



UNIVERSIDAD DE VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN

**Inteligencia artificial aplicada al
incremento de resolución en imágenes por
satélite**

Autor:

Juan Carlos López Cerrato

Tutores:

Javier Manuel Aguiar Pérez

María Ángeles Pérez Juárez

Valladolid, abril 2024

Agradecimientos

Quiero expresar mi agradecimiento a mis tutores Javier y María Ángeles por toda la ayuda que, durante estos meses, y pese a la distancia, me han ofrecido para la realización de este Trabajo, sirviendo como guía fundamental en este proceso.

También quiero agradecer a mi gente cercana, a mis padres y hermana y a mis amigos; el apoyo incondicional que me han dado en estos años, siendo la parte más importante de mi vida.

TÍTULO: **Inteligencia artificial aplicada al
incremento de resolución en imágenes
por satélite**

AUTOR: **Juan Carlos López Cerrato**

TUTORES: **Javier Manuel Aguiar Pérez**

María Ángeles Pérez Juárez

DEPARTAMENTO: **Teoría de la Señal y Comunicaciones e
Ingeniería Telemática**

TRIBUNAL

PRESIDENTE: **María Ángeles Pérez Juárez**

SECRETARIO: **Javier Manuel Aguiar Pérez**

VOCAL: **David González Ortega**

SUPLENTE 1: **Míriam Antón Rodríguez**

SUPLENTE 2: **Mario Martínez Zarzuela**

FECHA:

CALIFICACIÓN:

Resumen

En la intersección del progreso tecnológico y la necesidad humana de entender mejor nuestro mundo, se sitúa la superresolución de imágenes como una herramienta revolucionaria, específicamente en el ámbito de la teledetección y la observación satelital. A medida que nos adentramos en una era donde la precisión de la información visual es crucial para aplicaciones tan diversas como la vigilancia medioambiental, la planificación urbana y la gestión de desastres naturales, el desafío de mejorar la calidad de las imágenes capturadas desde el espacio se ha vuelto más apremiante. Este Trabajo de Fin de Grado se centra en la aplicación de técnicas avanzadas de inteligencia artificial, específicamente en la superresolución de imágenes de satélite, para superar las limitaciones inherentes a la resolución espacial de los datos disponibles.

Utilizando un enfoque multidisciplinario que combina modelos de aprendizaje profundo preentrenados con herramientas de superresolución en línea, este estudio busca no solo mejorar la resolución de imágenes de satélite, sino también evaluar de manera crítica la eficacia de estas técnicas a través de una amplia gama de métricas. Al hacerlo, se pretende arrojar luz sobre cómo la inteligencia artificial puede ser aprovechada para extraer información más detallada y precisa de las imágenes de satélite, potenciando así su aplicabilidad en campos críticos para la sociedad.

Este trabajo representa un esfuerzo por cerrar la brecha entre la capacidad tecnológica actual y las crecientes demandas de información visual de alta calidad. Al explorar y validar metodologías de superresolución, se aspira a establecer un marco de referencia que pueda ser utilizado por investigadores y profesionales para mejorar la interpretación de datos satelitales, contribuyendo así a una comprensión más profunda y un manejo más efectivo de los recursos y desafíos globales.

Palabras clave

Deep learning, Imágenes de satélite, Inteligencia Artificial, Métricas de evaluación, Superresolución de imágenes.

Abstract

At the intersection of technological progress and the human need to better understand our world, super-resolution of images emerges as a revolutionary tool, specifically in the field of remote sensing and satellite observation. As we enter an era where the precision of visual information is crucial for diverse applications such as environmental monitoring, urban planning, and natural disaster management, the challenge of enhancing the quality of images captured from space has become more pressing. This Bachelor's Thesis focuses on the application of advanced artificial intelligence techniques, specifically in satellite image super-resolution, to overcome the inherent limitations of the spatial resolution of available data.

Using a multidisciplinary approach that combines pre-trained deep learning models with online super-resolution tools, this study aims not only to improve the resolution of satellite images but also to critically evaluate the effectiveness of these techniques through a wide range of metrics. In doing so, it seeks to shed light on how artificial intelligence can be used to extract more detailed and accurate information from satellite images, thereby enhancing their applicability in critical societal fields.

This work represents an effort to bridge the gap between current technological capability and the growing demands for high-quality visual information. By exploring and validating super-resolution methodologies, the goal is to establish a framework that can be utilized by researchers and professionals to enhance the interpretation of satellite data, contributing to a deeper understanding and more effective management of global resources and challenges.

Keywords

Artificial Intelligence, Deep Learning, Evaluation metrics, Image super-resolution, Satellite Images.

Índice

1.	Introducción	11
1.1.	Motivación.....	11
1.2.	Objetivos.....	11
1.3.	Metodología.....	12
1.4.	Estructura de la memoria	13
2.	Superresolución de imágenes	14
2.1.	Definición de superresolución de imágenes.....	14
2.2.	Historia de la superresolución de imágenes	15
2.2.1.	Los primeros algoritmos matemáticos	15
2.2.2.	Los modelos basados en aprendizaje automático.....	17
2.2.3.	El futuro de la superresolución de imágenes	18
2.3.	Aplicaciones de la superresolución de imágenes.....	18
3.	Arquitecturas para superresolución de imágenes	20
3.1.	Arquitecturas basadas en aprendizaje supervisado	20
3.1.1.	SRCNN (Super-Resolution Convolutional Neural Network).....	20
3.1.2.	VDSR (Very Deep Super-Resolution)	22
3.1.3.	DRRN (Deeply-Recursive Residual Network).....	23
3.2.	Arquitecturas basadas en aprendizaje no supervisado	25
3.2.1.	GAN (Generative Adversarial Network)	25
3.2.2.	SRGAN (Super-Resolution Generative Adversarial Network)	27
3.2.3.	ESRGAN (Enhanced Super-Resolution Generative Adversarial Network)	29
3.3.	Arquitecturas híbridas	30
4.	Métricas para evaluar el rendimiento de la superresolución de imágenes	32
4.1.	Métricas de bajo nivel	32
4.1.1.	PSNR (Peak Signal-to-Noise Ratio).....	32
4.1.2.	SSIM (Structural Similarity Index).....	34
4.1.3.	FSIM (Feature Similarity Index).....	35
4.1.4.	RMSE (Root Mean Squared Error)	36
4.1.5.	MAE (Mean Absolute Error).....	37
4.1.6.	IOU (Intersection Over Union)	38
4.2.	Métricas de alto nivel	40

4.2.1.	LPIPS (Learned Perceptual Image Patch Similarity)	40
4.2.2.	NIQE (Natural Image Quality Evaluator).....	41
4.2.3.	VQI & VIQI.....	42
4.3.	Otras métricas	43
4.3.1.	Preservación de bordes.....	43
4.3.1.1.	BRI (Boundary Response Index)	44
4.3.1.2.	GBRI (Gradient Boundary Response Index) es.....	44
4.3.2.	Reproducción de texturas.....	44
4.3.3.	Mejora en detalles específicos	45
4.3.3.1.	MDI (Mean Directional Index).....	45
4.3.3.2.	EDM (Error Distortion Measure)	45
5.	Casos de uso de superresolución de imágenes	47
5.1.	Tecnologías	48
5.1.1.	Python	48
5.1.2.	PyTorch.....	50
5.1.3.	Numpy	51
5.1.4.	CUDA	53
5.1.5.	OpenCV	54
5.1.6.	Modelos preentrenados.....	54
5.1.7.	Aplicaciones web	56
5.2.	Implementación	56
5.2.1.	ESRGAN.....	57
5.2.2.	Upscale.media.....	60
5.2.3.	Cutout.pro.....	62
5.2.4.	Implementación de métricas	64
5.2.4.1.	Métricas de bajo nivel	65
5.2.4.2.	Métricas de alto nivel	67
5.2.4.3.	Otras métricas.....	68
6.	Conclusiones y líneas futuras	70
6.1.	Conclusiones.....	70
6.2.	Líneas futuras	70
	Referencias	72

Índice de figuras

Figura 1 Esquema del vecino más próximo	15
Figura 2 Esquema de la interpolación bilineal.....	16
Figura 3 Esquema de la interpolación bicúbica	16
Figura 4 Comparación de vecino más próximo, interpolación bilineal e interpolación bicúbica.....	17
Figura 5 Capas de la arquitectura SRCNN.....	21
Figura 6 Capas de la arquitectura VDSR	22
Figura 7 Arquitectura del modelo DRRN.....	24
Figura 8 Comparación de los modelos VDSR (izquierda) y DRRN (derecha)	25
Figura 9 Arquitectura de un modelo GAN	26
Figura 10 Arquitectura de un modelo SRGAN.....	28
Figura 11 Arquitectura de un modelo ESRGAN	30
Figura 12 Funcionamiento del modelo IOU	39
Figura 13 Comando para utilizar la GPU	51
Figura 14 Comando para instalar PyTorch.....	51
Figura 15 Representación de datos con Numpy.....	52
Figura 16 Operaciones matemáticas con Numpy	52
Figura 17 Generación de datos aleatorios con Numpy.....	52
Figura 18 Comando para utilizar la GPU	53
Figura 19 Comando para utilizar la GPU	55
Figura 13 Conjunto de imágenes de satélite de baja resolución	57
Figura 20 Selección de la carpeta con imágenes de baja resolución	57
Figura 21 Iteración sobre la carpeta de baja resolución	57
Figura 22 Cuerpo del código del modelo preentrenado	57
Figura 23 Conjunto de imágenes de baja resolución (izquierda), mejoradas con ESRGAN (derecha)	60
Figura 24 Comparación entre la imagen original y la imagen escalada de forma tradicional (izquierda), y la imagen original y la imagen escalada con inteligencia artificial (derecha) con la opción Enhance Quality activada.....	60
Figura 25 Comparación entre la imagen original y la imagen escalada de forma tradicional (izquierda), y la imagen original y la imagen escalada con inteligencia artificial (derecha) con la opción Enhance Quality desactivada	61

Figura 26 Imagen de baja resolución (abajo), mejorada con <i>upscale.media</i> y la opción <i>enhanced</i> activada (izquierda) y desactivada (derecha)	62
Figura 27 Imagen de baja resolución (abajo), mejorada con <i>cutout.pro</i> en la opción <i>Quality</i> (izquierda) y la opción <i>Fidelity</i> (derecha)	63
Figura 28 Imagen de baja resolución (abajo), mejorada con <i>cutout.pro</i> en la opción <i>Quality</i> (izquierda) y la opción <i>Fidelity</i> (derecha)	63
Figura 29 Imagen de baja resolución (abajo), mejorada con <i>cutout.pro</i> en la opción <i>Quality</i> (izquierda) y la opción <i>Fidelity</i> (derecha)	64
Figura 30 Imagen mejorada con 5 modelos diferentes siguiendo el orden de la Tabla 166	

1.Introducción

En la era digital es indiscutible que el contenido multimedia es cada vez más relevante, y, por tanto, los mecanismos que nos permitan mejorar la calidad de este de la manera más eficiente y precisa posible son cada vez más necesarios. El auge de las Inteligencias Artificiales en los últimos años ha demostrado que también resultan de gran utilidad en el sector de la mejora de imágenes, proporcionando unos resultados más que aceptables a la hora de aumentar la resolución de estas.

1.1. Motivación

Las imágenes pueden presentar deficiencias, causadas por diferentes motivos, como un mal enfoque, una deformación en la lente, o simplemente una degradación de la imagen. El cerebro humano puede completar estas deficiencias utilizando la información visible. Sin embargo, esta tarea matemáticamente es laboriosa y lenta, por lo que se utilizan herramientas automatizadas.

Estas herramientas no siempre proporcionan resultados satisfactorios, y si lo hacen, probablemente sea bajo un cierto número de premisas, o en un número limitado de situaciones, por lo que no son fiables a la hora de utilizarlas de manera genérica o continuada.

Para mejorar los resultados, se ha desarrollado un nuevo sistema de reparación de imágenes basado en inteligencia artificial. Este sistema imita la función del cerebro humano a la hora de completar las deficiencias, “alucinando” los detalles faltantes a la hora de mejorar las imágenes.

La evaluación de los resultados de cualquier sistema de reparación de imágenes puede ser subjetiva o cuantitativa. Depende de la utilidad que se le quiera dar, será más relevante utilizar métricas objetivas para evaluar una característica cuantitativa que se pueda medir, o métricas subjetivas para evaluar el aspecto resultante de la imagen a ojos del ser humano.

1.2. Objetivos

El objetivo principal de este Trabajo es evaluar mediante diferentes métricas, tanto de alto nivel, como de bajo nivel; el rendimiento de diferentes modelos de inteligencia artificial, en concreto un modelo ESRGAN (*Enhanced Super-Resolution Generative Adversarial Networks*) preentrenado, así como modelos en línea; en el campo específico de las imágenes de satélite.

El empleo de estas tecnologías permitirá mejorar tareas como la vigilancia medioambiental, la planificación urbana o la gestión de desastres naturales, entre otras; obteniéndose mejores resultados que si se realizase con otros algoritmos más clásicos. Además, el uso de diferentes modelos, así como de diferentes métricas, nos proporcionará un análisis completo de los resultados.

1.3. Metodología

El primer paso es realizar un trabajo de investigación de la información ya existente sobre el campo de la superresolución de imágenes. Para ello, hay que realizar una búsqueda de documentos científicos oficiales que proporcionen una visión más teórica y precisa de las tecnologías que se van a emplear, así como de foros o páginas de internet para obtener una visión más informal y subjetiva acerca del tema. Para ello se ha investigado sobre inteligencia artificial, redes GAN, *Deep Learning*, superresolución de imágenes, métricas cuantitativas, etc. Posteriormente, es necesario obtener herramientas parecidas a la que se va a implementar, así como los correspondientes estudios que haya sobre ellas, para poder tener en cuenta algunos aspectos importantes como las tecnologías más eficientes para el desarrollo de este tipo de herramientas, de las cuales se hablará más adelante.

