen alteraciones como el “efecto cuadrícula” observado con la red DIDN y en mayor medida con la red AEPP2.

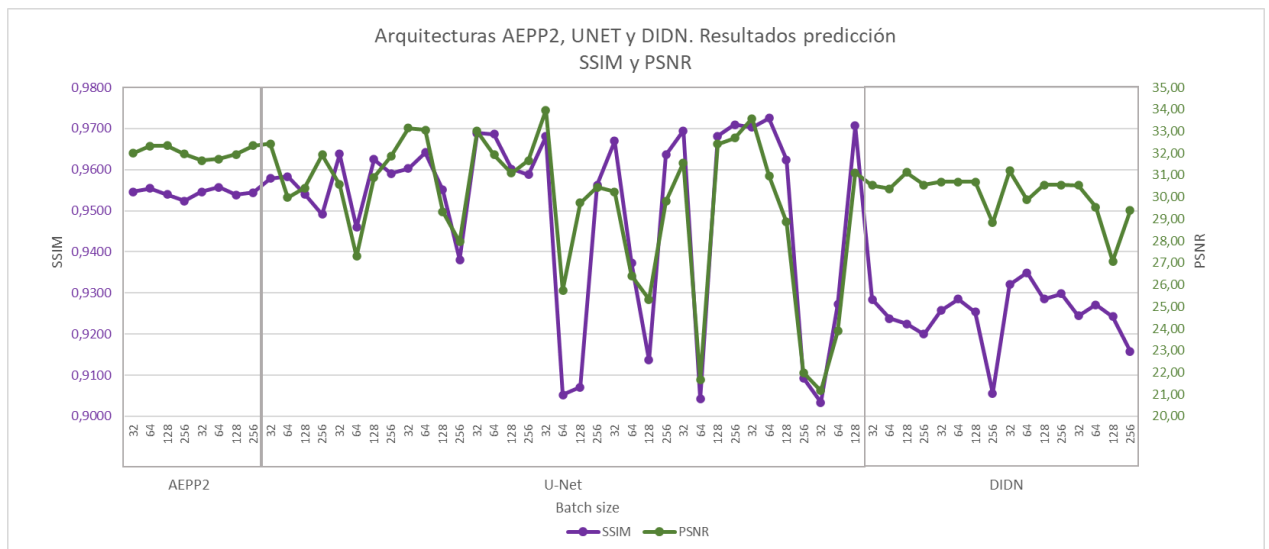
La gran flexibilidad y fácil escalabilidad de la red U-Net hace posible la experimentación y la ejecución de entrenamientos que permitan determinar la mejor parametrización para maximizar los mejores resultados. Todo ello contribuye a contrarrestar la relativa variabilidad de los modelos, que como se ha explicado, hace que algunas pruebas no ofrezcan resultados satisfactorios. También es destacable la rápida convergencia que presentan este modelo de red, que implica por otra parte la vigilancia para evitar el *overfitting*.

La arquitectura AEPP2, aunque ofrece resultados satisfactorios, no llega al nivel de calidad de la U-Net. Como se ha descrito, en ocasiones la reconstrucción de las imágenes no es completamente satisfactoria, incluso a simple vista. Además, la aparición de la anomalía descrita como “efecto cuadrícula” supone un importante problema imposible de ignorar. Con todo ello se puede afirmar que el uso de la arquitectura ARPP2 no es válido para la deconvolución de imágenes de galaxias.

Respecto al modelo DIDN, los resultados son los peores de las tres alternativas, obteniendo imágenes generalmente mal reconstruidas. La pequeña mejora que supone utilizar dos unidades de subprocesamiento hace pensar que la inclusión de un mayor número de ellas permita mejorar estos resultados, sin embargo, y como se ha explicado, las numerosas pruebas realizadas no han conseguido converger satisfactoriamente, por lo que esta posible mejora no se ha podido constatar. Otro aspecto que la red DIDN ha puesto de manifiesto es su lenta convergencia, haciendo necesaria la ejecución de un gran número de ciclos o *epoch* para obtener métricas reducidas. La característica positiva de este modelo es su buen comportamiento frente al *overfitting*, el mejor de las tres arquitecturas probadas.

En la Figura 30 se muestran los valores de SSIM y PSNR de todas las pruebas realizadas.

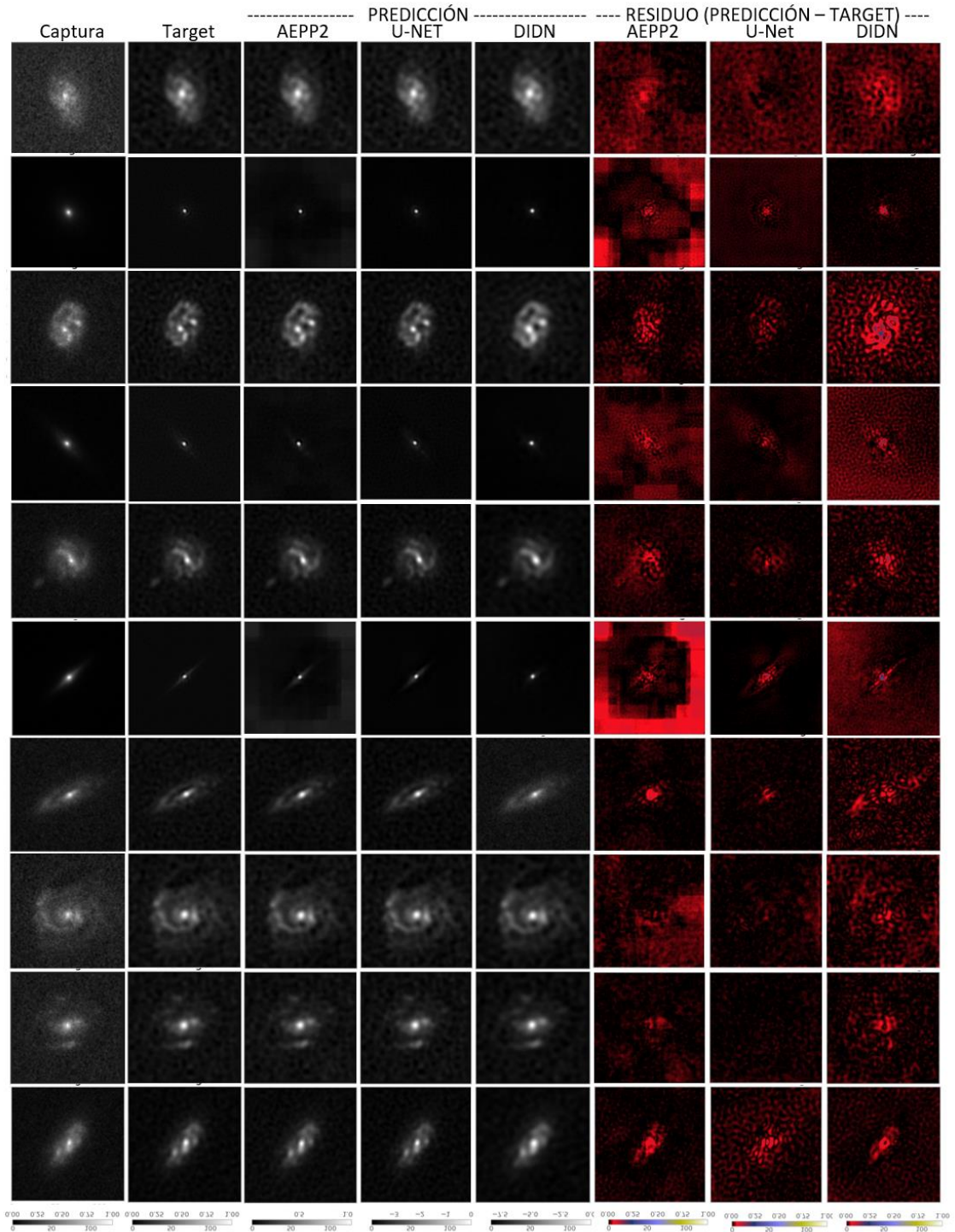
Figura 30 - Resultados de las predicciones de las tres arquitecturas.



En la Figura 31 se muestran comparativamente los resultados de varios ejemplos con cada modelo de red neuronal.

Figura 31 - Ejemplos de predicciones realizadas con los mejores modelos de cada arquitectura.

De izquierda a derecha, imagen capturada, target, predicción con AEPP2, Unet, DIDN e imagen residual en AEPP2, Unet y DIDN.



7. Conclusiones y trabajo futuro

7.1. Conclusiones

La problemática de las aberraciones producidas en la captura de imágenes astronómicas ha motivado el desarrollo de técnicas que permitan su eliminación y ayuden a los astrónomos en sus estudios científicos.

La reconstrucción de estas imágenes con algoritmos matemáticos ha dado paso a nuevas técnicas basadas en inteligencia artificial, que están demostrando su validez como alternativa.

El uso de aprendizaje automático supervisado con redes neuronales para la deconvolución de imágenes astronómicas supone una importante alternativa frente a los procedimientos usados habitualmente. La posibilidad de reconstruir imágenes a cómo serían realmente sin disponer de la función de dispersión de punto, es similar al proceso de deconvolución ciega, con la ventaja de haber aprendido a realizar este proceso con datos validados previamente.

En este proyecto se han usado capturas del telescopio espacial Hubble con las que se han entrenado distintas configuraciones de tres arquitecturas de redes neuronales. El modelo UNet se posiciona como una opción robusta para la deconvolución eficaz de las imágenes astronómicas. El proceso de compresión y descompresión extrayendo características significativas, a la vez que se restauran las imperfecciones, ha demostrado un buen rendimiento, que, unido a las particularidades de la arquitectura, fundamentalmente la escalabilidad y rápida convergencia, hacen que sirva de gran ayuda a la comunidad científica para el estudio de las imágenes de galaxias libres de aberraciones.

7.2. Líneas de trabajo futuro

Como propuestas de mejora se plantean las siguientes líneas de trabajo:

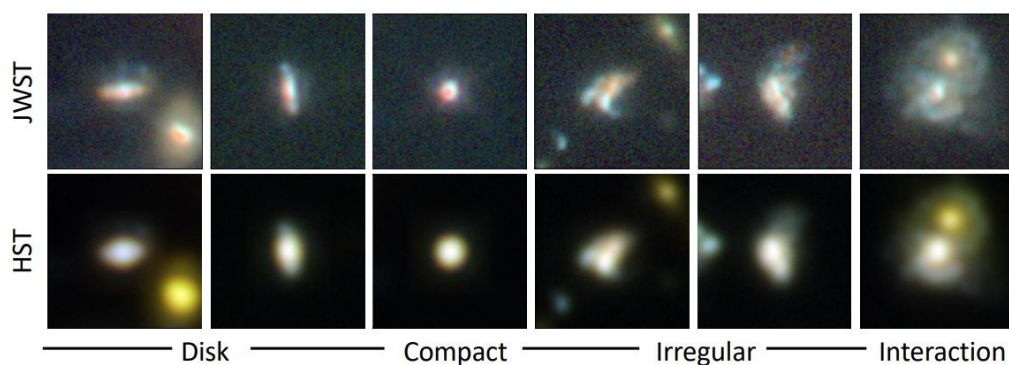
1.- Aumento del tamaño de dataset de entrenamiento: Una de las limitaciones que se ha tenido es el número de imágenes con las que entrenar. Aunque el *dataset* origen contiene más de 118.000 capturas, las utilizadas en las pruebas se han limitado a 9.625, que, sin ser una cifra pequeña, sí que sería mejorable para conseguir un entrenamiento más variado y efectivo. El motivo de esta limitación ha sido la elección de un tamaño de imagen de 128x128

que se ha considerado visualmente representativo, y compatible con las arquitecturas construidas. Sin embargo, el *dataset* de partida contiene mayoritariamente imágenes más pequeñas, que podrían ser igualmente utilizadas adaptando la red. Lo ideal sería poder usar el mayor número de imágenes de galaxias, escalándolas a un tamaño fijo superior, procesarlas con la red neuronal y reduciéndolas de nuevo. La clave sería estudiar en qué afecta esta operativa al uso de la PSF para la deconvolución de la imagen *target*.

2.- Uso de imágenes reales como *target*: Como se ha explicado previamente, las imágenes de trabajo se han obtenido de la campaña de observación astronómica COSMOS, *The Cosmic Evolution Survey*. Esta misma campaña se ha planificado para hacerse con el nuevo telescopio espacial James Webb, en el estudio COSMOS-Web, que va a realizar capturas del mismo sector del firmamento. Este ciclo de capturas ya ha comenzado, pero se completará durante los años 2023 y 2024. Una vez finalizado, puesto que las galaxias capturadas serán las mismas y están identificadas y catalogadas, se podría construir un dataset de entrenamiento formado con imágenes del Hubble e imágenes del Webb que serían el *target* al tener una mayor calidad y resolución. De esta forma las predicciones de nuevas capturas se basarían en un modelo entrenado con información real de mayor calidad.

Con las primeras observaciones del estudio COSMOS-Web ya se han empezado a realizar comparativas usando las capturas de ambos telescopios espaciales, como se hace en (Casey et al., 2022)

Figura 32 – Imágenes de las mismas galaxias capturadas con el telescopio espacial James Webb (arriba) y el telescopio espacial Hubble (abajo) (Casey et al., 2022)



3.- Uso de redes GAN (*Generative Adversarial Network*). Este tipo de redes neuronales implican un planteamiento distinto para el entrenamiento, generando aproximaciones de la imagen objetivo hasta que esta es considerada válida. Para este caso de uso, al tratarse de un aprendizaje no supervisado, la reconstrucción se realizaría sin ningún tipo de información adicional a la imagen capturada, lo cual evita tener que deconvolucionar previamente para obtener el *target*. Los resultados obtenidos por este tipo de arquitectura se están mostrando sorprendentes y útiles en las áreas donde se experimenta con ellas, y las hace merecedoras de realizar pruebas en la reconstrucción de imágenes astronómicas.

REFERENCIAS BIBLIOGRÁFICAS

- Akhaury, U., Starck, J.-L., Jablonka, P., Courbin, F., & Michalewicz, K. (Eds.). (2022). Deep learning-based galaxy image deconvolution. *Frontiers In Astronomy And Space Sciences*, 9. <https://doi.org/10.3389/fspas.2022.1001043>
- Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein Generative Adversarial Networks. *Proceedings of the 34th International Conference on Machine Learning*, 214-223.
- Casey, C. M., Kartaltepe, J. S., Drakos, N. E., Franco, M., Ilbert, O., Rose, C., Cox, I. G., Nightingale, J. W., Robertson, B. E., Silverman, J. D., Koekemoer, A. M., Massey, R., McCracken, H. J., Rhodes, J., Akins, H. B., Amvrosiadis, A., Arango-Toro, R. C., Bagley, M. B., Capak, P. L., ... Zavala, J. A. (2022). COSMOS-Web: An Overview of the JWST Cosmic Origins Survey. *arXiv*, arXiv:2211.07865.
- Chiappetti, L., Currie, M. J., Allen, S., Dobrzycki, A., Pence, W. D., Rots, A., Shaw, R., & Thompson, W. D. (2018). *Definition of the Flexible Image Transport System (FITS) The FITS Standard Version 4.0: Updated 2016 July 22 by the IAUFWG Original document publication date: 2016 July 22 Language-edited document publication date: 2018 August 13* [Working paper]. <https://openaccess.inaf.it/handle/20.500.12386/27619>
- Gan, F. K., Bekki, K., & Hashemizadeh, A. (2021). *SeeingGAN: Galactic image deblurring with deep learning for better morphological classification of galaxies* (arXiv:2103.09711). arXiv. <https://doi.org/10.48550/arXiv.2103.09711>
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139-144. <https://doi.org/10.1145/3422622>
- Hasinoff, S. W. (2014). Photon , Poisson noise. *Computer Vision, A Reference Guide*, 4.
- Hernández Afonso, J. (2022). *Redes Neuronales Convolucionales para la Reconstrucción de Imágenes de Galaxias* [TFM]. Universidad Internacional de La Rioja.