Una vez adquiridos todos los conocimientos esenciales para confeccionar el proyecto, es necesario elegir el entorno de trabajo, seleccionar el lenguaje de programación, y elegir el algoritmo adecuado para el funcionamiento de la herramienta. También se deben considerar las principales bibliotecas requeridas durante el desarrollo del código, previamente habiendo revisado las bibliotecas más frecuentemente empleadas en proyectos similares. La implementación se realiza a través de Pycharm, que proporciona una Interfaz Gráfica de Usuario (GUI) que facilita la instalación de bibliotecas y la creación de distintos entornos virtuales de trabajo. La utilización de esta interfaz posibilitará disponer de los diversos entornos de trabajo virtuales. Además de la herramienta principal de restauración de imágenes, será necesario también desarrollar diversas métricas cuantitativas para su evaluación y la instalación de otros proyectos similares para realizar comparaciones. Por lo tanto, para cada una de las tareas mencionadas, es aconsejable crear entornos de trabajo separados. Esto evita posibles incompatibilidades al instalar las bibliotecas necesarias para cada tarea.

El lenguaje de programación seleccionado ha sido Python, ya que es ampliamente utilizado en el desarrollo de proyectos de superresolución de imágenes debido a su compatibilidad con bibliotecas fundamentales como Tensorflow, CV2 o Numpy, las cuales son esenciales para llevar a cabo el desarrollo del modelo y las métricas. Una vez se cuentan con los componentes esenciales para iniciar el desarrollo del modelo, se puede comenzar con su implementación. Después de desarrollar, entrenar, configurar y probar el modelo, se procede a evaluar los resultados obtenidos. La evaluación puede realizarse tanto de manera cualitativa como cuantitativa. Sin embargo, las evaluaciones subjetivas tienden a ser más variables, ya que dependen del criterio individual. El objetivo es obtener valores de evaluación fijos para poder compararlos con los de otras herramientas de manera objetiva. Por lo tanto, es necesario seleccionar métricas relevantes que midan diversas características de las imágenes generadas, como la calidad de la imagen o

las similitudes con la imagen original. Finalmente, una vez definidas las métricas, se lleva a cabo una comparación con las métricas obtenidas al utilizar otras herramientas similares, como herramientas en línea, lo que ayuda a determinar la eficacia de los resultados obtenidos.

1.4. Estructura de la memoria

Esta memoria está dividida en seis capítulos, cuyo contenido se describe brevemente a continuación:

En el [Capítulo 1](#) (Introducción), en el que se encuentra este apartado; se ha realizado la introducción de este Trabajo. En él se han presentado las motivaciones principales para la realización de este Trabajo, los objetivos que se desean cumplir, la metodología que se ha seguido, y la estructura que posee esta memoria.

En el [Capítulo 2](#) (Superresolución de imágenes) se presenta el tema principal de este trabajo, que es la super resolución de imágenes, se da contexto histórico al mismo, recorriendo desde los primeros algoritmos matemáticos hasta el futuro de la superresolución de imágenes, así como su aplicación.

En el [Capítulo 3](#) (Arquitecturas para superresolución de imágenes) se habla de las diferentes arquitecturas por las que se puede optar a la hora de mejorar la resolución de imágenes, diferenciando entre arquitecturas basadas en aprendizaje supervisado y arquitecturas basadas en aprendizaje no supervisado, como lo son las GANs (Generative Adversarial Networks).

En el [Capítulo 4](#) (Métricas para evaluar el rendimiento de la superresolución de imágenes) se explican y desarrollan las métricas que evalúan el rendimiento de la superresolución de imágenes, diferenciando entre métricas de alto y bajo nivel. Estas métricas jugarán un papel clave a la hora de evaluar los resultados obtenidos en este trabajo, sirviendo como eje fundamental, en este ámbito y en muchos otros, a la hora de clasificar los resultados.

En el [Capítulo 5](#) (Casos de uso de superresolución de imágenes) se explicarán y desarrollarán las diferentes tecnologías utilizadas en este trabajo, así como su posterior implementación en la superresolución de imágenes de satélite, comparando los diferentes resultados obtenidos en base a diferentes métricas, también explicadas en el capítulo 4.

Por último, en el [Capítulo 6](#) (Conclusiones y líneas futuras) se termina el trabajo haciendo una conclusión de todo lo presentado en este, así como de los resultados que se han obtenido en la implementación de los modelos. También se presentan las líneas futuras en las que se podría trabajar para mejorar el ámbito de la superresolución de imágenes, así como las métricas de este mismo campo.

2. Superresolución de imágenes

2.1. Definición de superresolución de imágenes

La resolución de una imagen es la cantidad de detalle que contiene dicha imagen. Se mide en píxeles, que son las unidades más pequeñas de una imagen. Una imagen con una resolución más alta tiene más píxeles por unidad de superficie que una imagen con una resolución más baja, lo que significa que contiene más detalle.

La inteligencia artificial (IA) se puede utilizar para aumentar la resolución de las imágenes. Esto se hace utilizando algoritmos de aprendizaje automático para inferir los píxeles faltantes de la imagen.

Hay dos enfoques principales para aumentar la resolución de las imágenes utilizando la IA:

- Superresolución. La superresolución es un enfoque de aprendizaje automático para aumentar la resolución de las imágenes. Funciona generando nuevos píxeles para la imagen basándose en la información de los píxeles existentes. En esta memoria nos vamos a centrar en este enfoque, ya que es el más utilizado para nuestro propósito.

Los modelos de superresolución se entrenan en un conjunto de datos de imágenes de alta resolución y baja resolución. El conjunto de datos de alta resolución se utiliza para enseñar al modelo a reconocer los patrones que se encuentran en las imágenes. El conjunto de datos de baja resolución se utiliza para probar el modelo y evaluar su rendimiento.

Hay dos tipos principales de modelos de superresolución:

- Modelos basados en redes neuronales. Estos modelos son los más comunes y suelen producir los mejores resultados. Las redes neuronales son un tipo de algoritmo de aprendizaje automático que puede aprender a reconocer patrones complejos.
- Modelos basados en difusión. Estos modelos son más rápidos que los modelos basados en redes neuronales, pero suelen producir resultados menos precisos. Los modelos basados en difusión funcionan difuminando una imagen de baja resolución hasta que se parece a una imagen de alta resolución.
- Interpolación. Este enfoque utiliza un algoritmo matemático para calcular los píxeles faltantes de la imagen. El algoritmo se basa en la información de los píxeles existentes en la imagen.

La superresolución tiene una serie de ventajas sobre la interpolación. En primer lugar, la superresolución suele producir resultados más precisos, además, la superresolución puede utilizarse para aumentar la resolución de las imágenes en cualquier dirección, mientras que la interpolación sólo puede utilizarse para aumentar la resolución en una dirección. Aunque una desventaja es que es más exigente en términos de potencia de procesamiento.

En este Trabajo de Fin de Grado nos centraremos en la utilización de la inteligencia artificial para la superresolución de imágenes de satélite.

2.2. Historia de la superresolución de imágenes

No siempre se han utilizado los mismos mecanismos para aumentar la calidad de las imágenes. A medida que el poder de computación y la tecnología avanzaban, han ido surgiendo nuevos métodos, cada vez más precisos y rápidos, para desempeñar estas funciones.

2.2.1. Los primeros algoritmos matemáticos

Los primeros algoritmos matemáticos para la superresolución de imágenes se basaban en la interpolación. La interpolación es un proceso matemático que consiste en estimar valores desconocidos a partir de valores conocidos. En el caso de la superresolución de imágenes, la interpolación se utiliza para estimar los píxeles de alta resolución a partir de los píxeles de baja resolución de la imagen original [1].

Los algoritmos de interpolación más comunes son:

- Vecino más próximo: Es el más simple de todos, los nuevos píxeles (o píxeles de alta resolución) tomarán el valor del píxel más cercano, tal y como se ve en la **Figura 1**. Los resultados obtenidos con esta técnica no son muy precisos.

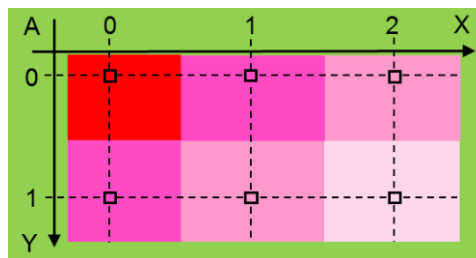


Figura 1 Esquema del vecino más próximo

- Interpolación bilineal: Este algoritmo interpola los píxeles de alta resolución calculando la media del valor de los cuatro píxeles de baja resolución adyacentes, como se puede observar en la **Figura 2**.

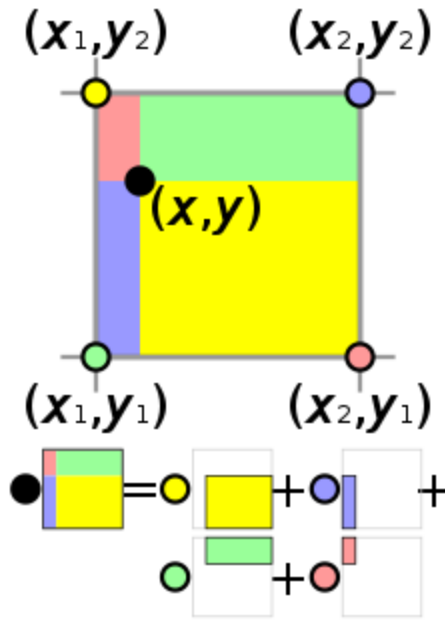


Figura 2 Esquema de la interpolación bilineal

- Interpolación bicúbica: Este algoritmo interpola los píxeles de alta resolución calculando la media de los píxeles de baja resolución adyacentes, así como los píxeles de baja resolución de las cuatro diagonales.

Los pesos utilizados para calcular la media ponderada se basan en la distancia entre el píxel de alta resolución y los píxeles de baja resolución. En la **Figura 3** se representa el píxel de alta resolución (punto verde) y su relación con los demás píxeles de baja resolución.

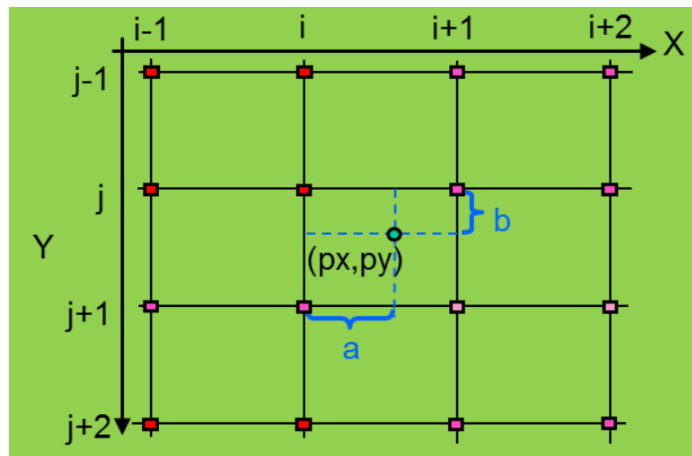


Figura 3 Esquema de la interpolación bicúbica

En este caso, cuanto más complejo es el algoritmo, mejor y más suave será la mejora de calidad, ya que más información de los píxeles adyacentes tendrá en cuenta el algoritmo a la hora de añadir píxeles nuevos a la imagen. Esto se puede observar en la **Figura 4**.

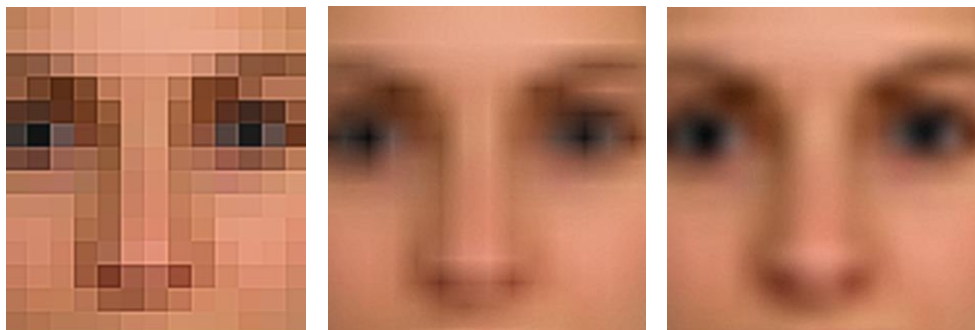


Figura 4 Comparación de vecino más próximo, interpolación bilineal e interpolación bicúbica

Estos algoritmos son sencillos y fáciles de implementar, pero tienen el inconveniente de que pueden introducir artefactos visuales en la imagen mejorada, como bordes borrosos o *moiré*, un efecto visual que se produce cuando se superponen dos patrones de líneas que tienen una frecuencia similar, apareciendo bandas o patrones de interferencia.

2.2.2. Los modelos basados en aprendizaje automático

A principios de la década de 2000, los investigadores comenzaron a explorar el uso del aprendizaje automático para la superresolución de imágenes. El aprendizaje automático es una rama de la inteligencia artificial que permite a las máquinas aprender de los datos sin ser explícitamente programadas.

Los modelos de aprendizaje automático para la superresolución de imágenes se basan en la idea de que una red neuronal puede aprender a generar píxeles de alta resolución a partir de píxeles de baja resolución. Estos modelos suelen estar entrenados en un conjunto de datos de imágenes de baja y alta resolución.

Los modelos de aprendizaje automático para la superresolución de imágenes han demostrado ser mucho más eficaces que los algoritmos matemáticos tradicionales. Pueden generar imágenes mejoradas (imágenes que nacen de aplicar la superresolución a la imagen de baja resolución del par de imágenes de alta y baja resolución) con una calidad comparable a las imágenes originales.