- Huertas-Company, M., & Lanusse, F. (2023). The Dawes Review 10: The impact of deep learning for the analysis of galaxy surveys. *Publications of the Astronomical Society of Australia*, 40, e001. <https://doi.org/10.1017/pasa.2022.55>
- Jefferies, S. M., Christou, J. C., Jefferies, S. M., & Christou, J. C. (1993). Restoration of Astronomical Images by Iterative Blind Deconvolution. *ApJ*, 415, 862. <https://doi.org/10.1086/173208>
- Koekemoer, A. M., Aussel, H., Calzetti, D., Capak, P., Giavalisco, M., Kneib, J.-P., Leauthaud, A., Fèvre, O. L., McCracken, H. J., Massey, R., Mobasher, B., Rhodes, J., Scoville, N., & Shopbell, P. L. (2007). The COSMOS Survey: Hubble Space Telescope Advanced Camera for Surveys Observations and Data Processing. *The Astrophysical Journal Supplement Series*, 172(1), 196. <https://doi.org/10.1086/520086>
- Krist, J. E., Hook, R. N., & Stoehr, F. (2011). 20 years of Hubble Space Telescope optical modeling using Tiny Tim. *Optical Modeling and Performance Predictions V*, 8127, 166-181. <https://doi.org/10.1117/12.892762>
- Lee, H. Y., Kwak, J. M., Ban, B., Na, S. J., Lee, S. R., & Lee, H.-K. (2017). GAN-D: Generative adversarial networks for image deconvolution. *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, 132-137. <https://doi.org/10.1109/ICTC.2017.8190958>
- Liu, P.-Y., & Lam, E. Y. (2018). *Image Reconstruction Using Deep Learning* (arXiv:1809.10410). arXiv. <https://doi.org/10.48550/arXiv.1809.10410>
- Lucy, L. B. (1974). An iterative technique for the rectification of observed distributions. *The Astronomical Journal*, 79(6).
- Makarkin, M., & Bratashov, D. (2021). State-of-the-Art Approaches for Image Deconvolution Problems, including Modern Deep Learning Architectures. *Micromachines*, 12(12), 1558. <https://doi.org/10.3390/mi12121558>
- Mandelbaum, R., Lackner, C., Leauthaud, A., & Rowe, B. (2012). *COSMOS real galaxy dataset* [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.3242143>

- Mao, X., Shen, C., & Yang, Y.-B. (2016). Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections. *Advances in Neural Information Processing Systems*, 29. <https://proceedings.neurips.cc/paper/2016/hash/0ed9422357395a0d4879191c66f4faa2-Abstract.html>
- Molina, R., Nunez, J., Cortijo, F. J., & Mateos, J. (2001). Image restoration in astronomy: A Bayesian perspective. *IEEE Signal Processing Magazine*, 18(2), 11-29. <https://doi.org/10.1109/79.916318>
- Nammour, F., Akhaury, U., Girard, J. N., Lanusse, F., Sureau, F., Ali, C. B., & Starck, J.-L. (2022). ShapeNet: Shape constraint for galaxy image deconvolution. *Astronomy & Astrophysics*, 663, A69. <https://doi.org/10.1051/0004-6361/202142626>
- Orieux, F., Giovannelli, J.-F., & Rodet, T. (2010). Bayesian estimation of regularization and point spread function parameters for Wiener–Hunt deconvolution. *Journal of the Optical Society of America A*, 27(7), 1593-1607. <https://doi.org/10.1364/JOSAA.27.001593>
- O’Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. En *ArXiv e-prints*. <https://doi.org/10.48550/arXiv.1511.08458>
- Ramzi, Z., Michalewicz, K., Starck, J.-L., Moreau, T., & Ciuciu, P. (2023). Wavelets in the Deep Learning Era. *Journal of Mathematical Imaging and Vision*, 65(1), 240-251. <https://doi.org/10.1007/s10851-022-01123-w>
- Richardson, W. H. (1972). Bayesian-Based Iterative Method of Image Restoration. *Journal of the Optical Society of America*, 62(1), 55-59. <https://doi.org/10.1364/JOSA.62.000055>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. En N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (pp. 234-241). Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28

- Roy, S. (2022). *U Net Lowered with TF-Keras* [Python]. <https://github.com/sagnik1511/U-Net-Reduced-with-TF-keras> (Original work published 2021)
- Ryon, J. E., Avila, R., Ubeda, L., Maybhate, A., Armstrong, A., Boffi, F. R., Pavlovsky, C., Gonzaga, S., Suchkov, A., & Jedrzejewski, R. (2022). *ACS Instrument Handbook Versión 21.0* (Baltimore: STScI, Ed.). <https://hst-docs.stsci.edu/acsihb>
- Sacek, V. (2006, julio 14). *Point spread function PSF*. Amateur telescope optics. https://www.telescope-optics.net/diffraction_image.htm
- Scoville, N., Aussel, H., Brusa, M., Capak, P., Carollo, C. M., Elvis, M., Giavalisco, M., Guzzo, L., Hasinger, G., Impey, C., Kneib, J.-P., LeFevre, O., Lilly, S. J., Mobasher, B., Renzini, A., Rich, R. M., Sanders, D. B., Schinnerer, E., Schminovich, D., ... Tyson, N. D. (2007). The Cosmic Evolution Survey (COSMOS): Overview. *The Astrophysical Journal Supplement Series*, 172(1), 1. <https://doi.org/10.1086/516585>
- Sparacino, G., De Nicolao, G., Pilonetto, G., & Cobelli, C. (2014). Deconvolution. En E. Carson & C. Cobelli (Eds.), *Modelling Methodology for Physiology and Medicine (Second Edition)* (pp. 45-68). Elsevier. <https://doi.org/10.1016/B978-0-12-411557-6.00003-3>
- Starck, J. L., Pantin, E., & Murtagh, F. (2002). Deconvolution in Astronomy: A Review. *Publications of the Astronomical Society of the Pacific*, 114(800), 1051. <https://doi.org/10.1086/342606>
- Sureau, F., Lechat, A., & Starck, J.-L. (2020). Deep learning for a space-variant deconvolution in galaxy surveys. *Astronomy & Astrophysics*, 641, A67. <https://doi.org/10.1051/0004-6361/201937039>
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600-612. <https://doi.org/10.1109/TIP.2003.819861>
- Yu, S. (2022). *Pytorch Implementation of «Deep Iterative Down-Up CNN for Image Denoising»*. *DIDN* [Python]. <https://github.com/SonghyunYu/DIDN> (Original work published 2019)

Yu, S., Park, B., & Jeong, J. (2019). Deep Iterative Down-Up CNN for Image Denoising. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2095-2103. <https://doi.org/10.1109/CVPRW.2019.00262>

ANEXO A. ESTADÍSTICAS DE ENTRENAMIENTOS

Entrenamiento de la arquitectura AEPP2

Tabla 8 - AEPP2. Hiperparámetros, métricas de entrenamiento y predicción.

Modelo AEPP2. 1.444.674 parámetros					Evaluación entrena- miento			Predicciones			
MODELO	EPOCH	BATCH SIZE	OPT	LR	LOSS (1-SSIM)	ME- TRIC (MAE)	T (seg)	MSE	RMSE	PSNR	SSIM
AEPP2	100	32	Adam	1.0E-3	0,0596	0,0159	952	41,1040	6,4112	31,9920	0,9546
AEPP2	100	64	Adam	1.0E-3	0,0569	0,0152	864	38,0581	6,1691	32,3263	0,9555
AEPP2	100	128	Adam	1.0E-3	0,0571	0,0154	819	37,8199	6,1498	32,3536	0,9540
AEPP2	100	256	Adam	1.0E-3	0,0576	0,0154	822	41,3706	6,4320	31,9639	0,9524
AEPP2	200	32	Adam	1.0E-3	0,0636	0,0163	1827	44,3327	6,6583	31,6636	0,9546
AEPP2	200	64	Adam	1.0E-3	0,0634	0,0160	1598	43,6169	6,6043	31,7343	0,9558
AEPP2	200	128	Adam	1.0E-3	0,0622	0,0161	1535	41,4919	6,4414	31,9512	0,9539
AEPP2	200	256	Adam	1.0E-3	0,0587	0,0156	1471	37,9209	6,1580	32,3420	0,9544

Figura 33 - AEPP. A la izquierda, resultados de entrenamiento (LOSS y MAE). A la derecha resultados de predicción con las imágenes de test (SSIM y PSNR)

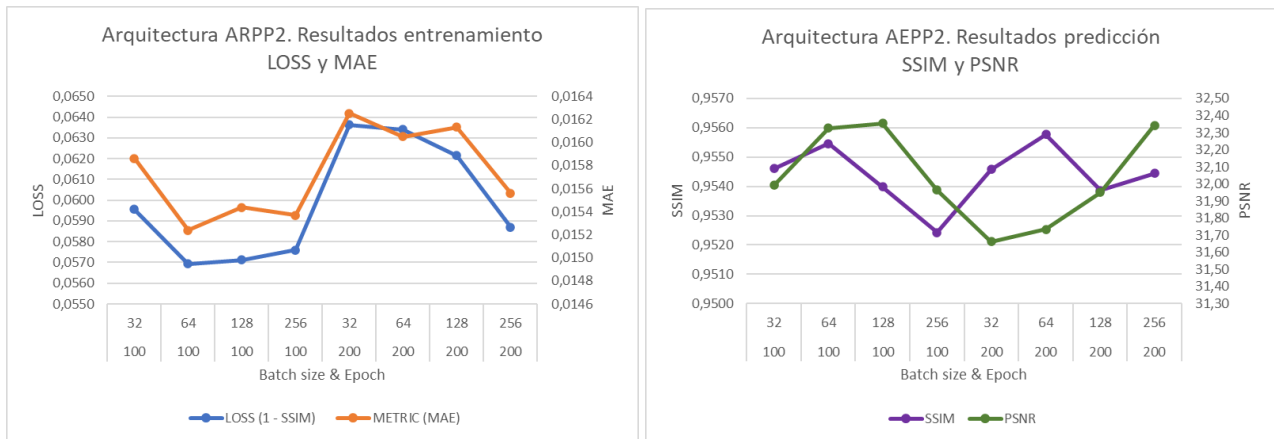


Figura 34 - AEPP2. Gráficas de entrenamiento. De arriba a abajo, valores 32, 64, 128 y 256 batch size. Columna izquierda, valores para 100 epoch. Columna derecha, valores para 200 epoch.

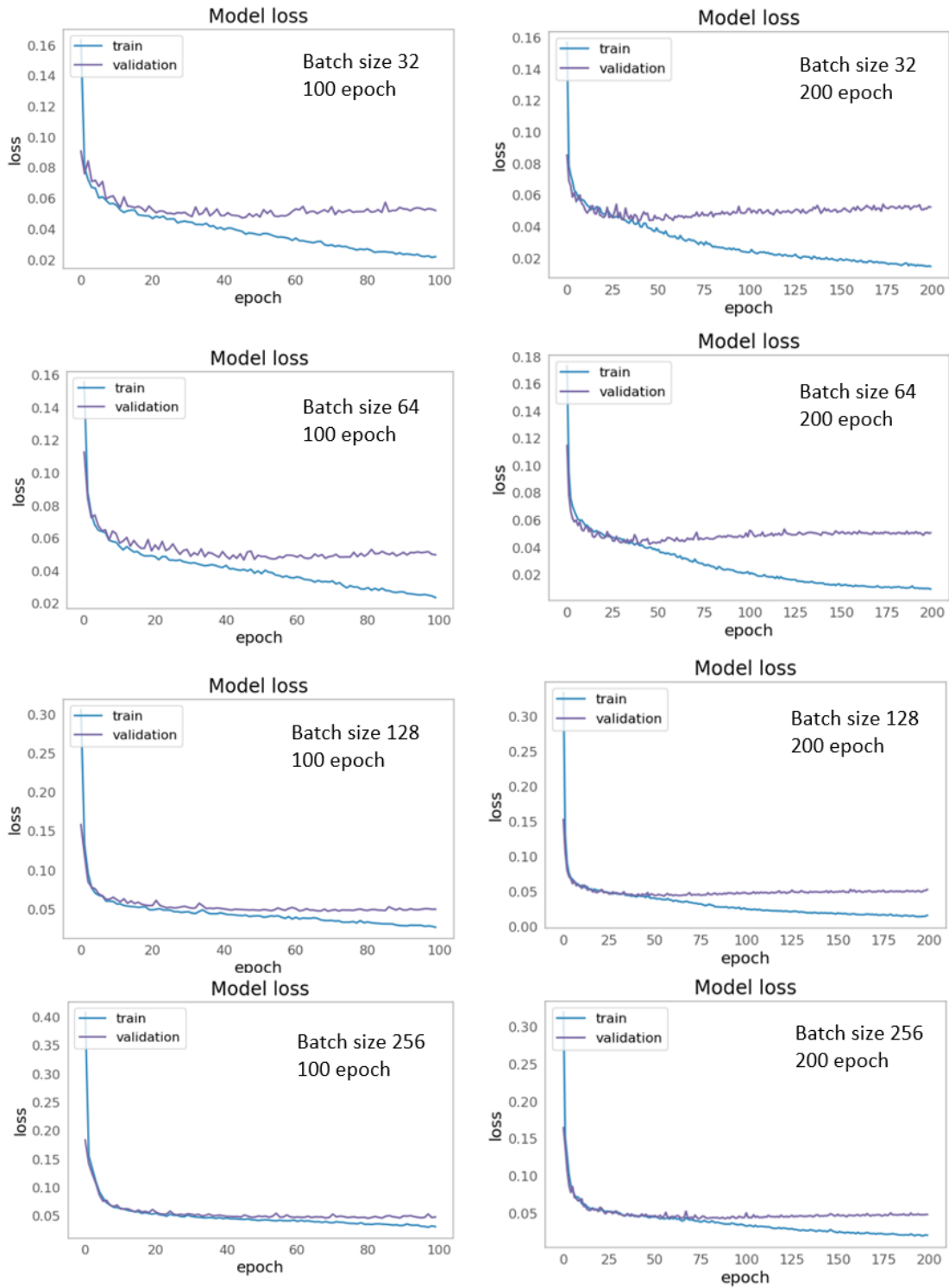
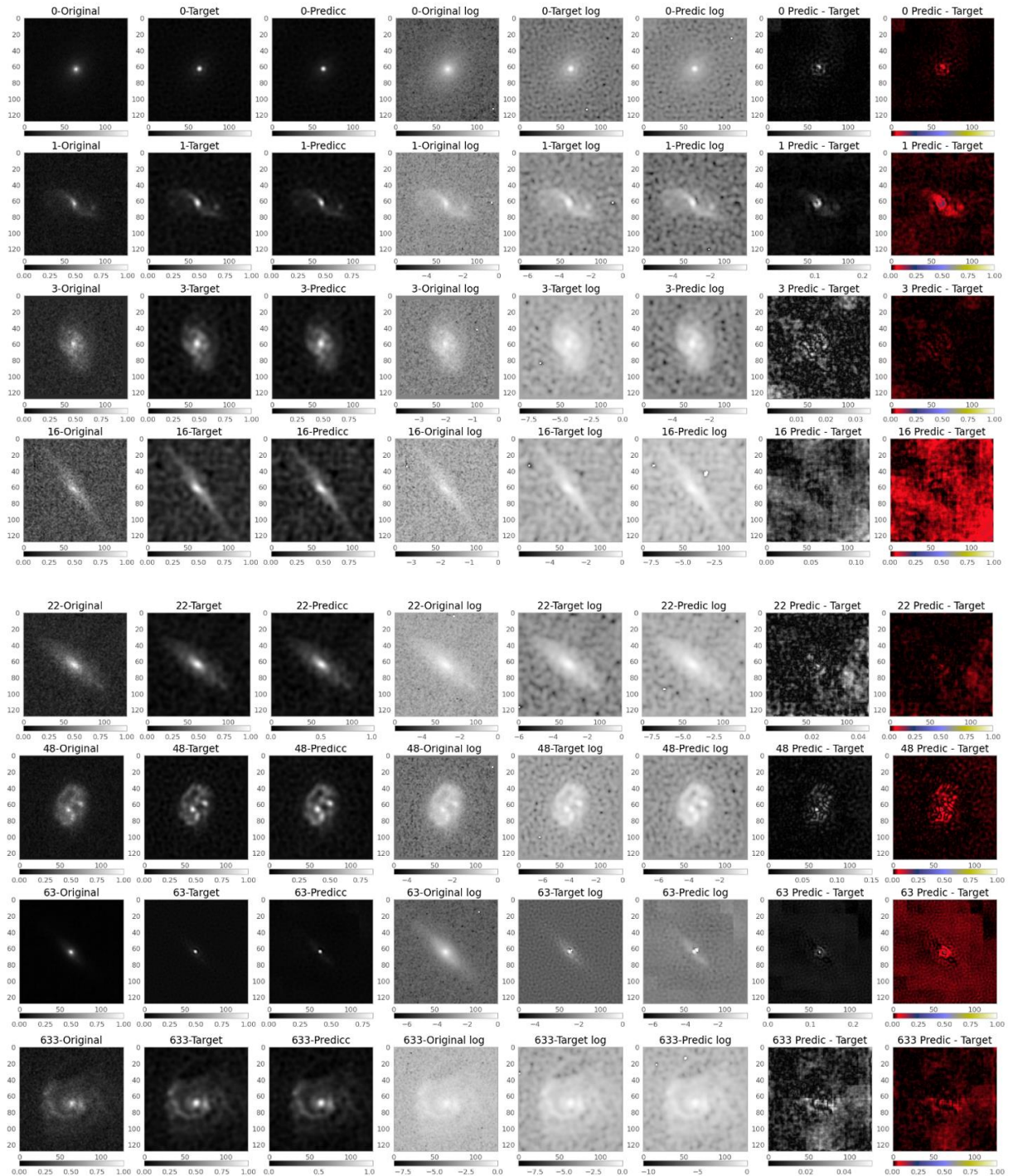
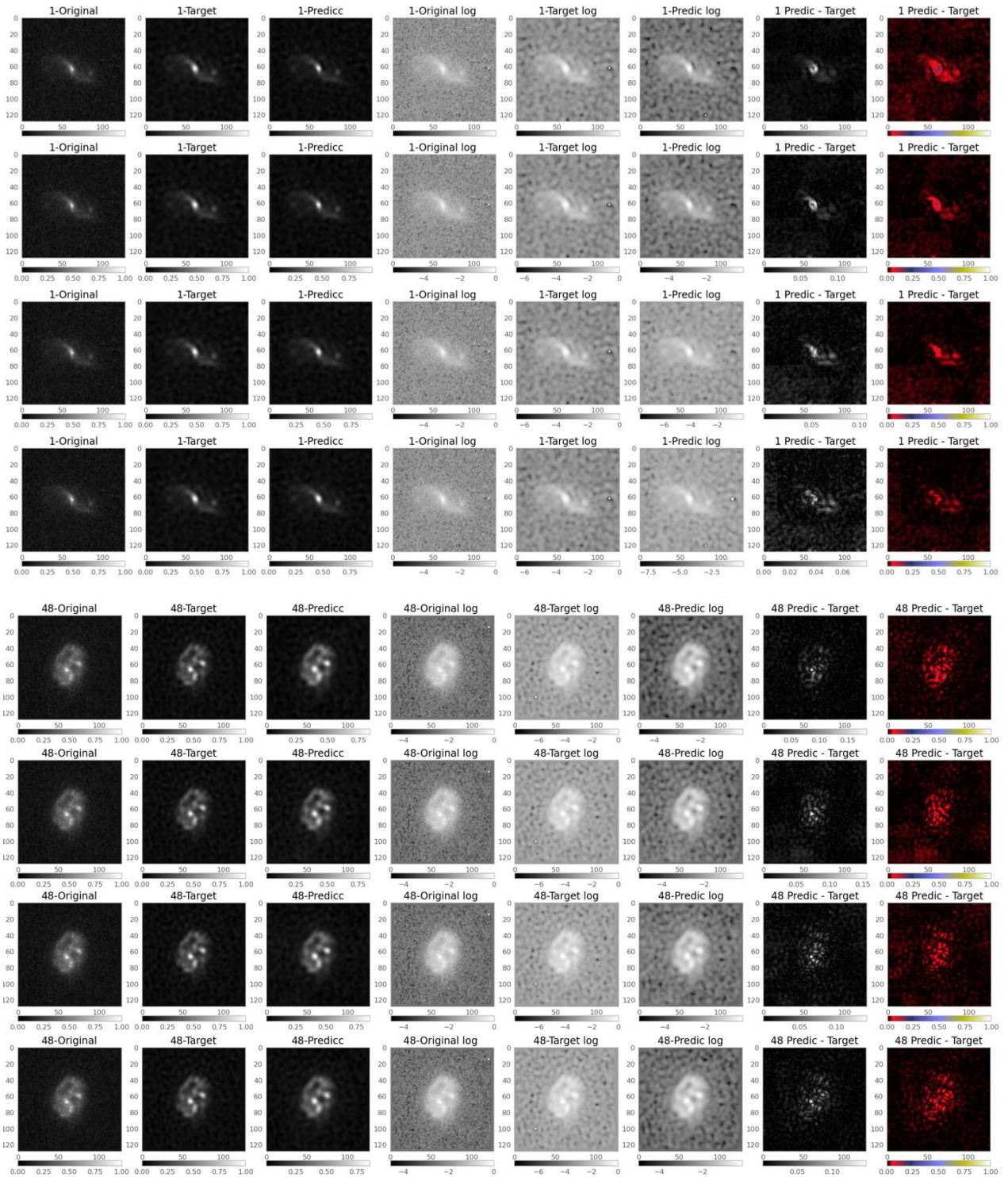


Figura 35 - AEPP2. Muestra de predicciones. De izquierda a derecha, imagen original, target y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y target, en escala de grises y en mapa de color "gist_stern".



Deep Learning como alternativa a la deconvolución de imágenes de galaxias capturadas con el telescopio espacial Hubble

Figura 36 - AEPP2. Comparativa de la predicción de 2 imágenes con 4 entrenamientos distintos, de arriba abajo en cada bloque, batch size de 32, 64, 128 y 256.



Entrenamiento de la arquitectura UNET**Tabla 9 - UNET. Hiperparámetros, métricas de entrenamiento y predicción. Los modelos marcados con (*) se han entrenado en Google Colab.**

Modelos UNET					Evaluación entrenamiento			Predicciones			
MODELO	EPOCH	BATCH SIZE	OPT	LR	LOSS (1-SSIM)	METRIC (MAE)	T (seg)	MSE	RMSE	PSNR	SSIM
UNET4	100	32	Adam	1.0E-3	0,0458	0,0148	589	37,0699	6,0885	32,4406	0,9579
UNET4	100	64	Adam	1.0E-3	0,0460	0,0176	440	65,3026	8,0810	29,9815	0,9583
UNET4	100	128	Adam	1.0E-3	0,0432	0,0157	391	59,0169	7,6822	30,4210	0,9541
UNET4	100	256	Adam	1.0E-3	0,0530	0,0152	380	41,6392	6,4528	31,9358	0,9491
UNET4	200	32	Adam	1.0E-3	0,0426	0,0159	1226	57,0713	7,5546	30,5666	0,9638
UNET4	200	64	Adam	1.0E-3	0,0691	0,0275	871	121,0796	11,0036	27,3001	0,9460
UNET4	200	128	Adam	1.0E-3	0,0430	0,0146	835	52,8784	7,2718	30,8980	0,9625
UNET4	200	256	Adam	1.0E-3	0,0459	0,0147	769	42,4452	6,5150	31,8525	0,9591
UNET8	100	32	Adam	1.0E-3	0,0406	0,0145	1067	31,5035	5,6128	33,1472	0,9603
UNET8	100	64	Adam	1.0E-3	0,0409	0,0149	648	32,1079	5,6664	33,0647	0,9642
UNET8	100	128	Adam	1.0E-3	0,0632	0,0216	589	75,7362	8,7027	29,3378	0,9551
UNET8	100	256	Adam	1.0E-3	0,0674	0,0260	873	103,8578	10,1911	27,9664	0,9381
UNET8	150	32	Adam	1.0E-3	0,0383	0,0136	1195	32,5463	5,7049	33,0058	0,9689
UNET8	150	64	Adam	1.0E-3	0,0373	0,0137	962	41,7678	6,4628	31,9224	0,9686
UNET8	150	128	Adam	1.0E-3	0,0461	0,0173	948	50,6426	7,1164	31,0856	0,9602
UNET8	150	256	Adam	1.0E-3	0,0482	0,0163	967	44,3164	6,6571	31,6652	0,9588
UNET16	100	32	Adam	1.0E-3	0,0363	0,0133	1327	26,1245	5,1112	33,9603	0,9681
UNET16	100	64	Adam	1.0E-3	0,1080	0,0315	1201	173,0836	13,1561	25,7482	0,9052
UNET16	100	128	Adam	1.0E-5	0,0759	0,0205	1150	69,1311	8,3145	29,7341	0,9071
UNET16*	100	256	Adam	1.0E-3	0,0551	0,0209	387	58,7060	7,6620	30,4440	0,9563
UNET16	200	32	Adam	1.0E-4	0,0448	0,0173	2690	61,4367	7,8382	30,2465	0,9670
UNET16	200	64	Adam	1.0E-5	0,0769	0,0263	2276	148,3386	12,1794	26,4183	0,9374
UNET16	200	128	Adam	1.0E-5	0,1083	0,0334	2229	190,5927	13,8055	25,3297	0,9136
UNET16*	200	256	Adam	1.0E-3	0,0493	0,0190	773	68,0646	8,2501	29,8016	0,9636
UNET32*	100	32	Adam	1.0E-4	0,0419	0,0155	824	45,3113	6,7314	31,5687	0,9695
UNET32*	100	64	Adam	1.0E-5	0,1659	0,0505	741	443,7784	21,0660	21,6591	0,9042
UNET32*	100	128	Adam	1.0E-3	0,0380	0,0140	680	37,3153	6,1086	32,4119	0,9682
UNET32*	100	256	Adam	1.0E-3	0,0316	0,0125	674	35,0267	5,9183	32,6868	0,9710
UNET64*	100	32	Adam	1.0E-4	0,0369	0,0137	1666	28,5203	5,3404	33,5793	0,9703
UNET64*	100	64	Adam	1.0E-4	0,0372	0,0143	1505	51,9664	7,2088	30,9736	0,9726
UNET64*	100	128	Adam	1.0E-4	0,0546	0,0213	1391	84,5257	9,1938	28,8609	0,9624
UNET64*	100	256	Adam	1.0E-5	0,1546	0,0460	1431	412,0526	20,2988	21,9814	0,9093
UNET128*	100	32	Adam	1.0E-6	0,1748	0,0521	4162	497,0567	22,2948	21,1667	0,9034
UNET128*	100	64	Adam	1.0E-5	0,1130	0,0409	3831	266,2644	16,3176	23,8777	0,9272
UNET128*	100	128	Adam	1.0E-4	0,0359	0,0147	3638	50,3681	7,0971	31,1092	0,9707

Figura 37 - UNET. Resultados de entrenamiento (LOSS y MAE).

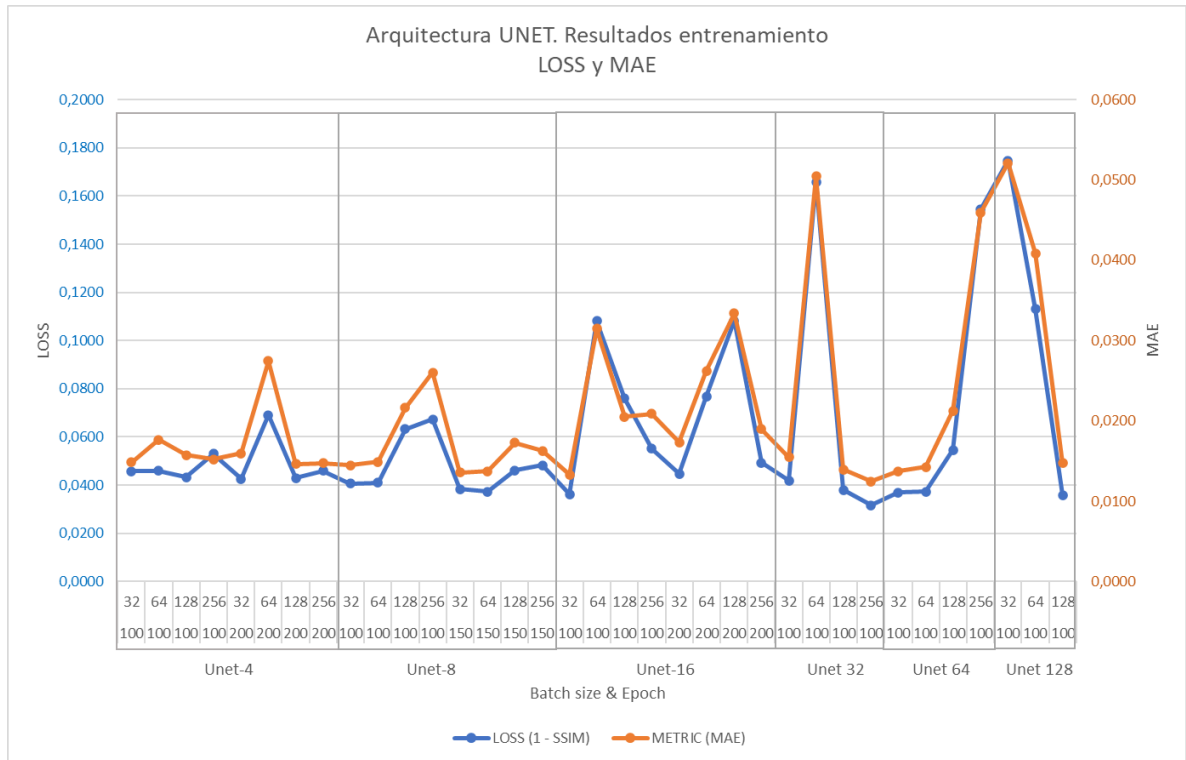


Figura 38 – UNET. Resultados de predicciones (SSIM y PSNR)

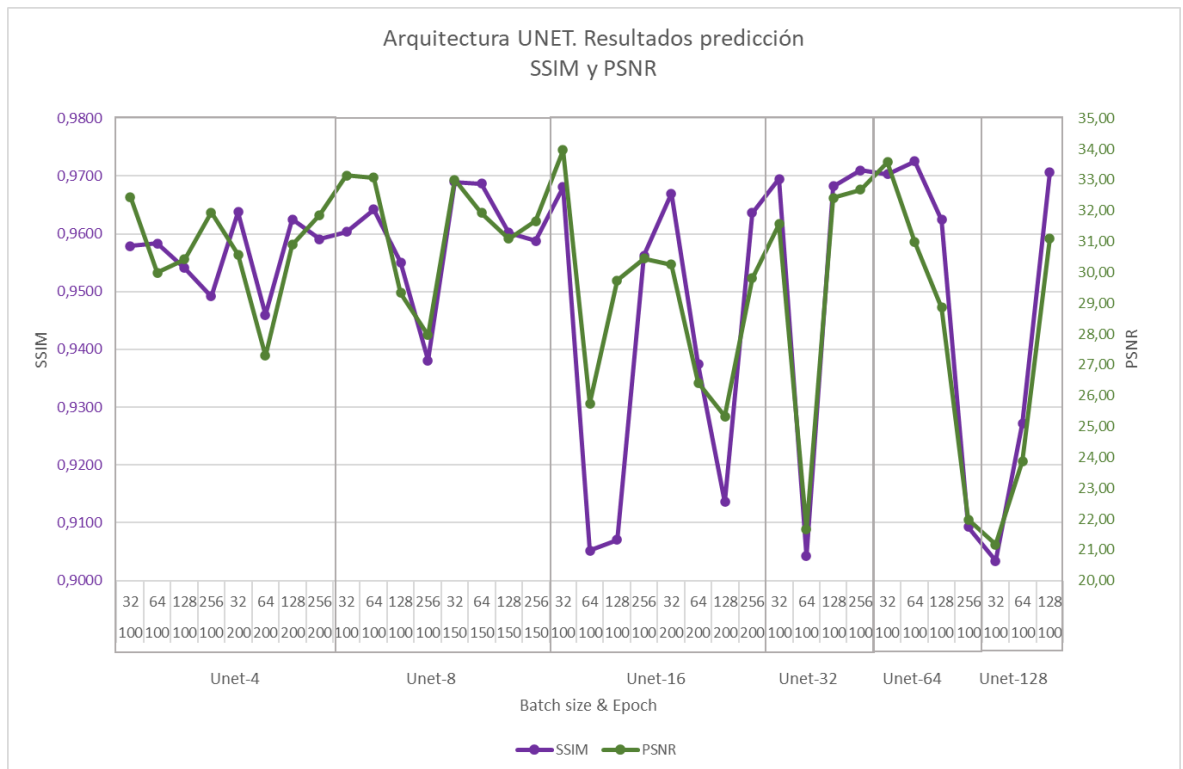
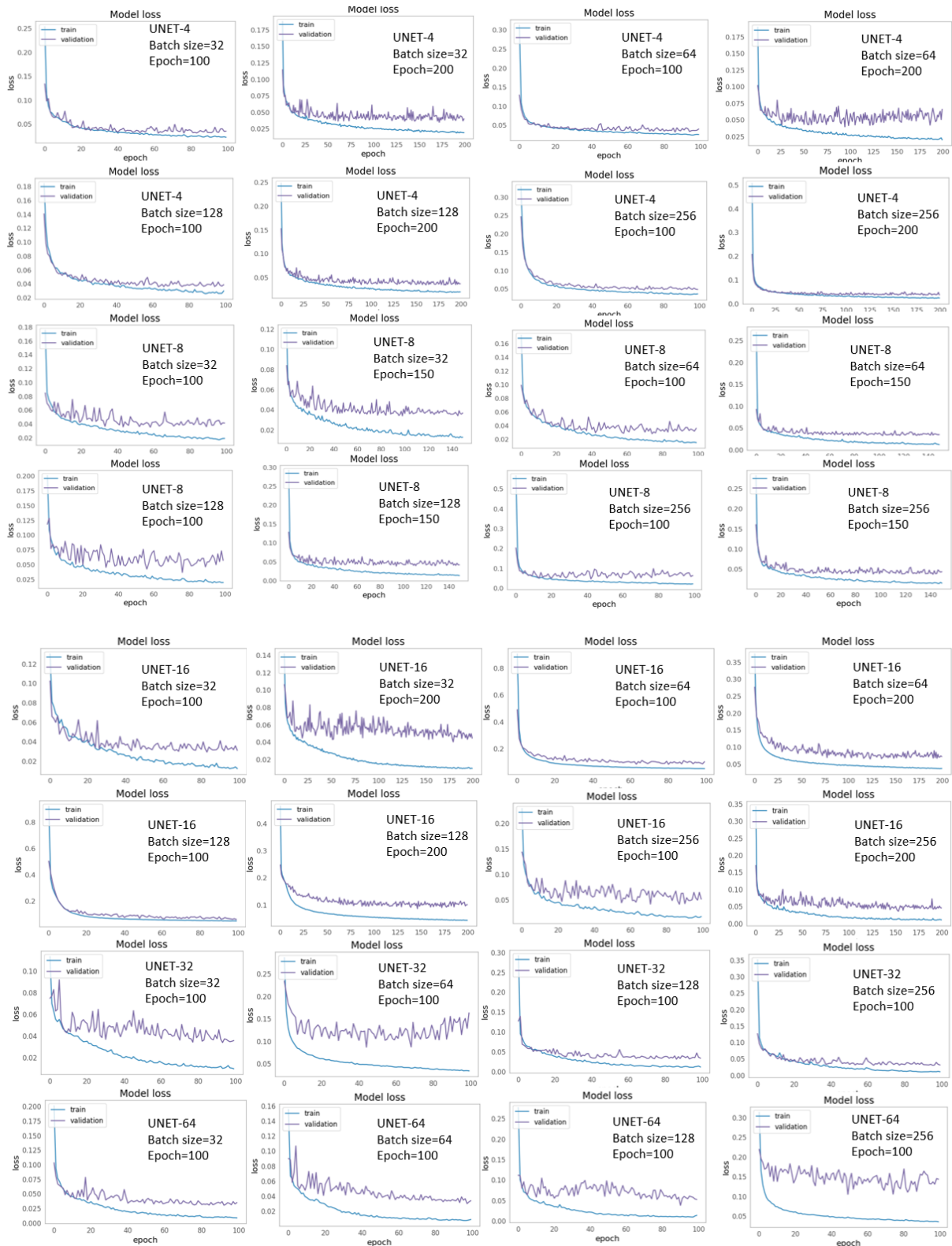


Figura 39 - Gráficas de entrenamiento de los modelos UNET.