Algunos de los modelos de aprendizaje automático para la superresolución de imágenes más populares son:

- SRCNN (*Super-Resolution Convolutional Neural Network*): Es uno de los primeros modelos de aprendizaje automático que demostró ser eficaz para la superresolución de imágenes.

SRCNN ha demostrado ser eficaz para la superresolución de imágenes en una variedad de aplicaciones, como la restauración de imágenes antiguas, la mejora de la calidad de las imágenes médicas y la creación de imágenes de alta resolución a partir de imágenes de baja resolución.

- eSRGAN (Enhanced Super-Resolution Generative Adversarial Network): Este modelo fue propuesto por Wang et al. en 2018. Es un modelo de red neuronal basado en una red generativa adversaria.

Estos modelos han sido utilizados en una amplia variedad de aplicaciones, concretamente, este trabajo se centrará en la mejora de imágenes de satélite utilizando modelos basados en aprendizaje automático [2].

2.2.3. El futuro de la superresolución de imágenes

La superresolución de imágenes está en constante evolución. A medida que los modelos de aprendizaje automático se vuelven más complejos y potentes, se espera que la calidad de las imágenes mejoradas siga incrementando.

En el futuro, la superresolución de imágenes podría utilizarse en una amplia variedad de aplicaciones, como:

- La restauración de imágenes antiguas: La superresolución de imágenes podría utilizarse para restaurar imágenes antiguas que han perdido su resolución original.
- La mejora de la calidad de las imágenes médicas: La superresolución de imágenes podría utilizarse para mejorar la calidad de las imágenes médicas, lo que podría facilitar el diagnóstico de enfermedades.
- La creación de imágenes de alta resolución a partir de imágenes de baja resolución: La superresolución de imágenes podría utilizarse para crear imágenes de alta resolución a partir de imágenes de baja resolución, lo que podría utilizarse en aplicaciones como la realidad virtual y la realidad aumentada.

2.3. Aplicaciones de la superresolución de imágenes

La superresolución de imágenes ha emergido como una disciplina fascinante en el campo de la visión por computadora, ofreciendo la posibilidad de mejorar la calidad y resolución de imágenes de manera significativa. Esta técnica se ha vuelto esencial en diversas aplicaciones, abriendo nuevas perspectivas en campos tan diversos como la medicina, la investigación científica, la vigilancia, y la fotografía digital [3].

- **Aplicaciones en Medicina:** En el ámbito médico, la superresolución de imágenes desempeña un papel crucial al permitir la obtención de detalles finos en imágenes médicas, como resonancias magnéticas y tomografías computarizadas. Esto facilita diagnósticos más precisos y contribuye al avance de la investigación biomédica al proporcionar imágenes más nítidas y detalladas de estructuras biológicas a

diferentes escalas. Aunque, para ello, la mejora de la calidad tiene que ser precisa y acertada.

- **Investigación Científica:** En el ámbito de la investigación científica, la superresolución de imágenes se ha convertido en una herramienta invaluable para estudiar fenómenos a nivel microscópico. Permite a los científicos obtener imágenes detalladas de muestras biológicas o materiales a una escala que antes era inalcanzable, facilitando así la observación y análisis de estructuras complejas.
- **Vigilancia y Seguridad:** La aplicación de superresolución en sistemas de vigilancia y seguridad es esencial para mejorar la capacidad de reconocimiento facial, identificación de matrículas y detalles cruciales en entornos de monitoreo.
- **Fotografía Digital:** En el ámbito de la fotografía digital, la superresolución permite a los fotógrafos mejorar la calidad de sus imágenes, especialmente en situaciones de baja iluminación o con limitaciones de hardware. Esto amplía las posibilidades creativas al ofrecer la capacidad de ampliar y recortar imágenes sin sacrificar la claridad y la nitidez.

Las imágenes de satélite son relevantes en varias de ellas. En la investigación científica se necesitan imágenes aéreas bien definidas a la hora de evaluar ecosistemas, diferentes sucesos climatológicos, o incluso distinguir diferentes tipos de especies a la hora de trazar su ruta migratoria. También en temas de vigilancia o seguridad se puede requerir una imagen nítida a la hora de inspeccionar un lugar concreto con más detalle.

La superresolución de imágenes ha demostrado ser una herramienta versátil con aplicaciones significativas en diversas disciplinas. Su capacidad para mejorar la calidad de las imágenes, revelando detalles previamente imperceptibles, promete impulsar avances significativos en múltiples campos. A medida que la tecnología continúa evolucionando, se espera que la superresolución desempeñe un papel cada vez más prominente en la optimización y enriquecimiento de la representación visual de numerosas maneras.

3.Arquitecturas para superresolución de imágenes

En el ámbito de la superresolución de imágenes, se han desarrollado diversas arquitecturas que buscan mejorar la resolución de las imágenes de manera efectiva.

Entre las arquitecturas más destacadas se encuentran las basadas en redes neuronales convolucionales (CNN). Estas redes se caracterizan por su capacidad para aprender patrones y características de alto nivel en los datos de entrada.

Otro enfoque relevante es el uso de arquitecturas de redes generativas adversarias (GAN), que introducen un componente de antagonismo en el proceso de generación de imágenes.

En este contexto, la elección de la arquitectura adecuada para la superresolución de imágenes depende de diversos factores, como el tipo de imágenes, los recursos disponibles y los requisitos específicos de la aplicación.

3.1. Arquitecturas basadas en aprendizaje supervisado

En el aprendizaje supervisado, el modelo se entrena utilizando pares de imágenes de baja resolución y alta resolución. Cada imagen de baja resolución tiene una contraparte correspondiente de alta resolución, y el modelo aprende a mapear de una a otra durante el proceso de entrenamiento.

Durante el entrenamiento, se utiliza una función de pérdida que mide la discrepancia entre la imagen mejorada generada por el modelo y la imagen de alta resolución real. El objetivo es minimizar esta pérdida, lo que impulsa al modelo a aprender a generar imágenes mejoradas que se asemejen lo más posible a las imágenes de referencia de alta resolución.

Algunos de los modelos que se basan en este tipo de arquitectura, y que se desarrollarán a continuación, son el SRCNN, VDSR o el DRRN.

3.1.1. SRCNN (Super-Resolution Convolutional Neural Network)

El SRCNN (*Super-Resolution Convolutional Neural Network*) es un modelo de red neuronal convolucional propuesto por Chao Dong et al. en 2014 [\[4\]](#). Es uno de los primeros modelos de aprendizaje automático que demostró ser eficaz para la superresolución de imágenes.

SRCNN consta de tres capas convolucionales, seguidas de una capa de *fully connected* (**Figura 5**). Las capas convolucionales se utilizan para extraer

características de la imagen de baja resolución. La capa de *fully connected* se utiliza para generar la imagen de alta resolución.

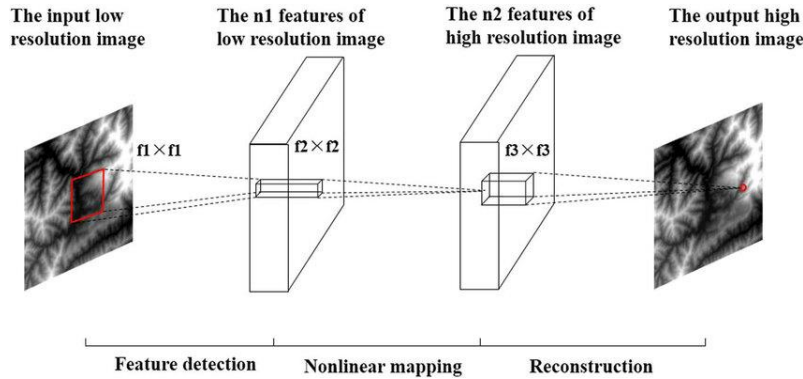


Figura 5 Capas de la arquitectura SRCNN

Las tres etapas son:

- Etapas de Extracción de Características (Conv1):
 - Convoluciones: La primera capa convolucional aplica filtros para extraer características de baja y media escala de la imagen de entrada.
 - Función de activación (Ecuación 1): Se utiliza una función de activación no lineal, típicamente ReLU (*Rectified Linear Unit*), para introducir no linealidades en la red y mejorar la capacidad de aprendizaje.

$$F_1(Y) = \max(0, W_1 * Y + B_1) \quad (1)$$

- Etapas de Mapeo de Características (Conv2):
 - Convoluciones: En esta etapa, se realizan más capas convolucionales para mapear las características extraídas a representaciones de mayor nivel y complejidad.
 - Función de activación (Ecuación 2): Se aplica nuevamente una función de activación, proporcionando no linealidades adicionales.

$$F_2(Y) = \max(0, W_2 * F_1(Y) + B_2) \quad (2)$$

- Etapas de Reconstrucción (Conv3):
 - Convoluciones: La última capa convolucional genera la imagen mejorada final al combinar las representaciones de características aprendidas en las etapas anteriores.
 - No hay función de activación después de esta capa, ya que se busca generar píxeles de salida reales con la ecuación 3.

$$F_3(Y) = W_3 * F_2(Y) + B_3 \quad (3)$$

El entrenamiento de SRCNN se realiza utilizando un conjunto de datos de imágenes de baja y alta resolución. El objetivo del entrenamiento es aprender los pesos de las capas convolucionales de manera que la imagen mejorada sea lo más similar posible a la imagen de alta resolución original.

Es importante destacar que, aunque SRCNN fue una contribución significativa, han surgido otras arquitecturas más complejas y sofisticadas con el tiempo, pero SRCNN estableció las bases para la aplicación exitosa de redes neuronales en la tarea de superresolución de imágenes.

3.1.2. VDSR (Very Deep Super-Resolution)

La arquitectura VDSR (*Very Deep Super-Resolution*) es otra propuesta destacada en el campo de la superresolución de imágenes, diseñada para abordar la mejora de la resolución a través de redes neuronales profundas. Desarrollada por Kim et al. en 2016 [5], VDSR se centra en la utilización de una red profunda para aprender representaciones más complejas y detalladas de las imágenes de baja resolución.

Al igual que SRCNN es un modelo de red neuronal convolucional (CNN), que utiliza capas de convolución y submuestreo para aprender a generar nuevos píxeles para una imagen de baja resolución.

La arquitectura VDSR se compone de 20 capas de convolución (en contraste con las 3 de SRCNN), cada una seguida de una capa de submuestreo. El submuestreo reduce la resolución de la imagen a la mitad, lo que obliga al modelo a aprender información más abstracta. Las capas de convolución luego restauran la resolución a su tamaño original (Figura 6).

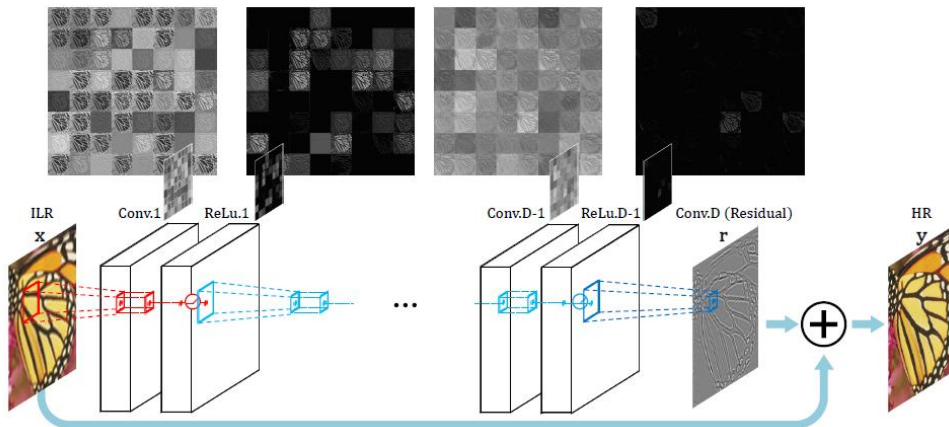


Figura 6 Capas de la arquitectura VDSR

La arquitectura VDSR:

- Capas de convolución: Las capas de convolución son la unidad básica de una CNN. Se utilizan para extraer características de una imagen. En el caso de VDSR, cada capa de convolución tiene 64 filtros.
- Capas de submuestreo: Las capas de submuestreo se utilizan para reducir la resolución de una imagen a la mitad. Esto se hace mediante un proceso llamado convolución de aplanamiento, que convierte una imagen en una matriz plana de números. Luego, esta matriz se divide por dos en cada dimensión.
- Capas de expansión: Las capas de expansión se utilizan para restaurar la resolución de una imagen. Esto se hace mediante un proceso llamado convolución de expansión, que convierte una matriz plana en una imagen.

La arquitectura VDSR se ha utilizado como base para muchos otros modelos de superresolución. Estos modelos han mejorado la arquitectura VDSR en una variedad de formas, incluyendo:

- Utilizar más capas de convolución: Los modelos más recientes utilizan más capas de convolución que VDSR. Esto permite al modelo aprender más información sobre la imagen y producir resultados de mejor calidad.
- Utilizar redes neuronales convolucionales residuales: Las redes neuronales residuales son un tipo de CNN que permite al modelo aprender a saltar sobre ciertas capas de la red. Esto puede ayudar a evitar el problema del desvanecimiento del gradiente, que puede hacer que los modelos de aprendizaje profundo sean difíciles de entrenar.
- Utilizar redes neuronales convolucionales atencionales: Las redes neuronales convolucionales atencionales son un tipo de CNN que permite al modelo aprender a enfocarse en partes específicas de la imagen. Esto puede ayudar a mejorar la precisión del modelo.

La arquitectura VDSR es un modelo de aprendizaje profundo importante que ha hecho contribuciones significativas al campo de la superresolución de imágenes. Su arquitectura sencilla y eficaz la ha convertido en una base popular para otros modelos de superresolución, y sus resultados superiores la han convertido en una referencia importante en el campo.

3.1.3. DRRN (Deeply-Recursive Residual Network)

DRRN (*Deeply-Recursive Residual Network*) es un modelo de aprendizaje profundo desarrollado en 2017 por un equipo liderado por Jianchao [6]. Es una extensión de la arquitectura VDSR que utiliza una estructura recursiva para mejorar la capacidad de aprendizaje del modelo (**Figura 7**).

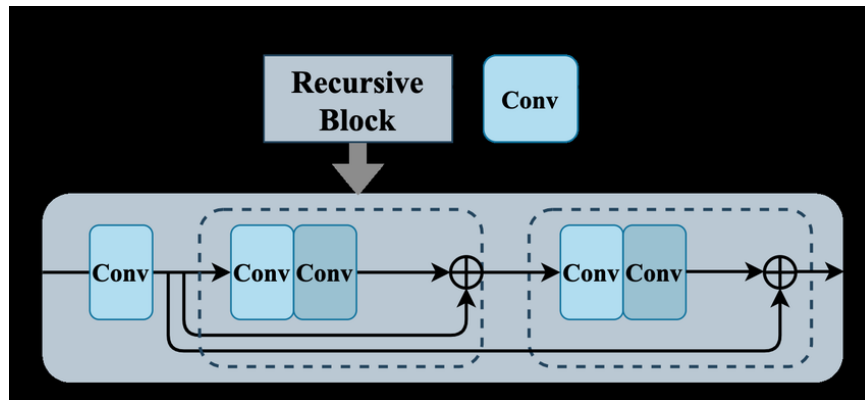


Figura 7 Arquitectura del modelo DRRN

A continuación se presenta una descripción más detallada de la arquitectura DRRN:

- Módulos de decodificación: Son similares a los módulos de convolución y submuestreo utilizados en la arquitectura VDSR. Los módulos de decodificación son responsables de restaurar la resolución de la imagen. Cada módulo de decodificación consta de una capa de convolución, una capa de submuestreo y una capa de expansión.
- Módulos de recursividad: Los módulos de recursividad son responsables de recuperar información de las capas más profundas de la red. Cada módulo de recursividad consta de dos capas de convolución, una capa de submuestreo y una capa de expansión.

Al igual que pasaba con el modelo VDSR, la arquitectura DRRN ha sido utilizada como base para muchos otros modelos de superresolución, los cuales:

- Utilizan más módulos recursivos: Esto permite al modelo recuperar más información de las capas más profundas de la red y producir resultados de mejor calidad.
- Utilizan redes neuronales convolucionales de profundidad variable: Esto puede ayudar a mejorar la calidad del modelo en áreas con diferentes niveles de detalle.
- Utilizan redes neuronales convolucionales atencionales: Esto puede ayudar a mejorar la precisión del modelo.

En la **Figura 8** se puede observar la diferencia en las capas del modelo VDSR con respecto a las del modelo DRRN, en el que no solo se suceden simples capas de convolución, si no que tenemos módulos de decodificación seguidos de los módulos de recursividad:

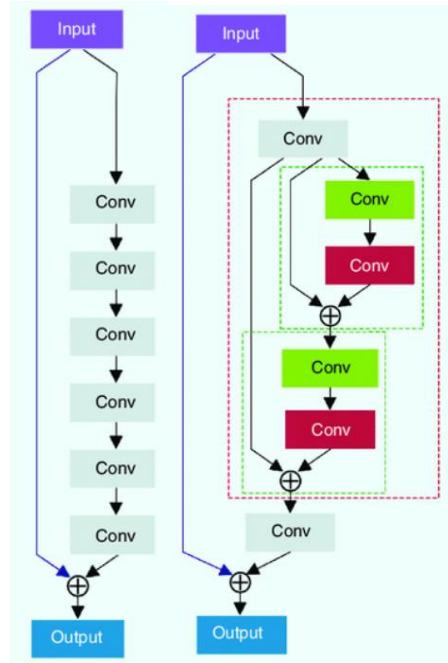


Figura 8 Comparación de los modelos VDSR (izquierda) y DRRN (derecha)

3.2. Arquitecturas basadas en aprendizaje no supervisado

En el aprendizaje no supervisado, el modelo no se entrena con pares de imágenes de baja y alta resolución. En cambio, el modelo busca aprender patrones y características únicas en los datos sin la necesidad de información específica sobre la relación entre las imágenes de entrada y salida.

Los modelos no supervisados pueden utilizar estrategias como redes generativas adversarias (GAN) o autoencoders para aprender representaciones de características y generar imágenes mejoradas. En este caso, el modelo aprende de manera autónoma, explorando la estructura y las características intrínsecas de los datos.

En este trabajo se explicarán los GANs. Estos modelos tienden a ser más flexibles en términos de adaptabilidad a diferentes conjuntos de datos, pero pueden requerir una mayor complejidad en su entrenamiento y ajuste.

3.2.1. GAN (Generative Adversarial Network)

GAN (*Generative Adversarial Network*) es un tipo de modelo de aprendizaje no supervisado que consiste en dos redes neuronales, el generador y el discriminador, que se entrenan simultáneamente a través de un proceso adversarial. Fue propuesto por Ian Goodfellow y su equipo en 2014 [7].

Componentes de un modelo GAN (**Figura 9**):

- Generador: El generador toma como entrada un vector de números aleatorios (también llamado vector latente) y genera datos, como imágenes, a partir de ese vector. Su objetivo es aprender a producir datos que sean indistinguibles de los datos reales.
- Discriminador: El discriminador actúa como "juez" evaluando la autenticidad de un conjunto de datos, determinando si proviene del generador (datos generados) o del conjunto de datos real. Su tarea es aprender a distinguir entre los datos reales y los generados por el generador.

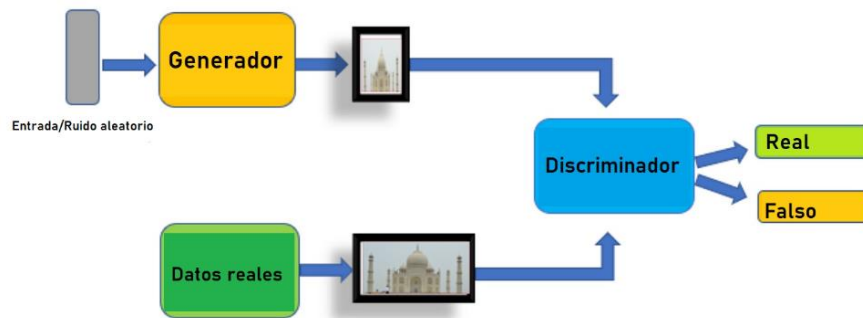


Figura 9 Arquitectura de un modelo GAN

El proceso de entrenamiento de un modelo GAN es el siguiente:

1. Generación de Datos: El generador produce datos sintéticos a partir de vectores latentes.
2. Etiquetado de Datos: Se etiquetan los datos generados como "falsos" y los datos reales del conjunto de entrenamiento como "verdaderos".
3. Evaluación por el Discriminador: El discriminador evalúa tanto los datos reales como los generados, asignándoles probabilidades de ser "verdaderos" o "falsos".
4. Cálculo de Pérdida: Se calcula la pérdida basada en la discrepancia entre las predicciones del discriminador y las etiquetas reales.
5. Retroalimentación al Generador: La pérdida calculada se utiliza para retroalimentar al generador, informándole sobre cómo mejorar para producir datos más auténticos.
6. Actualización del Discriminador: El discriminador también se actualiza utilizando la retroalimentación de la evaluación de datos reales y generados.
7. Repetición del Proceso: Este proceso se repite en ciclos, con el generador y el discriminador mejorando continuamente en sus respectivas tareas hasta alcanzar un equilibrio.

Con el tiempo, el generador mejora su capacidad para generar datos que son cada vez más difíciles de distinguir de los datos reales, y el discriminador se vuelve más hábil para realizar la distinción. Esto lleva a la creación de datos sintéticos de alta calidad.

Los GAN son aplicados en diversas tareas, como la generación de imágenes realistas, la traducción de estilo de imágenes, la superresolución de imágenes y la generación de datos sintéticos para tareas de entrenamiento en aprendizaje profundo. Sin embargo, la capacitación de GANs puede ser delicada y requerir ajustes cuidadosos para evitar problemas como el colapso del generador o la inestabilidad en el entrenamiento.

3.2.2. SRGAN (Super-Resolution Generative Adversarial Network)

SRGAN (*Super-Resolution Generative Adversarial Network*) es una arquitectura específicamente diseñada para abordar la superresolución de imágenes mediante el uso de redes generativas adversarias (GAN). Fue propuesto por Christian Ledig y su equipo en 2017 [8], y ha demostrado ser eficaz en la generación de imágenes mejoradas y de alta calidad.

Componentes del modelo SRGAN (Figura 10):

- **Generador:** El generador en SRGAN toma una imagen de baja resolución como entrada y busca generar una versión mejorada de alta calidad. A diferencia de los generadores en GANs tradicionales, el generador de SRGAN incorpora capas de mejora de la resolución para producir detalles finos.
- **Discriminador:** Similar a otros GANs, el discriminador en SRGAN evalúa la autenticidad de las imágenes mejoradas generadas por el generador en comparación con las imágenes de alta resolución reales. Sin embargo, en SRGAN, el discriminador se centra en evaluar la calidad perceptual en lugar de simplemente distinguir entre imágenes reales y generadas.
- **Función de Pérdida:** SRGAN utiliza una función de pérdida específica llamada Perceptual Loss (pérdida perceptual), que combina la diferencia de contenido (similar a la pérdida de contenido en VGG) y una pérdida adversarial. La pérdida perceptual se basa en la idea de que la calidad visual percibida por los humanos es más importante que la coincidencia píxel a píxel.
- **Capas de Normalización Espectral y Capas Residuales:** SRGAN incorpora capas de normalización espectral en el discriminador y utiliza bloques residuales en el generador. Estos elementos ayudan a estabilizar y acelerar el proceso de entrenamiento.

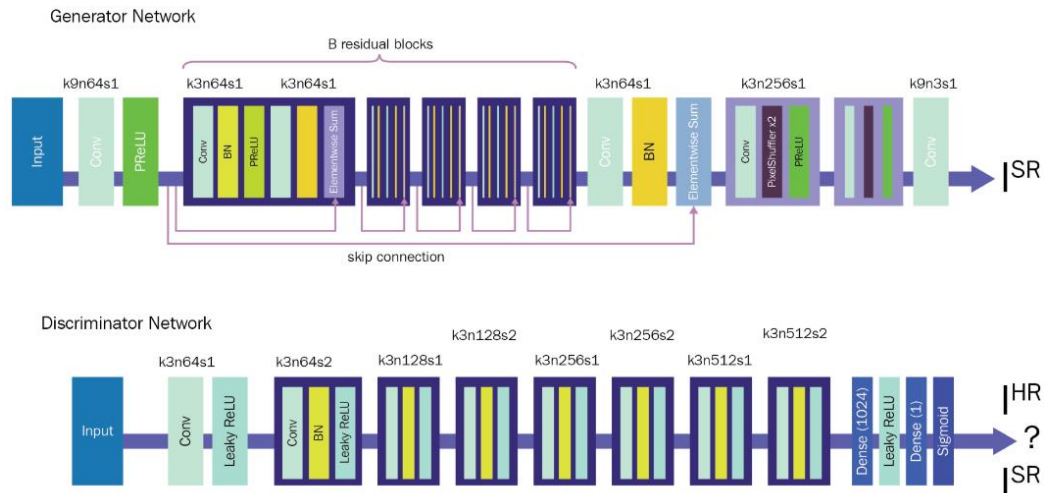


Figura 10 Arquitectura de un modelo SRGAN

Proceso de Entrenamiento:

1. Generación de imágenes mejoradas: Como en los GANs, el generador toma imágenes de baja resolución y genera imágenes de alta calidad.
2. Evaluación por el Discriminador: Como en los GANs, el discriminador evalúa la autenticidad y calidad perceptual de las imágenes recibidas del generador.
3. Cálculo de Pérdida: La pérdida perceptual se calcula mediante la comparación de las imágenes de alta calidad generadas con las imágenes de alta resolución reales.
4. Retroalimentación y Optimización: Se retroalimenta al generador utilizando la pérdida perceptual y se optimiza a través de técnicas de retropropagación. Este proceso se repite iterativamente para mejorar la capacidad del generador.

Después de repetir el proceso las suficientes veces, el SRGAN será capaz de lograr generar imágenes mejoradas que no solo son más nítidas, sino que también conservan mejor los detalles estructurales y texturales. La aplicación de la pérdida perceptual contribuye significativamente a la mejora de la calidad visual percibida.

En resumen, SRGAN es una arquitectura GAN diseñada específicamente para abordar la superresolución de imágenes, centrándose en la mejora de la calidad perceptual de las imágenes mejoradas generadas. Este enfoque ha llevado a mejoras significativas en la generación de imágenes de alta resolución y ha influido en desarrollos posteriores en el campo de la superresolución con inteligencia artificial.

3.2.3. ESRGAN (Enhanced Super-Resolution Generative Adversarial Network)

ESRGAN (*Enhanced Super-Resolution Generative Adversarial Network*) es un tipo de red neuronal generativa adversarial (GAN) al igual que SRGAN, diseñada para realizar la tarea de superresolución de imágenes, fue propuesto por Xintao Wang y su equipo en 2018 [9]. La principal diferencia radica en las mejoras realizadas en ESRGAN para abordar ciertos problemas y mejorar la calidad de las imágenes generadas.