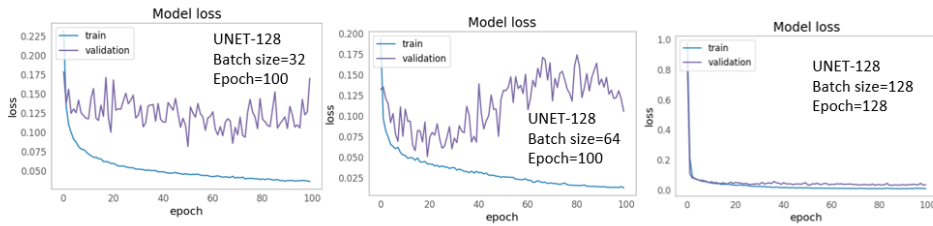
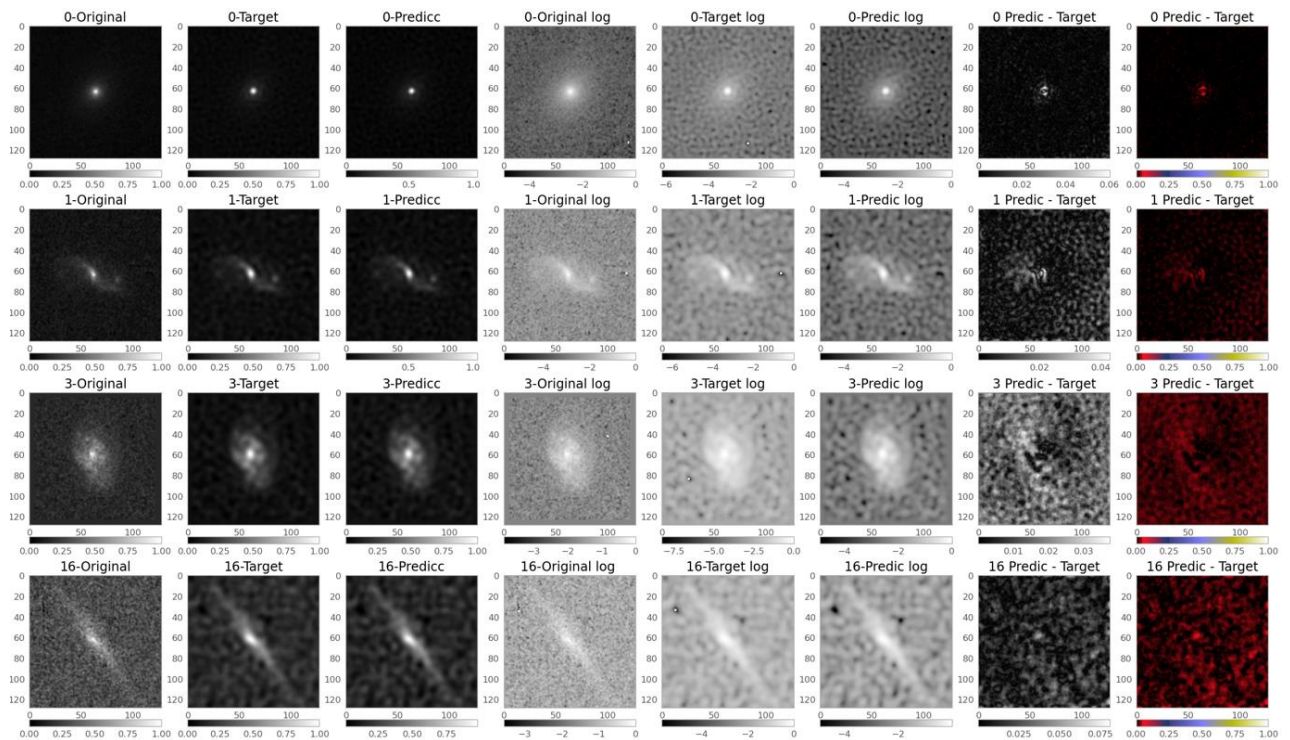


Figura 40 - UNET. Muestra de predicciones. De izquierda a derecha, imagen original, target y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y target, en escala de grises y en mapa de color "gist_stern".



Deep Learning como alternativa a la deconvolución de imágenes de galaxias capturadas con el telescopio espacial Hubble

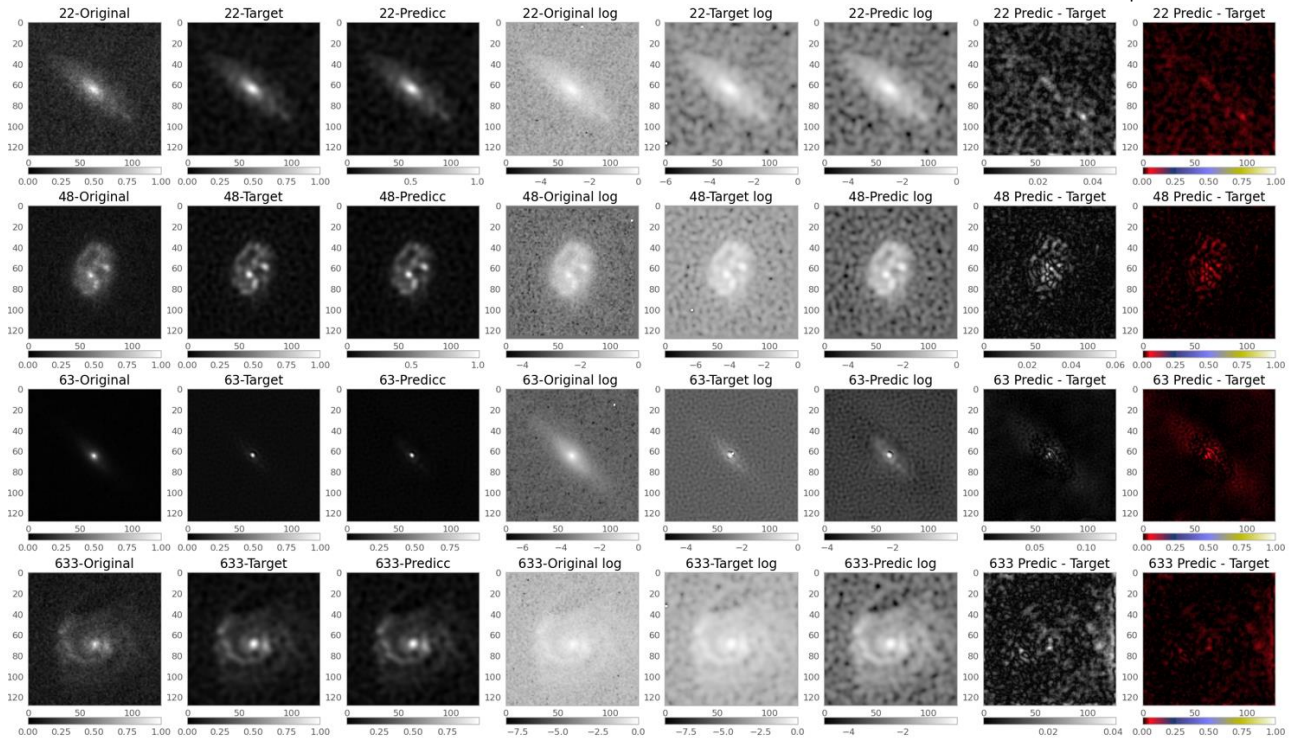
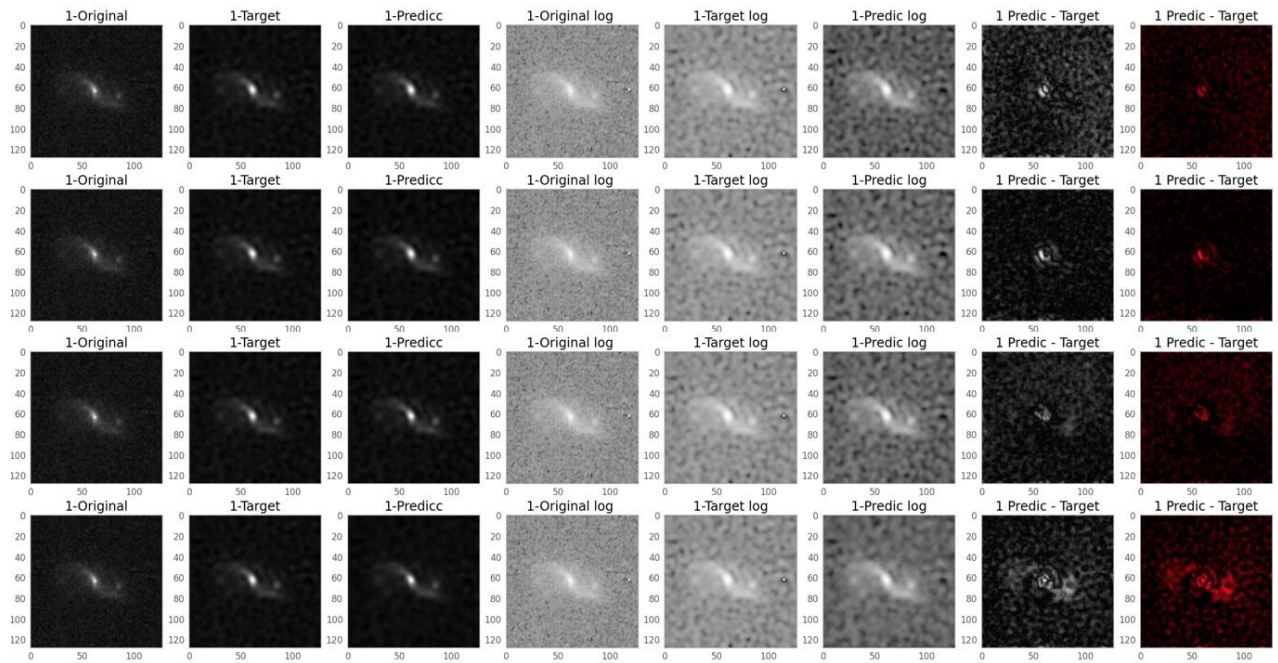


Figura 41 – UNET 64. Comparativa de la predicción de 2 imágenes con 4 entrenamientos distintos, de arriba abajo en cada bloque, batch size de 32, 64, 128 y 256.



Deep Learning como alternativa a la deconvolución de imágenes de galaxias capturadas con el telescopio espacial Hubble

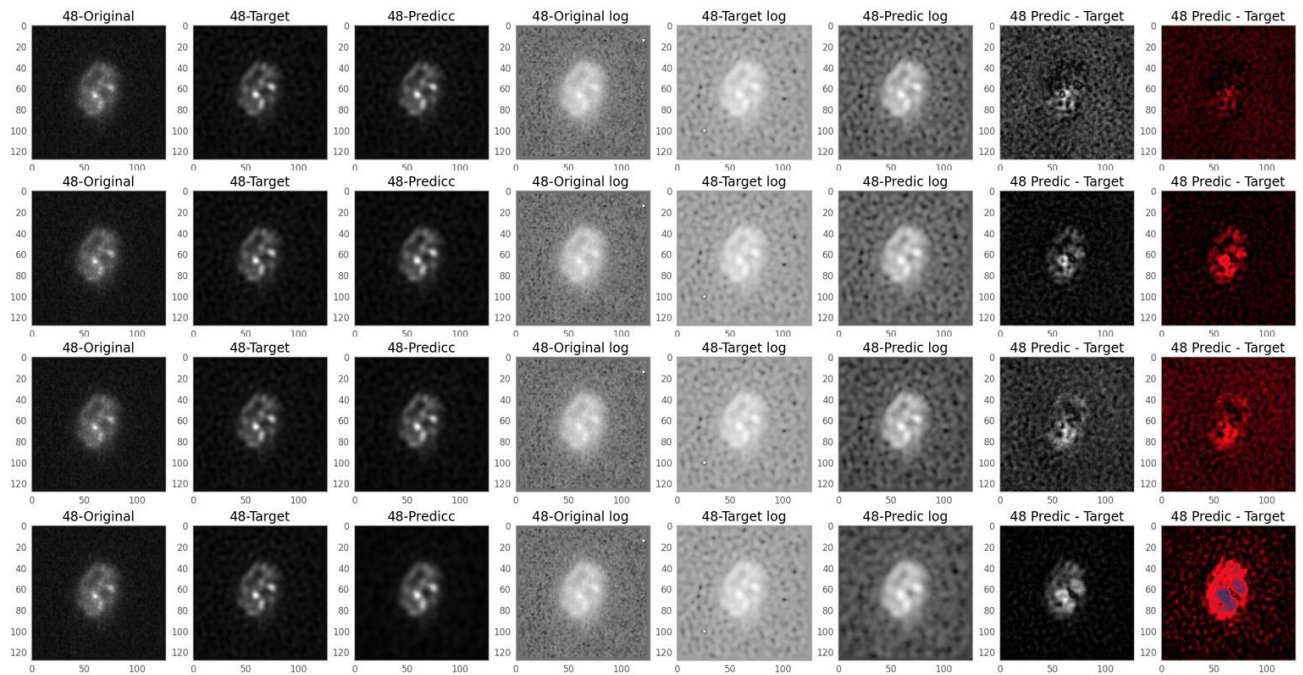
**Entrenamiento de la arquitectura DIDN**

Tabla 10 - DIDN. Hiperparámetros, métricas de entrenamiento y predicción. Los modelos marcados con (*) se han entrenado en Google Colab.