Los componentes del ESRGAN son estructuralmente los mismos que los del SRGAN (**Figura 11**), con pequeñas diferencias entre cada uno de ellos:

- Arquitectura del Generador:
 - SRGAN: Utiliza una arquitectura generativa adversarial convencional.
 - ESRGAN: Introduce mejoras en la arquitectura del generador, incorporando capas adicionales y ajustes para mejorar la calidad de las imágenes mejoradas.
- Capacidad de Generación de Detalles:
 - SRGAN: Aunque SRGAN es capaz de mejorar la resolución de las imágenes, a veces puede generar imágenes con artefactos o falta de detalles finos.
 - ESRGAN: Se diseñó específicamente para abordar las limitaciones de SRGAN en términos de detalles finos. ESRGAN utiliza una arquitectura más sofisticada y ha demostrado ser más eficaz en la generación de detalles realistas en las imágenes mejoradas.
- Perceptual Loss:
 - SRGAN: Emplea una función de pérdida perceptual basada en características extraídas de una red pre-entrenada, como la VGG (Visual Geometry Group) en el caso original.
 - ESRGAN: Introduce el concepto de pérdida perceptual mejorada, que utiliza la red VGG junto con una red pre-entrenada llamada ESRGAN para una representación más precisa y perceptualmente relevante.

- Capacidades Adversariales:
 - SRGAN: Utiliza una red adversarial para evaluar la calidad de las imágenes mejoradas.
 - ESRGAN: Mantiene el enfoque adversarial, pero mejora la arquitectura tanto del generador como del discriminador para obtener resultados más realistas.

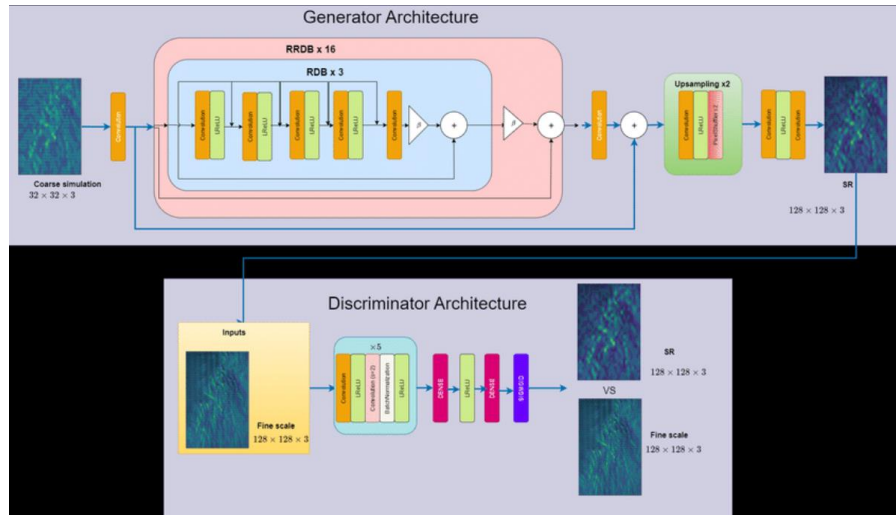


Figura 11 Arquitectura de un modelo ESRGAN

El entrenamiento de ESRGAN implica iteraciones repetidas de este proceso hasta que el generador pueda producir imágenes de alta resolución que sean visualmente indistinguibles de las imágenes reales.

En general, ESRGAN se considera una evolución y mejora significativa con respecto a SRGAN, ya que aborda deficiencias específicas y logra resultados superiores en términos de calidad visual y detalles en las imágenes mejoradas. La introducción de pérdida perceptual mejorada y ajustes en la arquitectura son aspectos clave que han contribuido a su éxito.

3.3. Arquitecturas híbridas

Las arquitecturas híbridas en la superresolución de imágenes mediante inteligencia artificial combinan enfoques tradicionales de procesamiento de imágenes con técnicas de aprendizaje profundo [10], de hecho; en algunos de los modelos ya explicados, como el ESRGAN; se combinan ambos métodos para optimizar los resultados, aunque sean esencialmente modelos de aprendizaje profundo, como en el caso del ESRGAN. Estas fusiones buscan aprovechar las fortalezas de ambas metodologías para mejorar la calidad y la eficacia del proceso de superresolución. Algunos aspectos y enfoques comunes en estas arquitecturas híbridas incluyen:

- Enfoques de Bajo y Alto Nivel: Utilizan características de bajo nivel, como bordes y texturas, extraídas a través de técnicas de procesamiento de imágenes tradicionales. Combinan estas características con representaciones de alto nivel aprendidas por modelos de aprendizaje profundo.
- Modelos de Redes Neuronales Pre-entrenadas: Incorporan modelos de aprendizaje profundo pre-entrenados, como VGG, para capturar características de alto nivel en las imágenes.
- Convoluciones Espaciales y Temporales: Aplican convoluciones tanto espaciales como temporales para capturar patrones locales y globales en las imágenes.
- Uso de Redes Generativas Adversariales (GAN): Integran GAN para aumentar la calidad perceptual de las imágenes mejoradas al generar detalles realistas.
- Fusión de Múltiples Modelos: Combina múltiples modelos, cada uno especializado en ciertos aspectos de la superresolución (por ejemplo, un modelo para bordes y otro para texturas).
- Regularización y Restricciones: Implementan técnicas de regularización y restricciones para evitar artefactos no deseados y mantener la coherencia visual en las imágenes mejoradas.
- Optimización Multiobjetivo: Consideran múltiples objetivos de optimización, como minimizar la pérdida de píxeles, maximizar la similitud estructural, y satisfacer criterios perceptuales para obtener resultados equilibrados.
- Aprovechamiento de Información de Escala: Utilizan información de múltiples escalas de la imagen para guiar el proceso de superresolución.

Estas arquitecturas híbridas buscan abordar desafíos específicos en la superresolución, como la generación de detalles finos, la preservación de bordes y texturas, y la mejora de la calidad perceptual. Al combinar métodos tradicionales con enfoques basados en aprendizaje profundo, buscan alcanzar un equilibrio entre la eficacia y la fidelidad visual en las imágenes mejoradas.

4. Métricas para evaluar el rendimiento de la superresolución de imágenes

Las métricas para evaluar el rendimiento de la superresolución de imágenes se utilizan para cuantificar la calidad de las imágenes mejoradas [11]. Estas métricas pueden ayudar a comparar diferentes algoritmos de superresolución y a identificar los que producen los mejores resultados en cada caso.

Se puede diferenciar entre métricas de bajo y alto nivel según la complejidad y abstracción de la información que evalúan. En términos generales, las métricas de bajo nivel tienden a centrarse en detalles específicos de píxeles, mientras que las métricas de alto nivel buscan evaluar la percepción global y la similitud estructural. La elección de la métrica dependerá de los objetivos específicos de la tarea y de la información que se desee evaluar.

4.1. Métricas de bajo nivel

Las métricas de bajo nivel se basan principalmente en comparaciones directas entre los valores de píxeles o en la estadística de píxeles entre las imágenes originales y las imágenes mejoradas. Estas métricas tienden a enfocarse en detalles específicos a nivel de píxeles, sin considerar la percepción visual humana o la estructura de alto nivel en la imagen.

La elección de la métrica adecuada para evaluar el rendimiento de la superresolución de imágenes depende de la aplicación específica. Por ejemplo, si la aplicación requiere imágenes con una alta resolución, entonces se puede utilizar una métrica como el PSNR o el SSIM. Si la aplicación requiere imágenes con una buena calidad visual, entonces se puede utilizar una métrica como el RMSE o el MAE. A continuación se explicarán algunas de las diferentes métricas de bajo nivel.

4.1.1. PSNR (Peak Signal-to-Noise Ratio)

El PSNR (*Peak Signal-to-Noise Ratio*) es una métrica ampliamente utilizada para medir la calidad de una imagen mejorada en comparación con una imagen de referencia original. Se basa en la relación entre la señal máxima posible y la potencia del ruido en la imagen [12].

La fórmula general del PSNR se muestra en la ecuación 4:

$$PSNR = 10 * \log_{10} \left(\frac{\text{Máximo valor de Píxel}^2}{MSE} \right) \quad (4)$$

Donde:

- **Máximo valor de Píxel:** Es el valor máximo que puede tener un píxel en la imagen (por ejemplo, 255 para imágenes en escala de grises o 1 para imágenes en escala de grises normalizadas y para cada canal de color en imágenes a color).
- **MSE (Ecuación 5):** Es el Error Cuadrático Medio, que se calcula como la suma de los cuadrados de las diferencias píxel a píxel entre la imagen mejorada (K) y la imagen de referencia (I), dividido por el número total de píxeles (M*N):

$$MSE = \frac{1}{M*N} \sum_{i=1}^M \sum_{j=1}^N (I_{ij} - K_{ij})^2 \quad (5)$$

El PSNR se expresa en decibelios (dB) y proporciona una medida de la relación entre la señal (la información de la imagen original) y el ruido (las diferencias entre la imagen original y la mejorada). Cuanto mayor sea el PSNR, menor será la diferencia entre las imágenes y, por lo tanto, mayor será la calidad percibida.

Es importante tener en cuenta algunas consideraciones sobre el PSNR:

- Sensibilidad a pequeñas diferencias: El PSNR es sensible a pequeñas diferencias entre las imágenes, y una pequeña variación en los píxeles puede tener un impacto significativo en el valor del PSNR.
- No refleja bien la percepción humana: Aunque el PSNR proporciona una medida cuantitativa de la calidad, no siempre se correlaciona bien con la percepción humana. Por ejemplo, dos imágenes con el mismo PSNR pueden ser percibidas de manera diferente por los observadores humanos.
- Linealidad de la escala: El PSNR se expresa en una escala logarítmica, lo que significa que mejoras pequeñas en el PSNR representan mejoras exponenciales en la calidad percibida.

A pesar de sus limitaciones, el PSNR sigue siendo ampliamente utilizado debido a su simplicidad y facilidad de cálculo. Sin embargo, en aplicaciones donde la percepción visual es crucial, otras métricas como el SSIM o el LPIPS pueden proporcionar evaluaciones más precisas.

4.1.2. SSIM (Structural Similarity Index)

El SSIM (*Structural Similarity Index*) es una métrica de calidad de imagen que evalúa la similitud estructural entre dos imágenes, considerando aspectos como luminancia, contraste y estructura [13]. A diferencia del PSNR, el SSIM intenta modelar las características percibidas por el sistema visual humano.

La fórmula general del SSIM se expresa como en la ecuación 6:

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

Donde:

- x e y son las dos imágenes que se están comparando.
- μ_x y μ_y son las medias de las intensidades de píxeles en las imágenes x e y , respectivamente.
- σ_x^2 y σ_y^2 son las varianzas de las intensidades de píxeles en las imágenes x e y , respectivamente.
- σ_{xy} es la covarianza entre las intensidades de píxeles en x e y .
- C_1 y C_2 son constantes que se utilizan para evitar la indeterminación en caso de que el denominador sea cercano a cero.

La fórmula del SSIM se desglosa en tres componentes multiplicativas que reflejan la luminancia, el contraste y la estructura:

- Luminancia: (Ecuación 7)

$$\frac{(2\mu_x\mu_y + C_1)}{(\mu_x^2 + \mu_y^2 + C_1)} \quad (7)$$

- Contraste: (Ecuación 8)

$$\frac{(2\sigma_{xy} + C_2)}{(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (8)$$

- Estructura: (Ecuación 9) (donde C_3 y C_4 son parámetros adicionales que controlan la influencia de la estructura)

$$\frac{(2\sigma_{xy} + C_3)}{(\sigma_x\sigma_y + C_4)} \quad (9)$$

El resultado final del SSIM es un valor en el rango $[-1,1]$, donde 1 indica una similitud perfecta entre las imágenes. Un SSIM más alto se asocia generalmente con una mayor similitud estructural y, por lo tanto, con una mayor calidad percibida.

El SSIM ha demostrado ser más consistente con las evaluaciones subjetivas de calidad visual que métricas más simples como el PSNR. Además, es capaz de capturar mejor la percepción humana al considerar la estructura y la relación entre los píxeles de las imágenes comparadas.

4.1.3. FSIM (Feature Similarity Index)

El FSIM (*Feature Similarity Index*) es otra métrica de calidad de imagen diseñada para evaluar la similitud estructural entre dos imágenes [14]. A diferencia de métricas más simples como el PSNR, el FSIM incorpora información de bajo, medio y alto nivel al utilizar características extraídas de la imagen. Esto lo hace más robusto en la evaluación de la calidad visual al considerar aspectos más complejos de la percepción humana.

La fórmula general del FSIM, representada en la ecuación 10, es bastante compleja y se basa en tres componentes principales que modelan la similitud en diferentes niveles de características:

$$FSIM(x, y) = \frac{l(x, y) * c(x, y) * s(x, y)}{l(x, x) * c(y, y) * s(x, x)} \quad (10)$$

Donde:

- $l(x, y)$ es la medida de similitud en el dominio de la luminancia.
- $c(x, y)$ es la medida de similitud en el dominio del contraste.
- $s(x, y)$ es la medida de similitud en el dominio de la estructura.

Cada uno de estos componentes se calcula utilizando fórmulas específicas que incorporan información de características de la imagen. Las fórmulas exactas son bastante complejas y pueden requerir el uso de operadores de transformación de imágenes.

Al igual que en el SSIM evaluaremos luminancia, contraste y estructura:

- Luminancia: (Ecuación 11)

$$\frac{(2\mu_x\mu_y+C_1)}{(\mu_x^2+\mu_y^2+C_1)} \quad (11)$$

- Contraste: (Ecuación 12)

$$\frac{(2\sigma_{xy}+C_2)}{(\sigma_x^2+\sigma_y^2+C_2)} \quad (12)$$

- Estructura: (Ecuación 13) (donde C_3 y C_4 son parámetros adicionales que controlan la influencia de la estructura)

$$\frac{(\sigma_{xy}+C_3)}{(\sigma_x\sigma_y+C_4)} \quad (13)$$

El resultado final del FSIM es el producto de estos tres componentes normalizado por las características de la imagen de referencia. Un FSIM más alto indica una mayor similitud estructural y, por lo tanto, una mayor calidad percibida.