Modelos UNET					Evaluación entrenamiento			Predicciones			
MODELO	EPOCH	BATCH SIZE	OPT	LR	LOSS (1-SSIM)	METRIC (MAE)	T (seg)	MSE	RMSE	PSNR	SSIM
DIDN16DUB1	150	32	Adam	1.0E-3	0,0699	0,0179	1747	57,3758	7,5747	30,5435	0,9284
DIDN16DUB1	150	64	Adam	1.0E-3	0,0756	0,0189	1607	59,6794	7,7252	30,3726	0,9237
DIDN16DUB1	150	128	Adam	1.0E-3	0,0690	0,0176	1523	50,2390	7,0879	31,1204	0,9225
DIDN16DUB1*	150	256	Adam	1.0E-3	0,0718	0,0181	556	57,3095	7,5703	30,5485	0,9199
DIDN16DUB1*	200	32	Adam	1.0E-3	0,0754	0,0185	926	55,4776	7,4483	30,6896	0,9257
DIDN16DUB1*	200	64	Adam	1.0E-3	0,0742	0,0182	804	55,4183	7,4443	30,6943	0,9285
DIDN16DUB1*	200	128	Adam	1.0E-3	0,0691	0,0175	759	55,3753	7,4415	30,6976	0,9253
DIDN16DUB1*	200	256	Adam	1.0E-3	0,0861	0,0234	717	85,2797	9,2347	28,8223	0,9056
DIDN16DUB2	150	32	Adam	1.0E-3	0,0576	0,0170	2361	49,4050	7,0289	31,1931	0,9320
DIDN16DUB2	150	64	Adam	1.0E-3	0,0581	0,0186	2228	67,0202	8,1866	29,8687	0,9349
DIDN16DUB2*	150	128	Adam	1.0E-3	0,0630	0,0185	735	57,2114	7,5638	30,5598	0,9285
DIDN16DUB2*	150	256	Adam	1.0E-3	0,0561	0,0173	699	57,2000	7,5631	30,5568	0,9299
DIDN16DUB2*	200	32	Adam	1.0E-3	0,0652	0,0193	1215	57,5786	7,5879	30,5284	0,9244
DIDN16DUB2*	200	64	Adam	1.0E-4	0,0577	0,0186	1030	72,0832	8,4902	29,5525	0,9271
DIDN16DUB2*	200	128	Adam	1.0E-3	0,0779	0,0271	967	127,6453	11,2980	27,0708	0,9242
DIDN16DUB2*	200	256	Adam	1.0E-3	0,0737	0,0211	923	74,7484	8,6456	29,3949	0,9158

Deep Learning como alternativa a la deconvolución de imágenes de galaxias capturadas con el telescopio espacial Hubble

Figura 42 - DIDN. A la izquierda, resultados de entrenamiento (LOSS y MAE). A la derecha resultados de predicción con las imágenes de test (SSIM y PSNR)

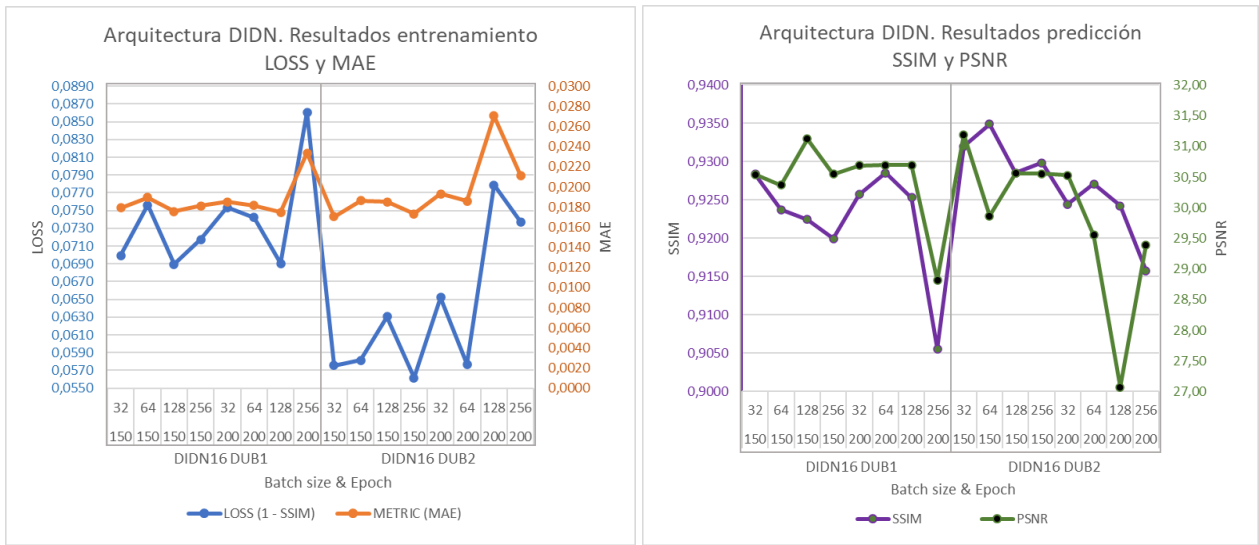
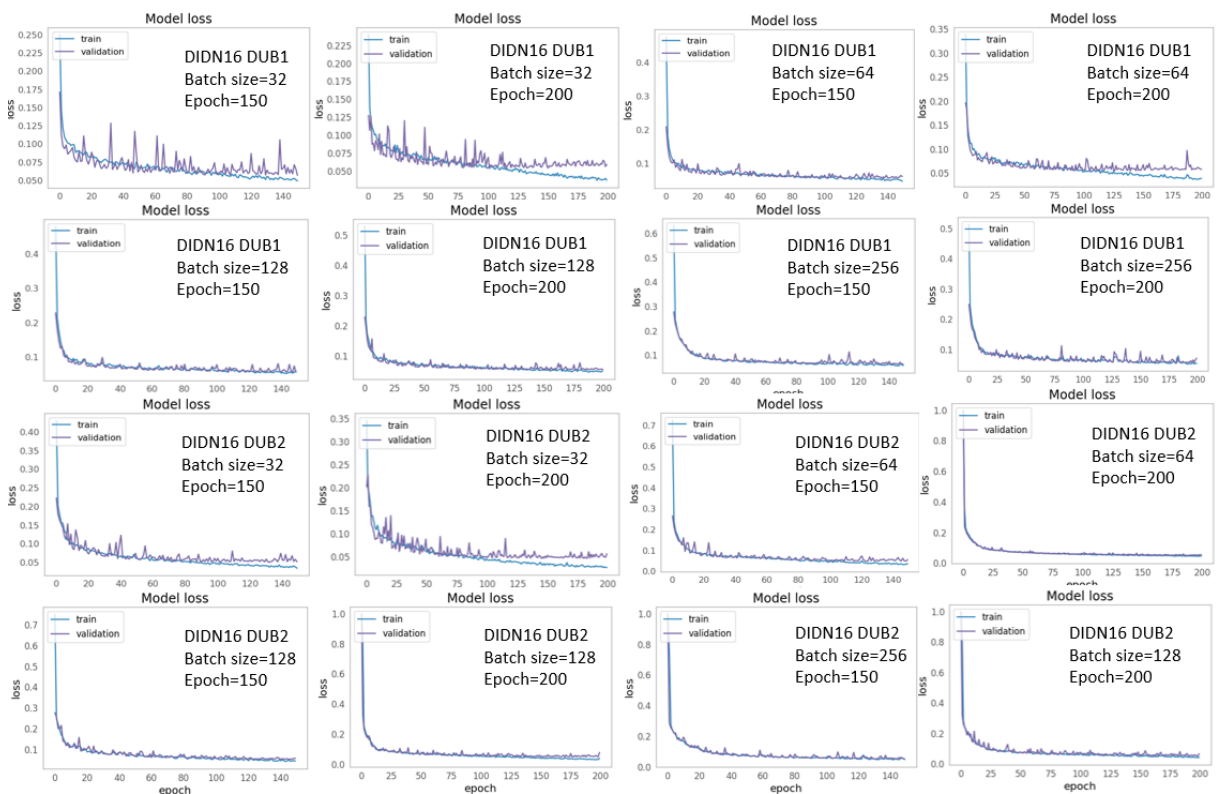


Figura 43 - Gráficas de entrenamiento de los modelos DIDN.



Deep Learning como alternativa a la deconvolución de imágenes de galaxias capturadas con el telescopio espacial Hubble

Figura 44 - DIDN. Muestra de predicciones. De izquierda a derecha, imagen original, target y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y target, en escala de grises y en mapa de color "gist_stern".

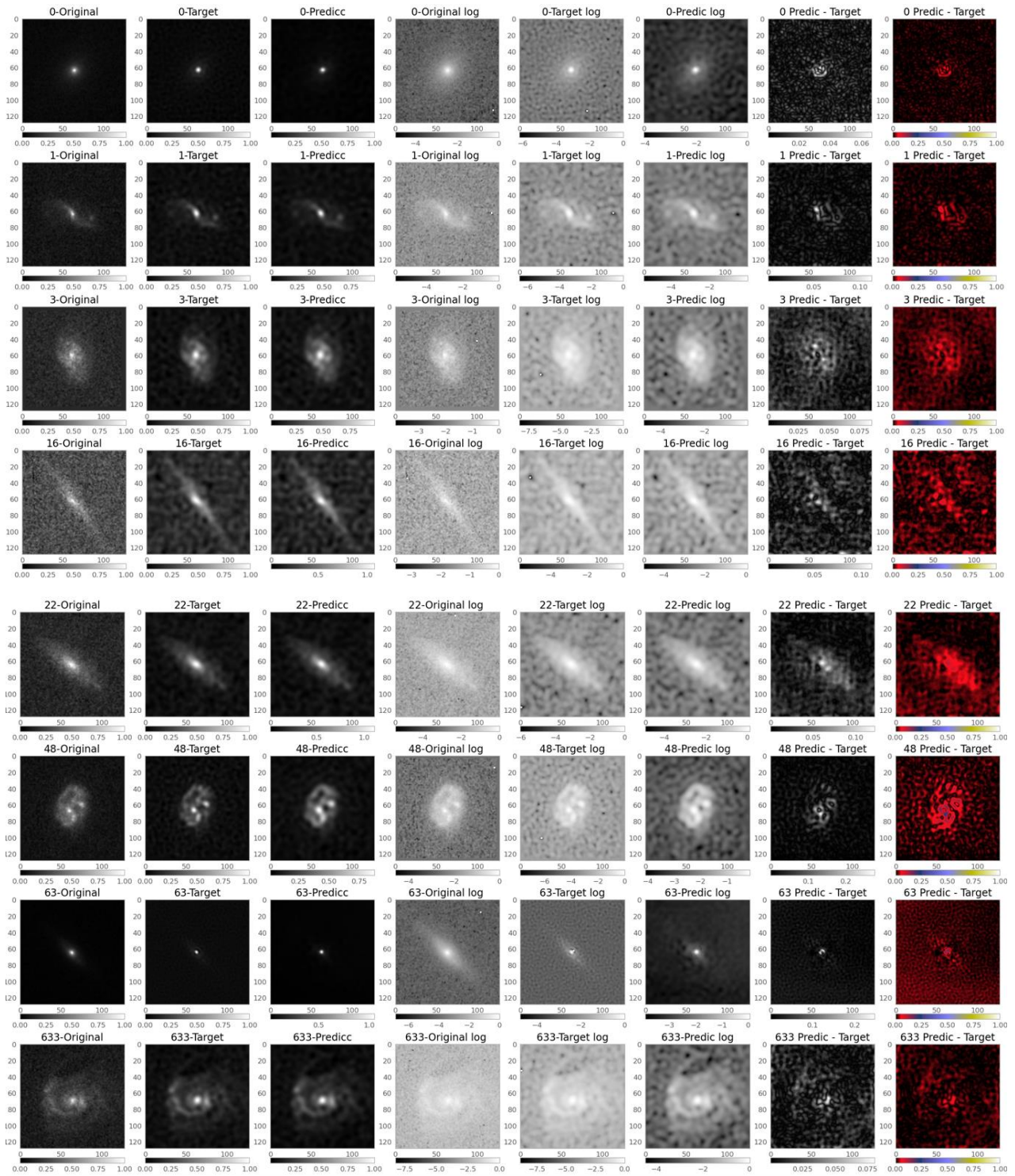
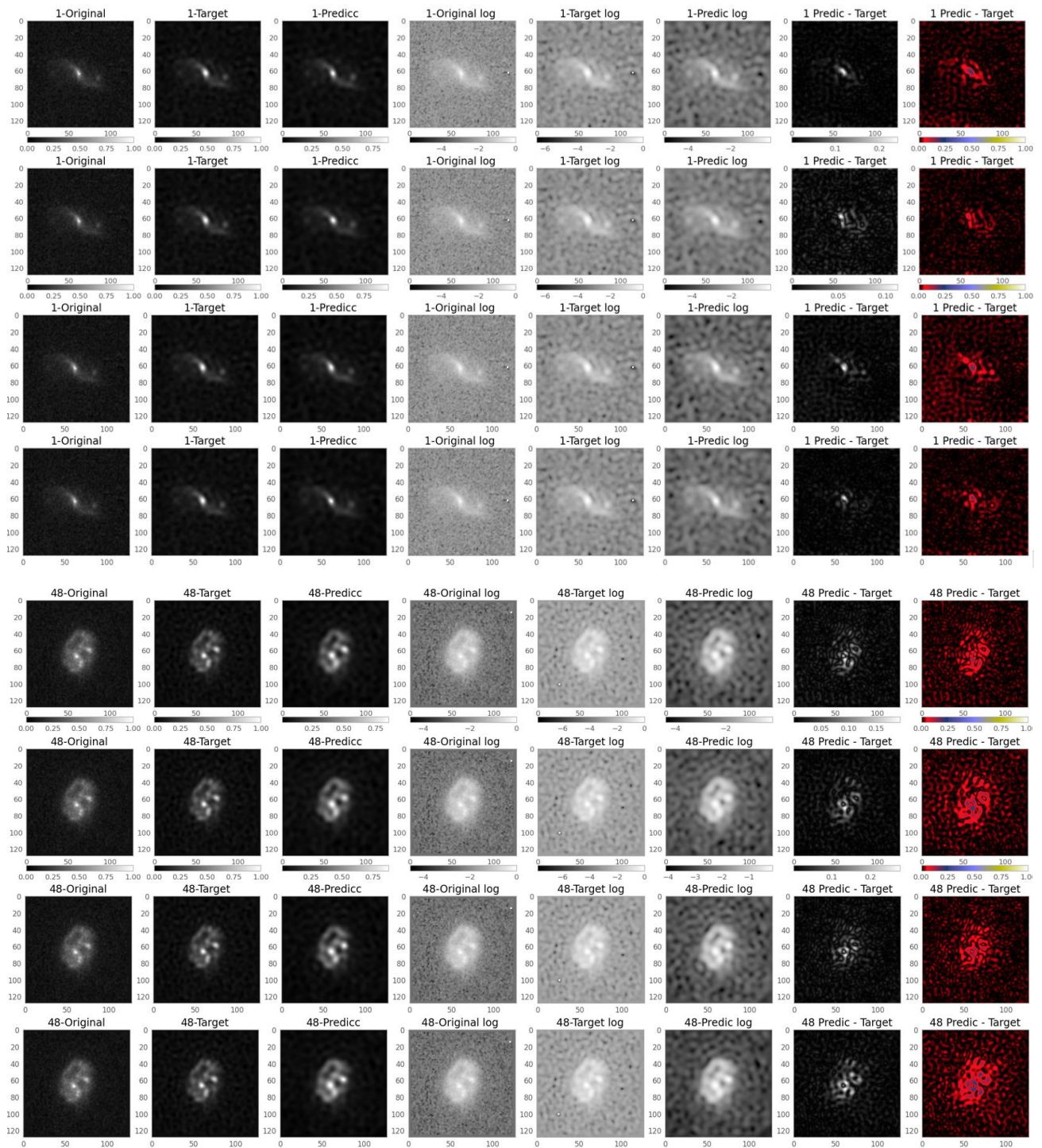


Figura 45 - DIDN16DUB2. Comparativa de la predicción de 2 imágenes con 4 entrenamientos distintos, de arriba abajo en cada bloque, batch size de 32, 64, 128 y 256.



ANEXO B. ARTÍCULO DE INVESTIGACIÓN

Deep Learning como alternativa a la deconvolución de imágenes de galaxias capturadas con el telescopio espacial Hubble

Luis Enrique Ramírez Peláez

Universidad Internacional de la Rioja, Logroño (España)

01/02/2023

RESUMEN

Las anomalías y artefactos de las capturas de imágenes de galaxias reales provocan la presencia de ruido que dificulta la labor de observación e investigación de los astrónomos. El ruido de Poisson y la Función de Dispersión de Punto (PSF) representan dos casos típicos que, de forma habitual, son objeto de tratamiento para su atenuación o supresión. Los avances de la Inteligencia Artificial han permitido la construcción de modelos que pueden ser entrenados para reconstruir imágenes de galaxias mitigando las aberraciones inherentes en el proceso de adquisición. En este proyecto se van a utilizar imágenes reales del telescopio espacial Hubble para entrenar tres arquitecturas de redes neuronales, AEPP2 (basada en Autoencoders), UNet y DIDN, y realizar una reconstrucción de estas, eliminando el efecto de la Función de Dispersión de Punto, conforme se haría con un algoritmo de deconvolución. Las pruebas realizadas aportan datos con los que se puede afirmar que el modelo de red U-Net es el que mejores resultados ofrece en este contexto.

PALABRAS CLAVE

Aprendizaje profundo, COSMOS survey, deconvolución, DIDN, función de dispersión de punto, imágenes astronómicas, telescopio espacial Hubble, UNet.

I. INTRODUCCIÓN

En la adquisición de imágenes astronómicas de galaxias se introducen anomalías provocadas por los instrumentos de observación y por la turbulencia atmosféricas. Dichas anomalías, denominadas artefactos afectan a la correcta interpretación de los datos recibidos, provocando errores de apreciación que afectan en mayor o menor medida a los estudios científicos.

Principalmente existen dos aberraciones que afectan a las imágenes capturadas de galaxias: El ruido de Poisson y la función de dispersión de punto (PSF). El ruido de Poisson [1] es característico de los sensores electrónicos asociados a los instrumentos ópticos empleados en la adquisición de imágenes y, debido al movimiento aleatorio de los fotones que inciden en el sensor, provoca fluctuaciones en la captación. Este valor de incertidumbre en la señal provoca un cierto nivel de ruido en la imagen final. La función de dispersión de punto, en inglés PSF (*point spread function*) [2] define la respuesta de un sensor óptico frente a la luz que incide en forma de impulsos, y es específica de cada instrumento de captación, manifestándose visualmente en un cierto grado de desenfoque.

La corrección de los errores provocados por el ruido de Poisson y la PSF se ha afrontado desde distintos puntos de vista con diferentes resultados. Los avances de la ingeniería informática, y en concreto, de la Inteligencia Artificial, ha permitido la introducción de nuevas vías de trabajo, que están en continuo desarrollo y de la que se están obteniendo avances significativos.

En este proyecto se va a abordar el uso de redes

neuronales para la reconstrucción de imágenes de galaxias afectadas por la función de dispersión de punto. Se ha elaborado un dataset lo suficientemente grande y completo para realizar el entrenamiento, construido a partir del “*COSMOS real galaxy dataset*” [3], formado por 118.992 imágenes de galaxias capturadas por el telescopio Hubble en el programa *COSMOS (The Cosmic Evolution Survey)* [4]. El dataset contiene los valores PSF de cada imagen lo que permite realizar una deconvolución para reconstruir cómo sería la imagen original sin ruido.

Las pruebas se han realizado entrenando tres redes neuronales: La arquitectura AEPP2 [5] basada en un modelo de *autoencoders*; la arquitectura U-Net [6] configurando distintos niveles de profundidad y la arquitectura DIDN [7] basada en U-Net a la que se añaden subprocesamientos adicionales.

La evaluación de las pruebas ha permitido determinar que la red U-Net es la que mejores resultados ofrece para la reconstrucción de imágenes de galaxias, obteniendo unos valores de rendimiento acorde a los estándares esperados.

II. ESTADO DEL ARTE

El efecto de la función de dispersión de punto

Las imágenes reales capturadas con un telescopio espacial son distorsionadas por el efecto de la PSF. Si esta PSF se considera invariable respecto a la posición en el plano del objeto, la imagen final se forma con la convolución de la real y la PSF. En el caso de imágenes astronómicas la convolución opera sobre la imagen real X y la matriz H correspondiente a la PSF, generando la imagen observada Y . Al proceso de convolución se le

suma además ruido de tipo Gaussiano, representado como N [8]

$$Y = H * X + N$$

Puesto que las imágenes capturadas contienen las aberraciones producidas por el ruido y la PSF, si lo que se pretende es invertir la dispersión de la luz reubicándola en su lugar original, lo que se hace matemáticamente es la operación inversa, llamada deconvolución [9] donde la clave es conocer la PSF correspondiente, cuyo cálculo, sin embargo, no es nada trivial.

Técnicas de deconvolución

Las técnicas de deconvolución clásicas como las descritas en [10] se basan en el post procesado de la imagen capturada utilizando la PSF conocida. Entre ellas se encuentra el método *CLEAN*, el algoritmo *Richardson - Lucy*, el filtro *Wiener* y las aproximaciones Bayesianas. Otras técnicas descritas en [11] usan la máxima entropía, gradientes, el algoritmo *R-L-EM* o *FMPE*. Sin embargo, no siempre es posible tener la imagen PSF o su estimación resulta compleja. En estos casos se puede hacer una deconvolución ciega (*blind deconvolution*) [12] donde, sin conocer la PSF, se hace una estimación de esta basándose en el uso de ciertas suposiciones, lo cual puede no generar resultados óptimos.

Deconvolución con aprendizaje profundo

Las técnicas de aprendizaje profundo han supuesto una nueva perspectiva en la resolución de la problemática de la captura de imágenes astronómicas. El entrenamiento de arquitecturas de redes neuronales con un suficiente número de pares *imagen capturada - imagen objetivo (target)* permite la predicción de cómo será una imagen real dada su captura, sin necesidad de conocer su PSF, ni real ni estimada, lo cual ahorra precisamente la parte más compleja de la reconstrucción de imágenes.

En [13] se hace una descripción de las distintas arquitecturas de redes neuronales que se emplean comúnmente para problemas de deconvolución de imágenes. Una opción son el uso de CNN clásicas (*Convolutional Neural networks*) [14] que emplean alternativamente capas de convolución con un tamaño determinado de *kernel*, y capas de *max pooling* para obtener características singulares. Aunque inicialmente podrían considerarse como válidas para la obtención de imágenes sin aberraciones, en la práctica se ha comprobado que no son tan exitosas, principalmente por el uso de *kernel* pequeños. Arquitecturas de tipo CNN como AlexNet o VGG incurren además en resultados pobres cuando el fondo de las imágenes no es uniforme, añadiendo áreas borrosas.

Otro tipo de arquitecturas son las redes *GAN* [15] que implican un planteamiento completamente distinto. Empleadas para deconvolución, la red generadora crea una imagen falsa que se parezca lo más posible a la original sin ruido, y la red discriminadora se entrena para distinguir las imágenes con ruido de las sin ruido. El proceso termina cuando el generador es capaz de construir una imagen que el discriminador considere libre de aberraciones. La mayor dificultad es aprender la naturaleza del ruido existente. En la propuesta de [16] se utiliza una CNN como red generadora para la GAN consiguiendo mejorar la predicción. Otro inconveniente habitual de las redes GAN, y que afecta a este escenario, es su dificultad para el

entrenamiento, que requiere de numerosas pruebas y ajustes.

El uso de arquitecturas de Autoencoders también permite abordar el proceso de deconvolución de imágenes, donde la primera fase de *encoder* transforma la imagen de entrada a una representación compacta intentando la eliminación del ruido. Posteriormente en la fase de *decoder* se recompone la imagen original sin las características eliminadas. La dificultad radica en encontrar un equilibrio entre la profundidad de la red y los resultados en la reconstrucción de la imagen original.

Una modificación al uso de Autoencoders que mejora su problemática es propuesta por [17]. Se trata de una red convolucional muy profunda de tipo *Autoencoders* que consta de una etapa de codificación y otra de decodificación. La particularidad radica en el uso de una característica de las redes *ResNet*, la existencia de *skip-connections*, que conectan capas no adyacentes de ambas etapas, lo que propaga en ambos sentidos la información, evitando la pérdida de características de la imagen de entrada.

Otras arquitecturas con una filosofía similar a Autoencoders son las U-net, que están adaptadas a los procesos de deconvolución y tienen su origen en tareas de segmentación de imágenes médicas [6]. Constan de una primera etapa de codificación en el que se extraen características propias de la imagen, y una segunda etapa de decodificación, simétrica a la primera. En ambas fases hay una comunicación entre capas simétricas, transfiriéndose información, y evitando que se pierdan características en la codificación. En [8] se utiliza una arquitectura U-Net para eliminar aberraciones en imágenes de galaxias simuladas. Partiendo de la imagen simulada y de la PSF, se hace un proceso de deconvolución empleando dos algoritmos, *Tikhonov* y ADMM. Se genera así un dataset de entrenamiento con el que la red *U-net* aprende los parámetros adecuados para, en la fase de predicción, ser capaz de eliminar la aberración de una imagen de prueba. El empleo del algoritmo *Tikhonov* y la red *U-Net* es denominado red *Tikhonet*.

Una evolución y mejora de la arquitectura *Tikhonet* es la propuesta de llamada *ShapeNet* que tiene en cuenta la forma de la estructura de las galaxias, añadiendo a la función de pérdida un nuevo término relacionado con el grado de elipticidad de estas.

En otros trabajos se construyen arquitecturas híbridas como *SeeingGAN* [18] donde se combinan *ResNet* y *U-Net* implementando además algunos cambios adicionales. Por otra parte, se modifica la función de pérdida usando una combinación de pérdida *Wasserstein* [19] y pérdida perceptual (pérdida sobre los mapas de características de las capas convolucionales).

Las arquitecturas U-Net se han posicionado como muy eficaces en situaciones donde hay escasez de datos, consiguiendo buenos resultados en dataset reducidos. Esta cualidad es también explorada con la arquitectura *Learnlet* [20] que se basa en el uso de la funcionalidad del descenso de gradiente para mejorar el poder expresivo de las wavelets. En este estudio se compara con U-Nets tradicionales con distinto grado de profundidad. Estas arquitecturas son también objeto de estudio para la deconvolución de imágenes de galaxias, como el trabajo realizado por [21] en el que se contrastan los resultados con *Tikhonet*, *Learnlet* y *U-Net64*, y se introducen

modificaciones para evaluar las métricas, basadas en el uso de una ventana en el espacio de la imagen donde se realizan los cálculos de error. Los autores concluyen que los dos últimos modelos y especialmente U-Net, obtienen resultados superiores al ser capaces de generalizar mejor que *Tikhonet*.

Arquitecturas como DIDN [17] amplían la complejidad de los modelos con unos resultados prometedores. Esta propuesta incluye etapas adicionales de subprocesamiento que operan como las U-Net. El proceso iterativo realizado consigue mejorar la eliminación de ruido en las imágenes.

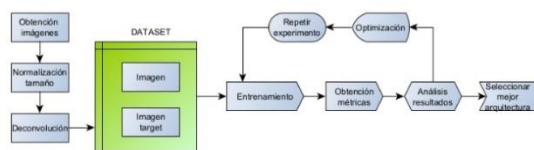
III. OBJETIVOS Y METODOLOGÍA

El objetivo principal es la reconstrucción de imágenes de galaxias capturadas con el telescopio Hubble afectadas por la función de dispersión de punto. Como objetivos secundarios se pretende:

1. Construir un dataset de imágenes homogéneo y válido para el entrenamiento, validación y test.
2. Conseguir un mecanismo eficiente para el tratamiento y almacenamiento de las imágenes.
3. Identificar y estudiar las técnicas de deconvolución existentes y sus parametrizaciones para obtener la imagen objetivo o *target* con el que entrenar.
4. Estudiar las arquitecturas de redes neuronales susceptibles de ser empleadas con éxito, así como la mejor parametrización.
5. Determinar las métricas de error que mejor se adapten a este tipo de estudios.
6. Identificar la red neuronal más apropiada para este ámbito.

Para el desarrollo del proyecto se ha seguido una metodología empírica cuantitativa siguiendo las fases de planteamiento del problema, formulación de hipótesis, análisis de los datos y extracción de conclusiones. En concreto y de forma particular al contexto de trabajo, el diseño de los experimentos se ha realizado conforme se muestra en la Figura 1.

Figura 1 - Diseño de los experimentos.



IV. CONTRIBUCIÓN

Atendiendo a la definición de los objetivos principal y secundarios, se ha propuesto una comparativa de arquitectura válidas para la reconstrucción de imágenes de galaxias. Se ha buscado un conjunto de capturas de las que se dispone su PSF, con la que se realizará una deconvolución, que será la imagen objetivo a predecir por la red neuronal. En términos prácticos la red neuronal realizará un proceso similar a una deconvolución ciega.

Construcción del dataset de trabajo

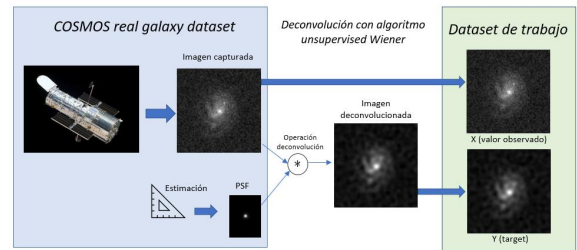
A partir de la colección de imágenes “*COSMOS real galaxy dataset*” se ha construido un *dataset* de trabajo formado por 9.625 imágenes de dimensiones 128x128 píxeles. Las capturas del Hubble incluyen tamaños desde algunos pocos píxeles hasta los 900x900. Se han

seleccionado todas aquellas comprendidas entre los 110x110 píxeles y los 140x140 píxeles, adaptando a la dimensión final eliminando filas y columnas desde los extremos o añadiéndolas con valor igual a 0, respetando de esta forma la información de la imagen.

Para el entrenamiento los valores de las imágenes son normalizados al rango [0,1] sin que por ello se vea afectada la información de la imagen.

El proceso de deconvolución se ha llevado a cabo con el algoritmo *Unsupervised Wiener* que hace una estimación automática del parámetro de regularización necesario para la deconvolución, lo cual permite automatizar el proceso, independizándolo del criterio visual del científico.

Figura 2 - Proceso de creación del dataset de trabajo.



Métricas de evaluación

La evaluación del error en las predicciones se realiza con las siguientes métricas:

Mean squared error (MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2$$

Peak Signal-to-Noise Ratio (PSNR)

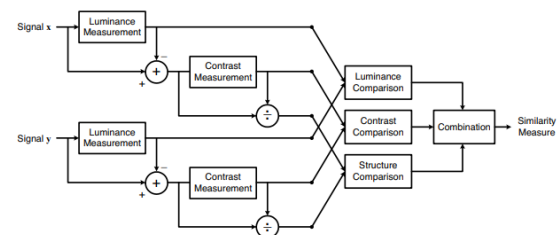
$$PSNR = 10 \log_{10} \left(\frac{MAX_i}{MSE} \right) = 20 \log_{10} \left(\frac{MAX_i}{\sqrt{MSE}} \right)$$

donde MAX_i es el máximo valor de los píxeles, representados con B bits por muestra, y $MAX_i = 2^B - 1$

La PSNR mide la relación entre la máxima energía posible de una señal y el ruido que le afecta. Medida en decibelios, los valores normales oscilan entre 30 dB y 50 dB, siendo mejores cuanto mayor sean.

Structural Similarity Index (SSIM). Esta medida propuesta por [21] se aproxima a lo que la visión humana realiza para comparar dos imágenes. En lugar de utilizar los datos en bruto de los valores de la imagen, emplea una combinación de luminancia, contraste y estructura con el objetivo de evaluar la similitud estructural entre los píxeles vecinos.

Figura 3 - Diagrama del cálculo de SSIM. Imagen:[22]



Cada uno de estos tres elementos se calcula de la siguiente manera:

$$Luminancia(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

μ_x es la media de los píxeles de la imagen x
 μ_y es la media de los píxeles de la imagen y

$$Contraste(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

σ_x^2 es la varianza de la imagen x
 σ_y^2 es la varianza de la imagen y

$$Estructuras(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

σ_{xy} es la covarianza
 C_1, C_2 y C_3 son coeficientes de estabilización para las divisiones

De esta forma, el cálculo de SSIM se expresa de la siguiente manera:

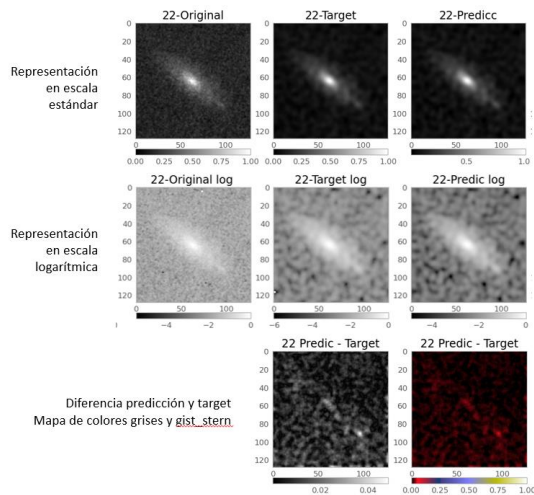
$$SSIM(x, y) = l(x, y)^\alpha \cdot c(x, y)^\beta s(x, y)^\gamma$$

α, β , y γ son pesos para ponderar cada uno de los componentes

El valor de SSIM oscila entre 0 y 1, siendo 1 cuando dos imágenes son idénticas. Aunque depende de cada caso, un valor superior a 0,97 indica que las imágenes son muy similares. Esta métrica es muy sensible y un pequeño cambio en el valor puede suponer una importante mejora o empeoramiento. Para el ámbito de este proyecto, el uso de SSIM permite evaluar el proceso de deconvolución simulado en la red neuronal, es decir, determinar si la imagen calculada por la red a partir de la original es suficientemente similar al *target*.

También se va a proporcionar un mecanismo para una evaluación visual de los resultados, comparado la imagen calculada con la imagen objetivo. Por una parte, se muestra el resultado de la ejecución del modelo sobre las imágenes de test en escala logarítmica (Figura 4), lo cual permite apreciar de mejor manera algunas estructuras de las galaxias.

Figura 4 - Resultados de una prueba. Primera fila y de izquierda a derecha, imagen original, objetivo y predicha. Fila central y de izquierda a derecha, imágenes originales, objetivo y predicha en representación logarítmica. Fila inferior de izquierda a derecha, diferencias entre la imagen predicha y la imagen objetivo, con mapas de color escala de grises y “gist_stern”.