El FSIM es una métrica avanzada que aborda las limitaciones de las métricas de bajo nivel y busca proporcionar una evaluación más completa de la similitud estructural entre imágenes.

4.1.4. RMSE (Root Mean Squared Error)

El RMSE (*Root Mean Squared Error*) es una métrica utilizada para evaluar la diferencia entre dos conjuntos de valores, comúnmente aplicada en problemas de regresión o para medir la precisión de modelos de predicción [15].

La fórmula general del RMSE se expresa como en la ecuación 14:

$$RMSE = \sqrt{\frac{1}{N-P} \sum_{j=1}^N (y_j - \hat{y}_j)^2} \quad (14)$$

Donde:

- N es el número total de observaciones o datos.
- y_j es el valor observado o real para la j -ésima observación.
- \hat{y}_j es el valor predicho o estimado para la j -ésima observación.
- P es el número de parámetros, incluyendo la constante.

El RMSE calcula la diferencia entre los valores reales y los valores predichos, eleva estas diferencias al cuadrado, toma el promedio de esas diferencias y finalmente, toma la raíz cuadrada de ese promedio. Este proceso tiene como objetivo penalizar los errores más grandes de manera proporcional, ya que las diferencias se elevan al cuadrado antes de tomar el promedio y la raíz cuadrada.

Algunas consideraciones sobre el RMSE:

- Sensibilidad a valores extremos: El RMSE es sensible a valores extremos o outliers, ya que eleva las diferencias al cuadrado. Si hay valores atípicos en los datos, pueden tener un impacto significativo en el RMSE.
- Unidad de medida: El RMSE tiene la misma unidad de medida que los datos originales. Esto puede facilitar la interpretación, ya que proporciona una medida de la magnitud del error en la misma escala que los datos.
- Utilizado en problemas de regresión: El RMSE es comúnmente utilizado para evaluar la precisión de modelos de regresión, donde se predice un valor numérico.

En el contexto de la superresolución de imágenes o evaluación de calidad de imágenes, el RMSE podría utilizarse para comparar los valores de píxeles en las imágenes mejoradas con los valores de píxeles en las imágenes de referencia originales. Sin embargo, métricas como el SSIM o el PSNR son más comúnmente utilizadas en este contexto, ya que están diseñadas específicamente para evaluar la similitud estructural y la percepción visual.

4.1.5. MAE (Mean Absolute Error)

El MAE (*Mean Absolute Error*) es una métrica comúnmente utilizada para evaluar la diferencia promedio entre dos conjuntos de valores, y es especialmente útil en problemas de regresión (al igual que el RMSE) o para medir la precisión de modelos de predicción [\[16\]](#).

La fórmula general del MAE se expresa como en la ecuación 15:

$$MAE = \frac{1}{N} \sum_{j=1}^N |y_j - \hat{y}_j| \quad (15)$$

Donde, al igual que en el RMSE:

- N es el número total de observaciones o datos.
- y_j es el valor observado o real para la j -ésima observación.
- \hat{y}_j es el valor predicho o estimado para la j -ésima observación.
- $|\cdot|$ denota el valor absoluto.

El MAE calcula la diferencia absoluta entre los valores reales y los valores predichos, toma el promedio de estas diferencias absolutas y proporciona una medida de la magnitud promedio de los errores. A diferencia del RMSE (*Root Mean Squared Error*), el MAE no eleva las diferencias al cuadrado, por lo que no penaliza más a los errores más grandes.

Algunas consideraciones sobre el MAE:

- Robustez a valores extremos: El MAE es menos sensible a valores extremos o outliers en los datos, ya que no eleva las diferencias al cuadrado. Si hay valores atípicos en los datos, no tendrán un impacto tan significativo como en el RMSE.
- Interpretación directa: El MAE tiene la misma unidad de medida que los datos originales, lo que facilita la interpretación, ya que proporciona una medida de la magnitud promedio de los errores en la misma escala que los datos.
- Utilizado en problemas de regresión: Al igual que el RMSE, el MAE es comúnmente utilizado para evaluar la precisión de modelos de regresión, donde se predice un valor numérico.

En el contexto de la superresolución de imágenes o evaluación de calidad de imágenes, el MAE podría utilizarse para comparar los valores de píxeles en las imágenes mejoradas con los valores de píxeles en las imágenes de referencia originales. Sin embargo, una vez más, otras métricas, como el SSIM o el PSNR, son normalmente preferidas en este contexto.

4.1.6. IOU (Intersection Over Union)

El IOU (*Intersection Over Union*) es una métrica utilizada en tareas de segmentación de imágenes y otros problemas de superposición, como la superresolución de segmentación semántica. Esta métrica mide la superposición entre dos regiones al calcular el área de intersección dividida por el área de unión [17].

La fórmula general del IOU se expresa como en la ecuación 16:

$$IOU = \frac{\text{Área de Intersección}}{\text{Área de Unión}} \quad (16)$$

Donde:

- El *Área de Intersección* es la región donde las dos áreas (por ejemplo, dos segmentaciones en una imagen) se superponen.
- El *Área de Unión* es la región total cubierta por ambas áreas.

El resultado del IOU está en el rango de 0 a 1, donde 0 indica ninguna superposición y 1 indica una superposición completa. Cuanto mayor sea el valor de IOU, mayor será la similitud o superposición entre las áreas comparadas.

Para calcular el IOU, primero se deben determinar las regiones de intersección y unión. En el contexto de la superresolución de imágenes, el IOU puede ser utilizado para comparar las áreas de superposición entre las regiones de la imagen mejorada y la imagen de referencia original (**Figura 12**).

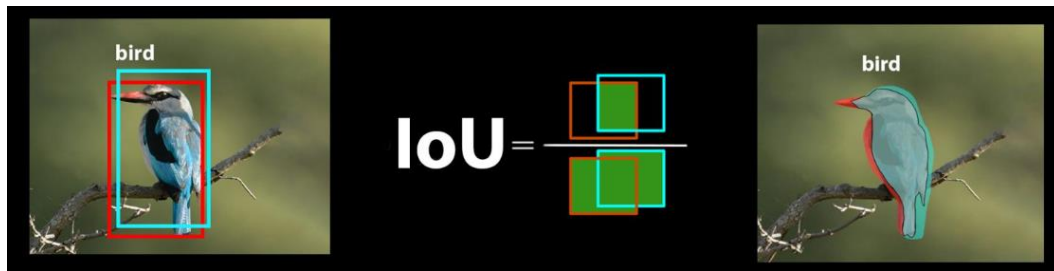


Figura 12 Funcionamiento del modelo IOU

Algunas consideraciones sobre el IOU:

- Sensibilidad a la superposición: El IOU proporciona una medida simple y efectiva de cuánto se superponen dos áreas, lo que lo hace útil en tareas de segmentación donde se evalúa la precisión espacial.
- No tiene en cuenta la posición exacta: El IOU no tiene en cuenta la posición exacta de los píxeles dentro de las áreas de intersección y unión, por lo que no distingue entre pequeñas desviaciones en la posición de los bordes.
- Ampliamente utilizado en problemas de segmentación: El IOU es comúnmente utilizado en problemas de segmentación de imágenes, como la segmentación semántica, para evaluar la precisión de los modelos al delinear áreas de interés.

En resumen, el IOU es una métrica valiosa en tareas donde la superposición entre áreas es crítica para la evaluación del rendimiento, como en problemas de segmentación de imágenes. En el contexto de la superresolución, podría ser utilizado para evaluar la precisión de las regiones detalladas en la imagen mejorada en comparación con las áreas correspondientes en la imagen de referencia.

4.2. Métricas de alto nivel

En el contexto de la superresolución de imágenes, las métricas de alto nivel son aquellas que evalúan la calidad perceptual y la utilidad de la imagen mejoradas desde una perspectiva más orientada al usuario. Estas métricas tienen en cuenta la percepción visual humana y buscan medir la calidad de la imagen resultante en términos de la apariencia global y la fidelidad visual.

Es importante destacar que las métricas de alto nivel no siempre capturan completamente la percepción subjetiva de la calidad de la imagen. La evaluación visual humana sigue siendo crucial para determinar la idoneidad de un modelo de superresolución en aplicaciones prácticas.

4.2.1. LPIPS (Learned Perceptual Image Patch Similarity)

LPIPS (*Learned Perceptual Image Patch Similarity*) es una métrica diseñada para evaluar la calidad perceptual de imágenes desde la perspectiva de la percepción visual humana, propuesta por Zhang et al. en 2018 como una alternativa a las métricas tradicionales de calidad de imagen, como PSNR y SSIM [18]. A diferencia de estas métricas, que se centran en la fidelidad estructural y la relación señal-ruido, LPIPS busca medir la percepción de las diferencias visuales entre dos imágenes.

Algunos puntos clave sobre LPIPS son:

- Enfoque perceptual: LPIPS se basa en modelos perceptuales para evaluar la calidad de las imágenes. Utiliza redes neuronales profundas entrenadas en datos de percepción visual humana para calcular la distancia perceptual entre dos imágenes.
- Redes neuronales convolucionales: La métrica utiliza una red neuronal convolucional (CNN) para extraer características perceptuales de las imágenes. Estas características capturan información sobre texturas, colores y estructuras visuales que son relevantes para la percepción humana.
- Distancia perceptual: LPIPS mide la distancia perceptual entre la imagen original y la imagen generada por un modelo. Cuanto menor sea la distancia

perceptual, más similar será la imagen generada a la imagen original desde la perspectiva de la percepción humana.

A diferencia de métricas más tradicionales, LPIPS ha demostrado ser más alineado con las evaluaciones perceptuales humanas. Puede ser particularmente útil cuando se trata de evaluar la calidad de imágenes generadas por modelos de inteligencia artificial que se centran en mejorar la percepción visual.

Es importante tener en cuenta que, aunque LPIPS es una métrica valiosa, no debe considerarse como la única medida de calidad. La evaluación visual humana y otras métricas tradicionales también son importantes para obtener una comprensión completa de la calidad de las imágenes generadas.

4.2.2. NIQE (Natural Image Quality Evaluator)

NIQE (*Natural Image Quality Evaluator*) es una métrica que fue propuesta por Mittal et al. en un artículo titulado "*Making a 'Completely Blind' Image Quality Analyzer*" [19]. Su objetivo principal es proporcionar una evaluación objetiva de la calidad de la imagen que coincida con las percepciones subjetivas de los humanos.

NIQE calcula una puntuación para una imagen comparando sus características estadísticas con las de un conjunto de imágenes naturales de referencia. Las puntuaciones más bajas indican una mejor calidad de imagen.

NIQE está formado por tres componentes principales:

- Extracción de características: El primer paso es extraer las características estadísticas de la imagen. Las características extraídas incluyen la distribución de intensidad, la correlación espacial y la estructura de frecuencia.
- Modelado de referencia: El segundo paso es crear un modelo de referencia de las características estadísticas de las imágenes naturales. El modelo de referencia se crea a partir de un conjunto de imágenes naturales de referencia.
- Cálculo de la puntuación: El tercer paso es calcular la puntuación NIQE comparando las características de la imagen con el modelo de referencia.

El NIQE tiene varias ventajas sobre otras métricas de calidad de imagen no referenciadas:

- Es relativamente robusto a los cambios en la iluminación y el contraste.
- Es capaz de detectar una amplia gama de distorsiones de imagen.

Sin embargo, el NIQE también tiene algunas limitaciones.

- Puede ser menos preciso para imágenes que contienen objetos artificiales o complejos.
- Puede ser sensible a las características de los conjuntos de datos de entrenamiento utilizados para crear el modelo de referencia.

En general, el NIQE es una métrica eficaz para evaluar la calidad de las imágenes mejoradas. Sin embargo, es importante tener en cuenta sus limitaciones al utilizarla.

4.2.3. VQI & VIQI

El Índice de Calidad Visual (VQI) y el Índice de Calidad de la Imagen Visual (VIQI) son dos métricas de calidad de imagen no referenciadas que se utilizan para evaluar la calidad de las imágenes que pueden estar distorsionadas o de baja calidad perceptual. Las métricas se basan en la idea de que la calidad percibida de una imagen está relacionada con su estructura visual [\[20\]](#). Estas dos métricas se usan menos que las otras explicadas en este trabajo.

El VQI calcula una puntuación para una imagen comparando su estructura visual con la de un conjunto de imágenes de referencia de alta calidad. Las puntuaciones más altas indican una mejor calidad de imagen.

El VIQI es similar al VQI, pero utiliza un conjunto de características visuales más complejo. El VIQI se ha encontrado que es más preciso que el VQI para evaluar la calidad de las imágenes distorsionadas.

El VQI se divide en:

- Extracción de características: El primer paso es extraer las características visuales de la imagen. Las características extraídas incluyen la textura, el contraste y la estructura espacial.
- Cálculo de la puntuación: El segundo paso es calcular la puntuación VQI comparando las características de la imagen con las de un conjunto de imágenes de referencia de alta calidad.

El VQI tiene varias ventajas sobre otras métricas de calidad de imagen no referenciadas. En primer lugar, el VQI es relativamente robusto a los cambios en la iluminación y el contraste. En segundo lugar, el VQI es capaz de detectar una amplia gama de distorsiones de imagen.

Sin embargo, el VQI también tiene algunas limitaciones. En primer lugar, el VQI puede ser menos preciso para imágenes que contienen objetos artificiales o complejos. En segundo lugar, el VQI puede ser sensible a las características de los

conjuntos de datos de entrenamiento utilizados para crear el conjunto de imágenes de referencia.