Por otra parte, se ha realizado una resta de valores de la imagen objetivo y la predicha por el modelo (valor residual), mostrándose en un mapa de color visualmente significativo, en concreto el mapa “gist_stern”, que permite apreciar fácilmente las pequeñas variaciones. una predicción buena, como la mostrada en la Figura 4, tendrá una imagen residual con tonos negros. Las predicciones peores irán apareciendo en tonos rojizos pasando a azules y amarillentos si estas son muy incorrectas.

Hiperparámetros

Se han empleado los mismos valores en todos los modelos para poder comparar en condiciones de ejecución similares y así determinar la mejor arquitectura.

Optimizador: Se ha empleado el optimizador *Adam* con valores de *learning rate* de 1.0E-3, 1.0E-4, 1.0E-5 y 1.0E-6 según las particularidades de cada entrenamiento.

Función de pérdida (loss): Se utiliza el índice de similaridad estructural (SSIM) al ser una métrica apropiada para comparar imágenes, invirtiendo su valor, es decir, restando de 1.

Metric: La métrica empleada en el entrenamiento es *Mean Absolute Error* (MAE). Esta función permite valorar el aprendizaje de la red, algo que ocurrirá cuando este valor vaya descendiendo paulatinamente.

Early Stop: La detención se produce cuando la tasa de pérdida *loss* sea menor a 0,02, o lo que es lo mismo, cuando la SSIM alcance el valor de 0.98.

Epoch: Ciclos de entrenamiento de 100, 150 y 200 epoch, adaptando a las características de cada red y los resultados preliminares.

Batch size: Tamaños de 32, 63, 128 y 256.

Distribución de los datos de entrenamiento, validación y test: Todos los modelos se han entrenado con las mismas imágenes de entrenamiento y validación, aunque en cada ejecución se han mezclado al inicio utilizando el parámetro *shuffle* de *Keras*. Todas las arquitecturas se han testeado con idénticas imágenes y así poder comparar eficientemente los resultados. En concreto se han usado un 70% de datos en entrenamiento, 17% datos de validación y 13% datos de test.

Recursos hardware y software

Las pruebas de entrenamiento se han ejecutado en dos entornos: local y online: Entorno local: Ordenador con CPU AMD RYZEN 5 5600X y GPU NVIDIA RTX 3060Ti 8GB. Entorno online: Google Colab. GPU NVIDIA A100-SXM4 40GB

Toda la parte de programación se ha realizado en Python v3.9.12 usando las siguientes librerías: TensorFlow v2.10.0, Keras v2.10.0, Scikit-image v0.19.2, Numpy v1.21.5, Matplotlib v3.5.1, Astropy 5.0.4

Arquitecturas empleadas

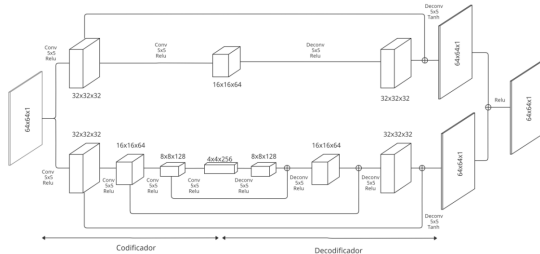
Arquitectura AEPP2

Se trata de un modelo basado en la propuesta de [23] que utiliza autoencoders en dos ramas, en las que se realizan operaciones de convolución y deconvolución. La arquitectura AEPP2 fue empleada por [5] considerando algunas modificaciones para una mejor adaptación a la eliminación de ruido.

Este modelo consta de 2 ramas, la superior incluye 2 operaciones de convolución (codificación o compresión) y otras 2 de deconvolución (decodificación o descompresión); la inferior consta de 3 operaciones de cada tipo. La idea es transformar la imagen original en una representación compacta intentando eliminar el ruido (compresión) y posteriormente recomponer la imagen original sin las características eliminadas (descompresión). La existencia de 2 ramas se justifica en

poder incluir en el mismo proceso dos niveles de eliminación de ruido. Así, la rama inferior produce una mayor compresión, eliminando más cantidad de ruido, pero recordando menos detalles. La rama superior, de menor capacidad de compresión, elimina menor ruido, pero permite recuperar mejor la imagen original.

Figura 5 - Arquitectura AEPP2. Imagen de [5]

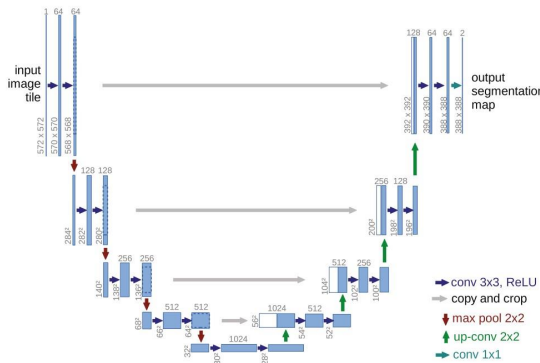


Para evitar la pérdida de información motivada por el uso de *autoencoders*, se incluyen *skip-connections* que son conexiones entre capas del inicio y del final para conservar las características de las imágenes. Este problema tiene su origen en el uso de demasiada profundidad en el procesamiento, lo que puede provocar un desvanecimiento de gradiente. De hecho, el modelo de partida propuesto por [23] limita esta profundidad

Arquitectura U-Net

Este modelo utiliza una filosofía parecida a los Autoencoders ejecutando dos etapas, una de codificación donde se extraen las características propias de la imagen, y una segunda etapa de decodificación, simétrica a la primera, y donde se reconstruye la imagen original. Existe una comunicación entre capas simétricas, lo cual transmite la información de la entrada a la salida, evitando que se pierdan demasiadas características. También se realiza una *maxpooling* tras cada etapa, reduciendo la dimensionalidad sin perder información importante.

Figura 6 - Arquitectura U-Net. Imagen: [6]



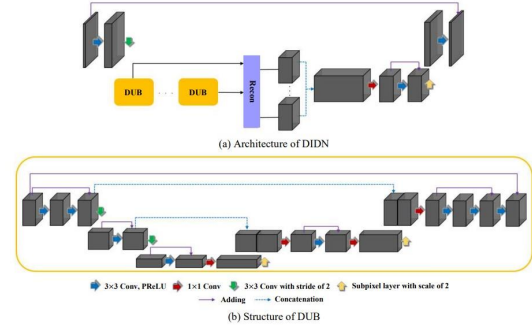
Para las pruebas se han construido 6 modelos con distinto nivel de profundidad (Tabla 2 del Anexo).

Arquitectura DIDN

Este modelo propuesto por [7] parte de la estructura de una U-Net pero añade un subprocesamiento iterativo del proceso de convolución (DUB), afinando y mejorando la eliminación de ruido. En este caso, la red consta de 4 etapas: extracción de características, reducción y escalado iterativo (subprocesamiento DUB), reconstrucción y mejora. El proceso iterativo realizado con la estructura es en esencia una red U-Net, y su iteración consigue mejorar

la eliminación de ruido en las imágenes. Para las pruebas se ha construido dos variantes simplificadas del modelo original, formada por 16 filtros en entrada, con 1 y 2 unidades de subprocesamiento DUB.

Figura 7 - Arquitectura DIDN. Esquema general (a) y subprocesamiento interno (b) – Imagen: [7]



V. EVALUACIÓN Y RESULTADOS

La evaluación de los resultados ha comprendido el análisis de las estadísticas de entrenamiento, el estudio de las métricas de la predicción y la comparativa visual de las imágenes, de las que se han incluido algunos ejemplos en el Apéndice.

Tabla 1 - Parámetros, métricas de entrenamiento y predicción de los mejores modelos de cada arquitectura.

ARQUITECTURA Y MODELO		EVALUACIÓN ENTRENAMIENTO			MÉTRICAS PREDICCIONES				
MODELO	EPOCH H	BATCH SIZE	LOSS (1-SSIM)	METRIC (MAE)	T (SEG)	MSE	RMSE	PSNR	SSIM
AEPP2	100	128	0,0571	0,0154	819	37,8199	6,1498	32,3536	0,9540
UNET4	200	32	0,0426	0,0159	1226	57,0713	7,5546	30,5666	0,9638
UNET8	100	64	0,0409	0,0149	648	32,1079	5,6664	33,0647	0,9642
UNET16	200	32	0,0448	0,0173	2690	61,4367	7,8382	30,2465	0,9670
UNET32	100	256	0,0316	0,0125	674	35,0267	5,9183	32,6868	0,9710
UNET64	100	32	0,0369	0,0137	1666	28,5203	5,3404	33,5793	0,9703
UNET128	100	128	0,0359	0,0147	3638	50,3681	7,0971	31,1092	0,9707
DIDN16	150	128	0,0690	0,0176	1523	50,2390	7,0879	31,1204	0,9225
DIDN16 DUB1									
DIDN16 DUB2	150	32	0,0576	0,0170	2361	49,4050	7,0289	31,1931	0,9320

Evaluación de la arquitectura AEPP2

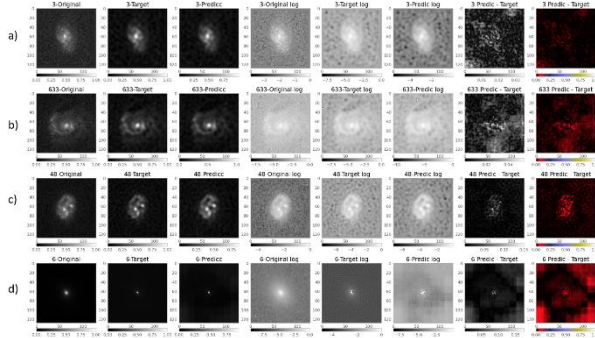
Se han realizado 8 experimentos con cada valor de *batch size*: 32, 64, 128 y 256 en dos tandas de 100 y 200 *batch*, observando que, generalmente, a partir de la *epoch* 40 la convergencia prácticamente se alcanzaba. Igualmente se constata que con tamaños más pequeños de *batch size* el modelo produce cierto *overfitting*, aspecto que se ve corregido con tamaños superiores.

En todas las pruebas los resultados son muy similares, con valores de 0,95 para SSIM (con diferencias de milésimas) y entre 31 y 32 para PSNR. El modelo con mejor comportamiento es el ejecutado con tamaño de *batch size* igual a 128 ya que ofrece además las mejores cifras para PSNR y MSE.

Respecto al resultado visual y en consonancia a los valores de SSIM obtenidos, las reconstrucciones realizadas son generalmente buenas, aunque con diferencias según el tipo de galaxia en cuestión. Sin embargo, no es trivial encontrar un patrón para mejores o peores resultados en función de la fisonomía de la galaxia. En la Figura 8 se pueden ver algunos ejemplos del comportamiento del modelo. En 8.a) se realiza una

reconstrucción muy buena. El residuo que queda al hacer la diferencia entre la imagen predicha y la imagen objetivo muestra valores bajos, sin apreciarse estructuras resaltadas de la galaxia.

Figura 8 - Ejemplos de predicciones. a) reconstrucción óptima. b) Reconstrucción eficiente de la estructura de la galaxia. c) Reconstrucción deficiente de la estructura de la galaxia. d) Efecto cuadrícula. De izquierda a derecha, imagen original, *target* y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y *target*, en escala de grises y en mapa de color "gist_stern".



En 8.b) la reconstrucción es peor al quedar un residuo con más información. En 8.c) la estructura de la galaxia se ve afectada negativamente por la predicción, algo que se aprecia a simple vista observando ambas imágenes y se ratifica con el valor residual. En 8.d) se observa una anomalía que se ha denominado “efecto cuadrícula”. La predicción genera una cuadrícula por toda la imagen, que se aprecia mejor en la escala logarítmica y en la imagen residual. Este efecto ocurre en gran parte de las predicciones, aunque suele ser muy atenuado y solo haciendo una ampliación abultada puede llegar a apreciarse. Esta cuadrícula la forman elementos de 16x16 píxeles, en total 8x8 mini bloques. Se da la circunstancia que la arquitectura emplea *kernel* paulatinamente hasta llegar a un nivel mínimo de imagen de 16x16, que parece ser el motivo de esta irregularidad.

Este comportamiento fruto de la arquitectura de la red neuronal hace que los datos relativamente buenos de las métricas se vean ensombrecidos, llevando a la conclusión de que este modelo no es adecuado al contexto de trabajo.

Evaluación de la arquitectura UNET

Se han realizado 35 pruebas con las 6 variaciones de la arquitectura inicial, con la configuración de hiperparámetros común, pero eligiendo determinados valores en función de los resultados preliminares, ya que en algunas ocasiones el modelo no convergía y se ha tenido que variar el plan de entrenamiento. Por lo general, a mayor complejidad de la red neuronal, menor debe de ser la *learning rate* para conseguir que el entrenamiento se realizase sin errores. También se ha observado que en los modelos más complejos (Unet32, Unet64 y Unet128), entrenados en *Google Colab*, la ejecución de 200 *epoch* hacía caer al modelo en *overfitting*, por lo que las pruebas se han realizado solamente con 100 *epoch*.

De forma general los resultados son buenos, superándose en casi todos los casos la cifra de 0.95 para el SSIM, llegando a 0.96 en numerosas ocasiones y alcanzando 4 veces un valor de 0.97, en las que además se obtienen cifras entre 30 y 33 para la PSNR. En términos absolutos el mejor modelo se corresponde a Unet-64 con

batch size igual a 32 (Tabla 1).

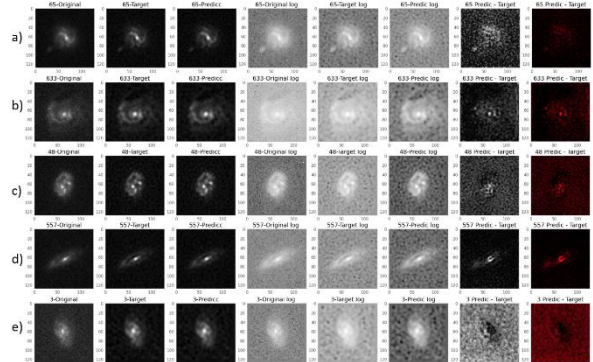
Existe una tendencia a obtener mejores resultados conforme se aumenta la complejidad de la arquitectura, pero, aunque el valor de SSIM es muy sensible a pequeños cambios y el objetivo de las pruebas es maximizarlo, el aumento del coste de entrenamiento asociado es significativamente superior a la ganancia en SSIM y PSNR. Los modelos sencillos ofrecen resultados más homogéneos, mientras que los complejos obtienen métricas muy buenas, pero también inferiores incluso a los sencillos.

Una característica de esta arquitectura es que la convergencia se alcanza de forma rápida. En la mayoría de los casos con poco más de 30 *epoch* se consiguen valores aceptables cercanos al óptimo.

En algunos entrenamientos se ha producido *overfitting*, Esta circunstancia ha tenido lugar en todos los tipos de modelos, independientemente de su complejidad, pero solo en algunas ocasiones ha sido un factor perjudicial, siendo por el contrario un sobreajuste muy leve.

La evaluación visual de los resultados ofrece un balance positivo. En general las reconstrucciones realizadas por las distintas configuraciones de la arquitectura U-Net son buenas. En la figura 9 se muestran algunos ejemplos.

Figura 9 - Ejemplos de predicciones realizadas con la red U-Net. De izquierda a derecha, imagen original, *target* y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y *target*, en escala de grises y en mapa de color “gist_stern”.



En 9.a) se consigue una reconstrucción óptima. En 9.b) y c) las diferencias son mínimas, respetándose la estructura de las galaxias. En 9.d) el proceso no es tan perfecto porque el cuerpo de la galaxia contiene algunas diferencias que se denotan por la imagen residual. En 9.e), aunque aparentemente el resultado es peor, la estructura de la galaxia se reconstruye bastante bien, mostrando la imagen residual tonos negros en el centro.

Con U-Net no se aprecia en ningún momento la ocurrencia del denominado “efecto cuadrícula”.

Evaluación de la arquitectura DIDN

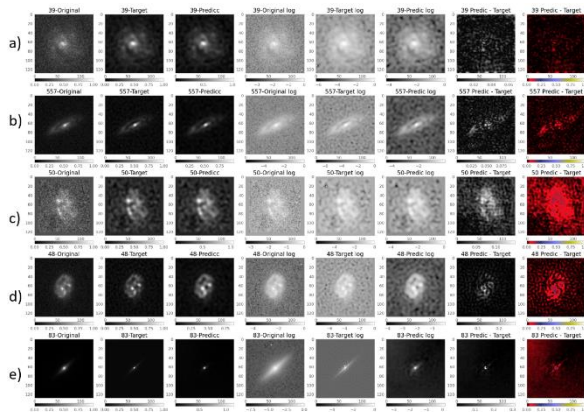
Las pruebas con esta arquitectura se han realizado con los 2 modelos de 1 y 2 unidades de sub procesado DUB.

Los resultados no alcanzan una calidad destacable, pues en ningún caso se supera el valor de 0.93 para SSIM. El modelo con 2 bloques de sub procesado DUB se comporta ligeramente mejor, siendo la prueba con *batch size* igual a 32 y entrenada con 150 *epoch* la que mejores resultados obtiene (Tabla 1).

Una característica común a todos los entrenamientos es que la convergencia es algo lenta pero estable, y no se ha observado tendencia al *overfitting*. Los valores de la tasa de pérdida descienden paulatinamente hasta las 100 epoch aproximadamente, momento en el que dicha tasa deja prácticamente de mejorar.

El análisis visual de las imágenes predichas permite apreciar de forma generalizada defectos en mayor o menor medida. Aun así, en algunas ocasiones estas predicciones son satisfactorias. En la Figura 10 se muestran algunos resultados. Se ha constatado la aparición del “efecto cuadrícula” observado con la arquitectura AEPP2. Este comportamiento no es muy frecuente y ha aparecido solo en el modelo con 1 bloque DUP. Los entrenamientos con el modelo de 2 DUB no muestra esta particularidad. Además, el efecto no es exactamente el mismo, porque en esta ocasión, la anomalía la forman una o dos franjas a modo de marco de la imagen.

Figura 10 - Ejemplos de predicciones realizadas con la red DIDN. De izquierda a derecha, imagen original, *target* y predicción. Sus correspondientes vistas en escala logarítmica. Finalmente, diferencia entre predicción y *target*, en escala de grises y en mapa de color “gist_stern”.



En la imagen 10.a) la predicción es óptima. En 10.b) la imagen residual indica que la reconstrucción falla en la estructura de la galaxia. Este comportamiento se manifiesta a simple vista en 10.d). En 10.c) el valor residual indica bastante diferencias que afectan a toda la imagen. En 10.e) aparentemente no hay demasiado valor residual, sin embargo, el halo de la galaxia se ve afectado y se pierde de forma sustancial.

VI. DISCUSIÓN

La arquitectura U-Net se alza como la mejor propuesta en esta comparativa. Los resultados cuantitativos obtenidos de las métricas SSIM y PSNR para algunas de sus configuraciones son los mejores, logrando además una cifra muy aceptable considerando la naturaleza de estas medidas de evaluación. El objetivo perseguido de deconvolucionar imágenes de galaxias capturadas con el telescopio espacial Hubble es posible con este modelo de red neuronal. Hay que resaltar que la arquitectura U-Net respeta enormemente la estructura y fisonomía de las imágenes, en este caso galaxias, no introduciendo anomalías ni deformaciones que suponga un perjuicio desde el punto de vista científico. Los resultados en las predicciones que en ocasiones no son satisfactorias implican una merma en la calidad de la reconstrucción, pero no introducen alteraciones como el “efecto cuadrícula” observado con la red DIDN y en mayor medida con la red AEPP2.

La gran flexibilidad y fácil escalabilidad de la red U-Net hace posible la experimentación y la ejecución de entrenamientos que permitan determinar la mejor parametrización para maximizar los mejores resultados. Todo ello contribuye a contrarrestar la relativa variabilidad de los modelos, que como se ha explicado, hace que algunas pruebas no ofrezcan resultados satisfactorios. También es destacable la rápida convergencia que presentan este modelo de red, que implica por otra parte la vigilancia para evitar el *overfitting*.

La arquitectura AEPP2, aunque ofrece resultados satisfactorios, no llega al nivel de calidad de la U-Net. Como se ha descrito, en ocasiones la reconstrucción de las imágenes no es completamente satisfactoria, incluso a simple vista. Además, la aparición de la anomalía descrita como “efecto cuadrícula” supone un importante problema imposible de ignorar. Con todo ello se puede afirmar que el uso de la arquitectura ARPP2 no es válido para la deconvolución de imágenes de galaxias.

Respecto al modelo DIDN, los resultados son los peores de las tres alternativas, obteniendo imágenes generalmente mal reconstruidas. La pequeña mejora que supone utilizar dos unidades de subprocesamiento hace pensar que la inclusión de un mayor número de ellas permita mejorar estos resultados. Otro aspecto que la red DIDN ha puesto de manifiesto es su lenta convergencia, haciendo necesaria la ejecución de un gran número de ciclos o *epoch* para obtener métricas aceptables. La característica positiva de este modelo es su buen comportamiento frente al *overfitting*, el mejor de las tres arquitecturas probadas.

VII. CONCLUSIONES

La problemática de las aberraciones producidas en la captura de imágenes astronómicas ha motivado el desarrollo de técnicas que permitan su eliminación y ayuden a los astrónomos en sus estudios científicos. La reconstrucción de estas imágenes con algoritmos matemáticos ha dado paso a nuevas técnicas basadas en inteligencia artificial, que están demostrando su validez como alternativa. El uso de aprendizaje automático supervisado con redes neuronales para la deconvolución de imágenes astronómicas supone una importante alternativa frente a los procedimientos usados habitualmente. La posibilidad de reconstruir imágenes a cómo serían realmente sin disponer de la función de dispersión de punto, es similar al proceso de deconvolución ciega, con la ventaja de haber aprendido a realizar este proceso con datos validados previamente.

En este proyecto se han usado capturas del telescopio espacial Hubble con las que se han entrenado distintas configuraciones de tres arquitecturas de redes neuronales. El modelo UNet se posiciona como una opción robusta para la deconvolución eficaz de las imágenes astronómicas. El proceso de compresión y descompresión extrayendo características significativas, a la vez que se restauran las imperfecciones, ha demostrado un buen rendimiento, que, unido a las particularidades de la arquitectura, fundamentalmente la escalabilidad y rápida convergencia, hacen que sirva de gran ayuda a la comunidad científica para el estudio de las imágenes de galaxias libres de aberraciones.

Propuestas de mejora

1.- Aumento del tamaño de dataset de entrenamiento: Una de las limitaciones que se ha tenido es el número de

imágenes con las que entrenar. Aunque el dataset origen contiene más de 118.000 capturas, las utilizadas en las pruebas se han limitado a 9.625. El motivo de esta limitación ha sido la elección de un tamaño de imagen de 128x128 que se ha considerado visualmente representativo, y compatible con las arquitecturas construidas. Sin embargo, el dataset de partida contiene mayoritariamente imágenes más pequeñas, que podrían ser igualmente utilizadas adaptando la red. Lo ideal sería poder usar el mayor número de imágenes de galaxias, escalándolas a un tamaño fijo superior, procesarlas con la red neuronal y reduciéndolas de nuevo. La clave sería estudiar en qué afecta esta operativa al uso de la PSF para la deconvolución de la imagen *target*.

2.- Uso de imágenes reales como *target*: Las imágenes de trabajo se han obtenido de la campaña de observación astronómica *COSMOS*, *The Cosmic Evolution Survey*. Esta misma campaña se ha planificado para hacerse con el nuevo telescopio espacial James Webb, en el estudio *COSMOS-Web*, que va a realizar capturas del mismo sector del firmamento. Este ciclo de capturas ya ha comenzado, pero se completará durante los años 2023 y 2024. Una vez finalizado, puesto que las galaxias capturadas serán las mismas y están identificadas y catalogadas, se podría construir un dataset de entrenamiento formado con imágenes del Hubble e imágenes del Webb que serían el *target* al tener una mayor calidad y resolución. De esta forma las predicciones de nuevas capturas se basarían en un modelo entrenado con información real de mayor calidad.

3.- Uso de redes GAN (*Generative Adversarial Network*). Este tipo de redes neuronales implican un planteamiento distinto para el entrenamiento, generando aproximaciones de la imagen objetivo hasta que esta es considerada válida. Para este caso de uso, al tratarse de un aprendizaje no supervisado, la reconstrucción se realizaría sin ningún tipo de información adicional a la imagen capturada, lo cual evita tener que deconvolucionar previamente para obtener el *target*. Los resultados obtenidos por este tipo de arquitectura se están mostrando sorprendentes y útiles en las áreas donde se experimenta con ellas, y las hace merecedoras de realizar pruebas en la reconstrucción de imágenes astronómicas.

APÉNDICES

Tabla 2 - Número de filtros y parámetros de las redes neuronales entrenadas.

MODELO	Nº FILTROS SUCEсивOS	Nº PARÁMETROS TOTALES
AEEP2	Rama superior: 32, 64, 32 Rama inferior: 32, 64, 128, 256, 128, 64, 32	1.444.674
U-Net 4	4, 8, 16, 32, 64, 32, 16, 8	121.725
U-Net 8	8, 16, 32, 64, 128, 64, 32, 16, 8	485.813
U-Net 16	16, 32, 64, 128, 256, 128, 64, 32, 16	1.941.093
U-Net 32	32, 64, 128, 256, 512, 256, 128, 64, 32	7.760.069
U-Net 64	64, 128, 256, 512, 1024, 512, 256, 128, 64	31.031.685
U-Net 128	128, 256, 512, 1024, 2048, 1024, 512, 256, 128	124.109.573
DIDN16 1DUB	16, (16, 32, 64, 128, 32, 64, 16, 16), 16	1.507.360
DIDN16 2DUB	16, (16, 32, 64, 128, 32, 64, 16, 16) x2, 16	2.005.792

Figura 11 - Resultados de las predicciones de todos los modelos de las tres arquitecturas entrenadas.

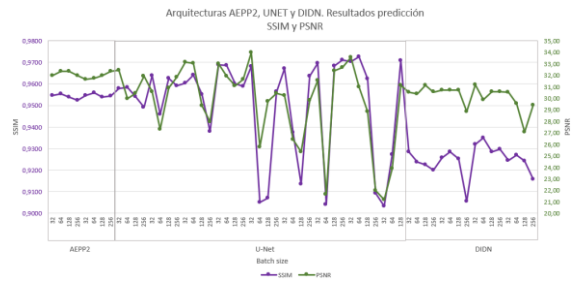
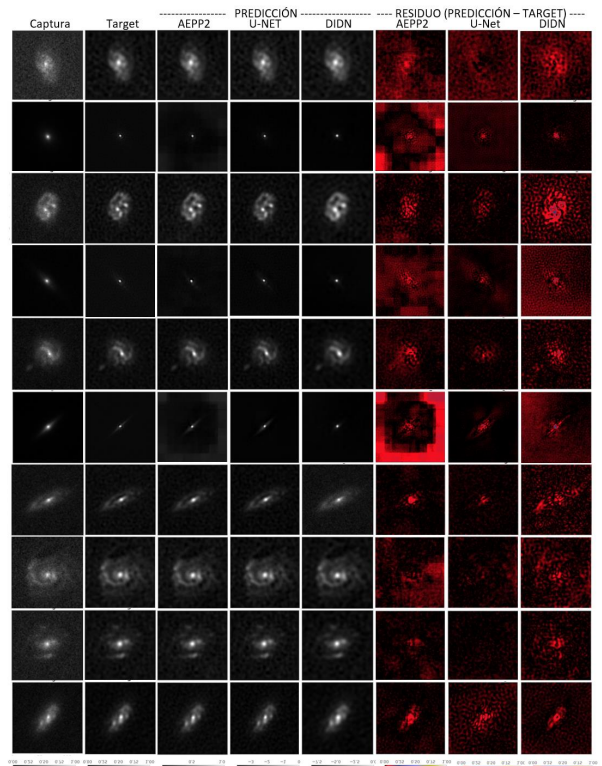


Figura 12 - Ejemplos de predicciones realizadas con los mejores modelos de cada arquitectura. De izquierda a derecha, imagen capturada, *target*, predicción con AEEP2, U-net, DIDN e imagen residual en AEEP2, U-net y DIDN.



REFERENCIAS

- [1] S. W. Hasinoff, «Photon, Poisson Noise», en *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed. Boston, MA: Springer US, 2014, pp. 608-610. doi: 10.1007/978-0-387-31439-6_482.
- [2] V. Sacek, «Point spread function PSF», *Amateur telescope optics*, 14 de julio de 2006. https://www.telescope-optics.net/diffraction_image.htm (accedido 20 de enero de 2023).
- [3] R. Mandelbaum, C. Lackner, A. Leauthaud, y B. Rowe, «COSMOS real galaxy dataset». Zenodo, 1 de enero de 2012. doi: 10.5281/zenodo.3242143.
- [4] N. Scoville *et al.*, «The Cosmic Evolution Survey (COSMOS): Overview», *ApJS*, vol. 172, n.º 1, p. 1, sep. 2007, doi: 10.1086/516585.
- [5] J. Hernández Afonso, «Redes Neuronales Convolucionales para la Reconstrucción de Imágenes de Galaxias», TFM, Universidad Internacional de La Rioja, 2022.
- [6] O. Ronneberger, P. Fischer, y T. Brox, «U-Net: Convolutional Networks for Biomedical Image

- Segmentation», en *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Cham, 2015, pp. 234-241. doi: 10.1007/978-3-319-24574-4_28.
- [7] S. Yu, B. Park, y J. Jeong, «Deep Iterative Down-Up CNN for Image Denoising», en *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, jun. 2019, pp. 2095-2103. doi: 10.1109/CVPRW.2019.00262.
- [8] F. Sureau, A. Lechat, y J.-L. Starck, «Deep learning for a space-variant deconvolution in galaxy surveys», *A&A*, vol. 641, p. A67, sep. 2020, doi: 10.1051/0004-6361/201937039.
- [9] G. Sparacino, G. De Nicolao, G. Pillonetto, y C. Cobelli, «Deconvolution», en *Modelling Methodology for Physiology and Medicine (Second Edition)*, E. Carson y C. Cobelli, Eds. Oxford: Elsevier, 2014, pp. 45-68. doi: 10.1016/B978-0-12-411557-6.00003-3.
- [10] J. L. Starck, E. Pantin, y F. Murtagh, «Deconvolution in Astronomy: A Review», *PASP*, vol. 114, n.º 800, p. 1051, oct. 2002, doi: 10.1086/342606.
- [11] R. Molina, J. Nunez, F. J. Cortijo, y J. Mateos, «Image restoration in astronomy: a Bayesian perspective», *IEEE Signal Processing Magazine*, vol. 18, n.º 2, pp. 11-29, mar. 2001, doi: 10.1109/79.916318.
- [12] S. M. Jefferies, J. C. Christou, S. M. Jefferies, y J. C. Christou, «Restoration of Astronomical Images by Iterative Blind Deconvolution», *ApJ*, vol. 415, p. 862, oct. 1993, doi: 10.1086/173208.
- [13] M. Makarkin y D. Bratashov, «State-of-the-Art Approaches for Image Deconvolution Problems, including Modern Deep Learning Architectures», *Micromachines*, vol. 12, n.º 12, p. 1558, dic. 2021, doi: 10.3390/mi12121558.
- [14] K. O'Shea y R. Nash, «An Introduction to Convolutional Neural Networks», nov. 2015. doi: 10.48550/arXiv.1511.08458.
- [15] I. Goodfellow *et al.*, «Generative adversarial networks», *Commun. ACM*, vol. 63, n.º 11, pp. 139-144, oct. 2020, doi: 10.1145/3422622.
- [16] H. Y. Lee, J. M. Kwak, B. Ban, S. J. Na, S. R. Lee, y H.-K. Lee, «GAN-D: Generative adversarial networks for image deconvolution», en *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, oct. 2017, pp. 132-137. doi: 10.1109/ICTC.2017.8190958.
- [17] X. Mao, C. Shen, y Y.-B. Yang, «Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections», en *Advances in Neural Information Processing Systems*, 2016, vol. 29. Accedido: 8 de febrero de 2023. [En línea]. Disponible en: <https://proceedings.neurips.cc/paper/2016/hash/0ed9422357395a0d4879191c66f4faa2-Abstract.html>
- [18] F. K. Gan, K. Bekki, y A. Hashemizadeh, «SeeingGAN: Galactic image deblurring with deep learning for better morphological classification of galaxies». arXiv, 28 de marzo de 2021. doi: 10.48550/arXiv.2103.09711.
- [19] M. Arjovsky, S. Chintala, y L. Bottou, «Wasserstein Generative Adversarial Networks», *Proceedings of the 34th International Conference on Machine Learning*, pp. 214-223, jul. 2017.
- [20] Z. Ramzi, K. Michalewicz, J.-L. Starck, T. Moreau, y P. Ciuciu, «Wavelets in the Deep Learning Era», *J Math Imaging Vis*, vol. 65, n.º 1, pp. 240-251, ene. 2023, doi: 10.1007/s10851-022-01123-w.
- [21] U. Akhaury, J.-L. Starck, P. Jablonka, F. Courbin, y K. Michalewicz, Eds., «Deep learning-based galaxy image deconvolution», *Frontiers In Astronomy And Space Sciences*, vol. 9, 2022, doi: 10.3389/fspas.2022.1001043.
- [22] Z. Wang, A. C. Bovik, H. R. Sheikh, y E. P. Simoncelli, «Image quality assessment: from error visibility to structural similarity», *IEEE Transactions on Image Processing*, vol. 13, n.º 4, pp. 600-612, abr. 2004, doi: 10.1109/TIP.2003.819861.
- [23] P.-Y. Liu y E. Y. Lam, «Image Reconstruction Using Deep Learning». arXiv, 27 de septiembre de 2018. doi: 10.48550/arXiv.1809.10410.