El VIQI se divide en:

- Extracción de características: El primer paso es extraer las características visuales de la imagen. Las características extraídas incluyen la textura, el contraste, la estructura espacial, la frecuencia y la orientación.
- Modelado de referencia: El segundo paso es crear un modelo de referencia de las características visuales de las imágenes de alta calidad. El modelo de referencia se crea a partir de un conjunto de imágenes de alta calidad.
- Cálculo de la puntuación: El tercer paso es calcular la puntuación VIQI comparando las características de la imagen con el modelo de referencia.

El VIQI tiene varias ventajas sobre el VQI. En primer lugar, el VIQI es más preciso para evaluar la calidad de las imágenes distorsionadas. En segundo lugar, el VIQI es menos sensible a los cambios en la iluminación y el contraste.

Sin embargo, el VIQI también tiene algunas limitaciones. En primer lugar, el VIQI es más complejo de calcular que el VQI. En segundo lugar, el VIQI puede ser más sensible a las características de los conjuntos de datos de entrenamiento utilizados para crear el modelo de referencia.

En general, el VQI y el VIQI son métricas de calidad de imagen no referenciadas eficaces para evaluar la calidad de las imágenes mejoradas. Sin embargo, es importante tener en cuenta las limitaciones de cada métrica al utilizarla.

4.3. Otras métricas

En algunos casos, las métricas de calidad de imagen generales no son suficientes para evaluar la calidad de una imagen en particular. Por ejemplo, una imagen puede tener una puntuación alta en una métrica general, pero puede tener bordes borrosos o texturas poco realistas. En estos casos, se pueden utilizar métricas específicas para evaluar características particulares, como la preservación de bordes, la reproducción de texturas y la mejora en detalles específicos.

4.3.1. Preservación de bordes

Una de las características más importantes de una imagen es la nitidez de los bordes. Los bordes son las líneas que separan objetos o regiones diferentes en una imagen. Los bordes borrosos pueden hacer que una imagen se vea borrosa o poco definida.

Para evaluar la preservación de bordes, se pueden utilizar métricas como el índice de nitidez de bordes (BRI) y el índice de nitidez de bordes de gradiente (GBRI). Estas métricas miden la cantidad de cambio de intensidad en los bordes de una imagen.

4.3.1.1. BRI (Boundary Response Index)

El BRI (*Boundary Response Index*) es una métrica de calidad de imagen que mide la nitidez de los bordes en una imagen. El BRI se calcula utilizando la ecuación 17:

$$BRI = 1 - \frac{(\sum_i^N |f(i) - g(i)|)^2}{\sum_i^N f(i)^2} \quad (17)$$

Donde:

- $f(i)$ es el valor de intensidad en la posición i de la imagen original
- $g(i)$ es el valor de intensidad en la posición i de la imagen mejorada

El BRI tiene un rango de valores de 0 a 1, donde 0 indica una imagen borrosa y 1 indica una imagen nítida.

4.3.1.2. GBRI (Gradient Boundary Response Index) es

El GBRI (*Gradient Boundary Response Index*) es una métrica de calidad de imagen similar al BRI, pero utiliza el gradiente de intensidad en lugar de la intensidad absoluta para medir la nitidez de los bordes. El GBRI se calcula utilizando la ecuación 18:

$$GBRI = 1 - \frac{(\sum_i^N |g(i)|)^2}{\sum_i^N g(i)^2} \quad (18)$$

Donde:

- $g(i)$ es el gradiente de intensidad en la posición i de la imagen original

El GBRI también tiene un rango de valores de 0 a 1, donde 0 indica una imagen borrosa y 1 indica una imagen nítida.

4.3.2. Reproducción de texturas

Las texturas son patrones repetitivos que se encuentran en muchas imágenes. Las texturas pueden ayudar a crear una sensación de profundidad y realismo en una imagen.

Para evaluar la reproducción de texturas, se pueden utilizar métricas como la similitud de textura y la medida de textura de correlación. Estas métricas miden la similitud entre las texturas de dos imágenes.

4.3.3. Mejora en detalles específicos

En algunos casos, se puede utilizar una métrica específica para evaluar la mejora de detalles específicos en una imagen. Por ejemplo, una métrica puede evaluar la mejora en la claridad de los ojos o la definición de las líneas de expresión.

Para evaluar la mejora en detalles específicos, se pueden utilizar métricas como el índice de mejora de detalles (MDI) y la medida de mejora de detalles (EDM). Estas métricas miden la diferencia entre la imagen original y la imagen mejorada.

4.3.3.1. MDI (Mean Directional Index)

El MDI (*Mean Directional Index*) es una métrica que evalúa la calidad de las imágenes al medir la coherencia de las direcciones de los bordes. Se centra en la consistencia de la orientación de los bordes en la imagen mejorada, proporcionando información sobre la preservación de la estructura direccional y detalles importantes en la imagen. El MDI se calcula utilizando la ecuación 19:

$$MDI = \frac{\sum_i^N |f(i) - g(i)|}{\sum_i^N f(i)} \quad (19)$$

Donde:

- $f(i)$ es el valor de intensidad en la posición i de la imagen original
- $g(i)$ es el valor de intensidad en la posición i de la imagen mejorada

El MDI tiene un rango de valores de 0 a 1, donde 0 indica que no hay mejora y 1 indica una mejora perfecta.

4.3.3.2. EDM (Error Distortion Measure)

La EDM (*Error Distortion Measure*) es una métrica de calidad de imagen similar al MDI, pero utiliza el error cuadrático medio (MSE) para medir la mejora de los detalles. La EDM se calcula utilizando la ecuación 20:

$$EDM = \frac{\sum_i^N (f(i) - g(i))^2}{\sum_i^N f(i)^2} \quad (20)$$

donde:

- $f(i)$ es el valor de intensidad en la posición i de la imagen original
- $g(i)$ es el valor de intensidad en la posición i de la imagen mejorada

La EDM también tiene un rango de valores de 0 a 1, donde 0 indica que no hay mejora y 1 indica una mejora perfecta.

5. Casos de uso de superresolución de imágenes

A lo largo de este trabajo hemos explicado qué es la resolución de imágenes. Además, centrándonos en la resolución de imágenes utilizando la inteligencia artificial como herramienta, hemos analizado las diferentes tecnologías que podemos utilizar, así como las diferentes arquitecturas que se han utilizado a lo largo del tiempo para este fin.

En este trabajo en concreto nos centraremos en la superresolución de imágenes de satélite, ya que podría tener un fin más relevante que la superresolución de cualquier otra imagen para uso lúdico o privado. La capacidad de mejorar de una manera precisa las imágenes generadas por un satélite podría permitirnos entre otras cosas:

- Mejorar la vigilancia y seguridad: Las imágenes más nítidas pueden ayudar a los gobiernos y a las fuerzas del orden a vigilar mejor las fronteras, detectar actividades ilegales y prevenir el terrorismo.
- Mejora de la gestión de desastres: Las imágenes más nítidas pueden ayudar a los equipos de respuesta a desastres a evaluar mejor la magnitud de un desastre y a desarrollar planes de respuesta más efectivos.
- Mejora de la planificación urbana: Las imágenes más nítidas pueden ayudar a los planificadores urbanos a comprender mejor el entorno urbano y a tomar decisiones más informadas sobre el desarrollo.
- Mejora de la agricultura: Las imágenes más nítidas pueden ayudar a los agricultores a monitorear el crecimiento de los cultivos, identificar plagas y enfermedades y tomar decisiones más informadas sobre la irrigación y la fertilización.
- Mejora de la investigación científica: Las imágenes más nítidas pueden ayudar a los científicos a estudiar el medio ambiente, el clima y otros fenómenos naturales.

Además, el poder mejorar las imágenes de satélite hará que las imágenes de satélite de alta calidad sean accesibles para más usuarios, ya que, pese a que hay satélites de menor resolución accesibles para el público general, el acceso a satélites de alta resolución suele ser muy costoso. Por lo tanto, mejorar a través de la superresolución el rendimiento de las imágenes recibidas de estos satélites de baja resolución hará que más usuarios puedan beneficiarse de unas buenas imágenes de satélite para sus propósitos.

A continuación se presentan algunos ejemplos específicos de cómo se están utilizando las imágenes de satélite de alta resolución:

- La Agencia Espacial Europea (ESA) está utilizando imágenes de satélite de alta resolución para monitorear la deforestación en la Amazonia.
- La NASA está utilizando imágenes de satélite de alta resolución para estudiar los efectos del cambio climático en los glaciares.
- *Google Earth Engine* está utilizando imágenes de satélite de alta resolución para proporcionar información sobre el uso de la tierra, la contaminación y otros problemas ambientales.

A medida que la tecnología de los satélites de alta resolución siga mejorando, es probable que veamos más aplicaciones innovadoras para estas imágenes.

5.1. Tecnologías

Para llevar a cabo las pruebas y comparaciones en este trabajo se han utilizado varias tecnologías diferentes, explicadas en esta sección.

5.1.1. Python

Python es un lenguaje de programación de alto nivel, interpretado, orientado a objetos, de código abierto y multiplataforma [21]. Es uno de los lenguajes de programación más populares del mundo, y es especialmente popular en el campo de la inteligencia artificial, incluido el *deep learning*.

Tiene una serie de ventajas que lo hacen ideal para el desarrollo de aplicaciones de *deep learning*. Estas ventajas incluyen:

- Simple y fácil de usar: Python es un lenguaje de programación de relativa simplicidad y fácil de aprender. Esto lo hace ideal para desarrolladores que no tienen una base sólida en ciencias de la computación.
- Fluidez y expresividad: Python es un lenguaje de programación fluido y expresivo. Esto lo hace ideal para el desarrollo de código complejo y sofisticado.
- Bibliotecas de *deep learning* de código abierto: Existe una amplia gama de bibliotecas de *deep learning* de código abierto disponibles para Python. Estas bibliotecas facilitan el desarrollo de aplicaciones de *deep learning* sin tener que empezar desde cero.

Como ya hemos comentado, Python se utiliza para una amplia gama de aplicaciones de *deep learning* dadas sus características. Algunas de estas aplicaciones son el entrenamiento de redes neuronales, la implementación de modelos de *deep learning* en aplicaciones reales o el análisis de datos de *deep learning* para comprender cómo funcionan los modelos y mejorar su rendimiento.

Algunas de las bibliotecas clave para la implementación de estas aplicaciones son:

- NumPy: Esencial para el manejo de arreglos y operaciones matriciales. Muchas bibliotecas de aprendizaje profundo se basan en NumPy. Más adelante en este trabajo se hablará de NumPy, puesto que se ha utilizado para implementar el modelo usado.
- Pandas: Ideal para manipulación y análisis de datos. Facilita la carga y manipulación de conjuntos de datos.
- Scikit-learn: Proporciona herramientas para el aprendizaje de máquinas y la minería de datos. Incluye implementaciones de algoritmos comunes.
- TensorFlow: Desarrollado por Google, TensorFlow es una de las bibliotecas más populares para la construcción y entrenamiento de modelos de aprendizaje profundo.
- PyTorch: Desarrollado por Facebook, PyTorch es otra biblioteca popular para el aprendizaje profundo. Destaca por su flexibilidad y una curva de aprendizaje más suave. Más adelante en este trabajo se hablará de PyTorch, puesto que también se ha utilizado para implementar el modelo usado.
- Keras: Aunque ahora es parte de TensorFlow, Keras es una interfaz de alto nivel para la construcción de modelos de aprendizaje profundo y es muy amigable para principiantes.
- Scikit-learn: Además de sus capacidades de aprendizaje de máquinas, Scikit-learn también tiene herramientas para el aprendizaje profundo.
- CV2: Biblioteca utilizada para visión por computadora, algo esencial en muchos problemas de inteligencia artificial. CV2 también se ha utilizado en este trabajo para implementar el modelo.

En resumen, Python es un lenguaje de programación esencial para el desarrollo de aplicaciones de *deep learning*. Es un lenguaje simple, flexible y potente que ofrece

una amplia gama de bibliotecas y herramientas para facilitar el desarrollo de aplicaciones de *deep learning*.

5.1.2. PyTorch

PyTorch es una biblioteca de Python de código abierto para el aprendizaje automático y el *deep learning*. Fue desarrollada por Facebook AI Research y se lanzó por primera vez en 2016 [22].

PyTorch es una biblioteca flexible y eficiente que se centra en la investigación. Ofrece una amplia gama de funciones para el desarrollo de aplicaciones de *deep learning*, incluyendo:

- Creación de redes neuronales: PyTorch proporciona funciones para crear redes neuronales de diferentes tipos, incluyendo redes neuronales profundas, redes neuronales convolucionales y redes neuronales recurrentes.
- Entrenamiento de redes neuronales: PyTorch proporciona funciones para entrenar redes neuronales utilizando diferentes algoritmos de aprendizaje automático, incluyendo descenso de gradiente estocástico, aprendizaje por refuerzo y aprendizaje por transferencia.
- Implementación de redes neuronales: PyTorch proporciona funciones para implementar redes neuronales en aplicaciones reales.

Lo que ha diferenciado a PyTorch para su uso en los modelos de *deep learning* ha sido:

- Los tensores: La estructura fundamental en PyTorch es el tensor. Un tensor es una matriz n-dimensional que puede representar escalares, vectores, matrices u otros tipos de datos. PyTorch proporciona una amplia variedad de operaciones tensoriales que facilitan la manipulación y transformación de datos.
- Diferenciación Automática: Una de las características más potentes de PyTorch es su sistema de diferenciación automática (*autograd*). Permite calcular automáticamente gradientes para las operaciones en tensores, lo que es esencial para el entrenamiento de modelos de aprendizaje profundo mediante optimización de gradientes.
- Grafos Computacionales Dinámicos: A diferencia de algunas otras bibliotecas, PyTorch utiliza grafos computacionales dinámicos. Esto significa que el grafo de computación se construye sobre la marcha durante

la ejecución, lo que proporciona flexibilidad para cambiar la arquitectura del modelo durante el tiempo de ejecución.

- Módulos y Capas: PyTorch proporciona la clase ‘torch.nn.Module’ que facilita la construcción y organización de modelos. Las capas y operaciones comunes se encuentran en el módulo ‘torch.nn’.
- Optimizadores: PyTorch incluye varios optimizadores como SGD, Adam, o RMSprop, que facilitan la optimización de los modelos durante el entrenamiento.
- GPU: PyTorch es compatible con GPU. Esto permite acelerar el entrenamiento de redes neuronales en hardware de alto rendimiento. Esto es lo que se ha hecho en este trabajo, ya que se ha seleccionado la GPU de la manera mostrada en la **Figura 13**:

```
device = torch.device('cuda')
```

Figura 13 Comando para utilizar la GPU

PyTorch se puede instalar utilizando la herramienta de administración de paquetes pip. Para instalar PyTorch, se abre una terminal y se ejecuta el comando mostrado en la **Figura 14**:

```
pip install torch
```

Figura 14 Comando para instalar PyTorch

PyTorch es una biblioteca poderosa y flexible que puede utilizarse para una amplia gama de aplicaciones de *deep learning*. Es una buena opción para desarrolladores que buscan una biblioteca que les permita crear modelos de *deep learning* de alto rendimiento.

5.1.3. Numpy

NumPy es una biblioteca de Python para la computación científica que proporciona una estructura de datos multidimensionales (matrices) [23]. PyTorch se integra bien con NumPy, lo que facilita la conversión entre tensores PyTorch y matrices NumPy.

En el caso concreto del *deep learning*, NumPy es una herramienta esencial. En particular, NumPy se utiliza para:

- Representar datos: En *deep learning*, los datos se representan comúnmente como tensores multidimensionales (**Figura 15**). NumPy proporciona la estructura de datos ndarray que es esencial para almacenar y manipular estos tensores.

```
import numpy as np

# Crear un tensor (arreglo multidimensional)
data = np.array([[1, 2, 3], [4, 5, 6]])
```

Figura 15 Representación de datos con Numpy

- Realizar operaciones matemáticas: Antes de alimentar datos a un modelo de *deep learning*, es común realizar operaciones de preprocesamiento como normalización, escalado y división de conjuntos de entrenamiento y prueba. Además, NumPy proporciona una amplia gama de funciones matemáticas para realizar operaciones matemáticas en matrices, así como realizar operaciones de manera eficiente en todos los elementos de un tensor sin la necesidad de bucles explícitos. Estas funciones son esenciales para el entrenamiento de redes neuronales, que requieren realizar operaciones matemáticas complejas en grandes cantidades de datos, como las realizadas en la **Figura 16**:

```
# Normalizar datos
normalized_data = (data - np.mean(data)) / np.std(data)

# Dividir conjunto de datos
train_data, test_data = np.split(data, [80])

# Multiplicación de matrices
result = np.dot(matrix_a, matrix_b)

# Operación elemento a elemento
result = np.sin(data)
```

Figura 16 Operaciones matemáticas con Numpy

- Generar datos aleatorios: NumPy proporciona funciones para generar números aleatorios (**Figura 17**). Estas funciones son útiles para generar datos de entrenamiento para redes neuronales.

```
# Generar un entero aleatorio en el rango (0, 10)
random_integer = np.random.randint(0, 10)

# Generar una matriz de forma (3, 2) con números aleatorios
random_array = np.random.rand(3, 2)
```

Figura 17 Generación de datos aleatorios con Numpy

NumPy es una herramienta esencial para el desarrollo de aplicaciones de *deep learning*. Proporciona una estructura de datos eficiente, funciones matemáticas potentes y funciones de generación de datos aleatorios. Estas características hacen que NumPy sea una herramienta indispensable para cualquier desarrollador que trabaje con datos numéricos y *deep learning*.

5.1.4. CUDA

CUDA (*Compute Unified Device Architecture*) es una plataforma de computación paralela desarrollada por NVIDIA [24]. Es específicamente diseñada para aprovechar el poder de procesamiento de las unidades de procesamiento gráfico (GPU) de NVIDIA. CUDA permite a los desarrolladores utilizar las GPU para realizar cálculos de propósito general, no solo para gráficos.

En el contexto del *deep learning*, CUDA es fundamental para acelerar el entrenamiento y la inferencia de modelos de aprendizaje profundo. Como ya se ha dicho anteriormente, se ha utilizado CUDA para acelerar la superresolución de imágenes con nuestros modelos preentrenados con el comando mostrado en la **Figura 18**:

```
device = torch.device('cuda')
```

Figura 18 Comando para utilizar la GPU

Algunas de las razones por las cuales CUDA es relevante en el ámbito del *deep learning* son:

- La capacidad de las GPU para realizar operaciones en paralelo masivo, esto es fundamental para acelerar el entrenamiento de modelos de *deep learning*. Las tareas computacionales intensivas, como el cálculo de gradientes y la multiplicación de matrices, se pueden realizar de manera significativamente más rápida en una GPU que en una CPU.
- Como ya se ha visto, muchas operaciones en *deep learning* involucran cálculos matriciales. CUDA, junto con bibliotecas optimizadas como cuBLAS (CUDA Basic Linear Algebra Subroutines), permite realizar estas operaciones de manera eficiente y paralela en la GPU.
- Además, las principales bibliotecas que se han explicado en este apartado (como TensorFlow y PyTorch) han sido optimizadas para utilizar GPU a través de CUDA. Estas bibliotecas realizan automáticamente las operaciones en la GPU cuando están disponibles, lo que acelera significativamente el entrenamiento de modelos.

En resumen, CUDA y las GPU de NVIDIA han desempeñado un papel crucial en el avance y la adopción del *deep learning*. Han permitido que los modelos de aprendizaje profundo sean entrenados en tiempos razonables y han hecho posible la implementación de soluciones de inteligencia artificial en aplicaciones del mundo real. La aceleración proporcionada por CUDA ha contribuido significativamente al progreso y éxito de la inteligencia artificial y el aprendizaje profundo.

5.1.5. OpenCV

OpenCV (*Open Source Computer Vision Library*), a menudo referido como cv2 en su interfaz de Python, es una biblioteca de código abierto para visión por computadora y procesamiento de imágenes [25]. OpenCV está diseñado para proporcionar una amplia variedad de herramientas y algoritmos para manipular, procesar y analizar imágenes y secuencias de vídeo. Fue desarrollada por Intel y se lanzó por primera vez en 2000. A continuación se presenta una explicación detallada de algunas características clave de OpenCV en Python (cv2).

Las características principales de CV2 incluyen:

- Funciones de procesamiento de imágenes: CV2 proporciona una amplia gama de funciones para realizar operaciones de procesamiento de imágenes, como la carga y visualización de imágenes, la manipulación de imágenes y el análisis de imágenes.
- Funciones de visión artificial: CV2 proporciona una amplia gama de funciones para realizar tareas de visión artificial, como la detección de objetos, el seguimiento de objetos y la clasificación de imágenes.
- Compatibilidad con GPU: CV2 es compatible con GPU, lo que permite acelerar el procesamiento de imágenes y la visión artificial.

CV2 es una biblioteca poderosa y versátil que puede utilizarse para una amplia gama de aplicaciones de procesamiento de imágenes y visión artificial. Es una herramienta esencial para cualquier desarrollador que trabaje con imágenes y visión artificial.

5.1.6. Modelos preentrenados

En este trabajo se ha empleado un modelo ESRGAN (*Enhanced Super-Resolution Generative Adversarial Network*), ya que de los modelos explicados es el que mejores resultados proporciona al tratarse de un modelo optimizado para la aplicación específica en superresolución de imágenes.

Los modelos preentrenados de un modelo ESRGAN son redes neuronales convolucionales que han sido entrenadas en un conjunto de datos de imágenes de baja resolución y sus correspondientes imágenes de alta resolución [26]. El proceso de entrenamiento de estos modelos es muy costoso y requiere una gran cantidad de datos. Por ello, como en este trabajo, es común utilizar modelos preentrenados que han sido entrenados por otros investigadores.

Para crear un modelo preentrenado de ESRGAN, se sigue el siguiente proceso:

- Se recopila un conjunto de datos de imágenes de baja resolución y sus correspondientes imágenes de alta resolución.
- Se entrena una red neuronal convolucional en el conjunto de datos.
- Se evalúa el rendimiento del modelo en un conjunto de datos de prueba.

El conjunto de datos de entrenamiento debe ser lo suficientemente grande y diverso para que el modelo pueda aprender a realizar la tarea de superresolución. El conjunto de datos de prueba se utiliza para evaluar el rendimiento del modelo y detectar posibles sesgos. Este proceso puede tardar varios días o semanas, dependiendo del tamaño del conjunto de datos y de la complejidad de la red neuronal.

Una vez que se ha creado un modelo preentrenado de ESRGAN, se puede utilizar para aumentar la resolución de imágenes de baja resolución. Para ello, se carga el modelo preentrenado en un programa de procesamiento de imágenes y se pasa la imagen de baja resolución al modelo, tal y como se muestra en la **Figura 19**. El modelo devolverá una imagen de alta resolución que es una estimación de la imagen original de alta resolución.

```
model_path = 'models/RRDB_ESRGAN_x4.pth'
# models/RRDB_ESRGAN_x4.pth o models/RRDB_PSNR_x4.pth

model = arch.RRDBNet(3, 3, 64, 23, gc=32)
model.load_state_dict(torch.load(model_path), strict=True)
# En la línea superior se carga el modelo preentrenado

...

test_img_folder = 'LR/*'
```

Figura 19 Comando para utilizar la GPU

Ambos modelos aumentarán la resolución de las imágenes de entrada (ubicadas en la carpeta LR) con un factor de 4. La opción *strict=True* garantiza que las claves en el archivo coincidan exactamente con las claves del modelo, es decir, el diccionario de pesos cargado (nuestro modelo preentrenado) debe tener las mismas claves que el modelo del programa, si hay alguna discrepancia, se genera un error. Este proceso es comúnmente utilizado cuando se desea restaurar un modelo previamente entrenado para continuar el entrenamiento o para realizar inferencias con el modelo ya entrenado.

5.1.7. Aplicaciones web

A parte de utilizar un programa de superresolución de imágenes utilizando modelos preentrenados, también se han utilizado herramientas online especializadas en la superresolución de imágenes.

En concreto se han utilizado dos aplicaciones web gratuitas para el usuario (aunque disponen de versión de pago): upscale.media [27] y cutout.pro [28].

Estas webs, en general, han dado mejores resultados (aunque más artificiales) que nuestro programa a la hora de aumentar la resolución, esto puede ser debido a que, al ser páginas tan concurridas, disponen de muchos más datos e imágenes para entrenar a sus modelos, así como servidores más potentes para hacerlo.

5.2. Implementación

En esta sección del trabajo pondremos a prueba el modelo preentrenado con imágenes externas, así como ambas webs, Upscale.media y Cutout.pro. Las imágenes se han extraído de Google Earth [29], ya que es una fuente gratuita y accesible, de donde se pueden extraer imágenes de satélite con relativa buena calidad, de casi cualquier rincón del planeta. Estas imágenes serán enviadas a los 5 modelos previamente explicados para analizar sus resultados, tanto subjetivamente como objetivamente, apoyándose en métricas, tanto de bajo nivel, como de alto nivel, así como otras métricas que se especialicen en otros aspectos; todas ellas ya explicadas en este trabajo.

En este caso, se han tomado capturas de una región al sur de Andalucía, siendo las 4 imágenes del mismo sitio, cada una a menos distancia que la anterior. Podemos ver las 4 imágenes en la **Figura 13**:





Figura 13 Conjunto de imágenes de satélite de baja resolución

5.2.1. ESRGAN

Primero se analizará el resultado que ofrece el modelo SRGAN preentrenado. Para ello se introducirán las imágenes que se quieren mejorar en la carpeta LR/ del proyecto, con el código mostrado en la **Figura 20**:

```
Test_img_folder = 'LR/'
```

Figura 20 Selección de la carpeta con imágenes de baja resolución

Y posteriormente se recorrerán las imágenes de este directorio haciendo uso de un *for* y de la biblioteca 'glob' para iterar sobre los archivos de nuestro directorio de la manera mostrada en la **Figura 21**:

```
For path in glob.glob(test_img_folder):
```

Figura 21 Iteración sobre la carpeta de baja resolución

En resumen, este código recorre todos los archivos en el directorio LR (o en cualquier subdirectorio dentro de LR) y realiza una superresolución con cada archivo (imagen) encontrado de la manera mostrada en la **Figura 22**:

```
for path in glob.glob(test_img_folder):
    idx += 1
    base = osp.splitext(osp.basename(path))[0]
    print(idx, base)
    # read images
    img = cv2.imread(path, cv2.IMREAD_COLOR)
    img = img * 1.0 / 255
    img = torch.from_numpy(np.transpose(img[:, :, [2, 1, 0]], (2,
0, 1))).float()
    img_LR = img.unsqueeze(0)
    img_LR = img_LR.to(device)

    with torch.no_grad():
        output =
model(img_LR).data.squeeze().float().cpu().clamp_(0, 1).numpy()
    output = np.transpose(output[[2, 1, 0], :, :], (1, 2, 0))
    output = (output * 255.0).round()
    cv2.imwrite('results/{:s}_r1t.png'.format(base), output)
```

Figura 22 Cuerpo del código del modelo preentrenado

- **idx += 1**: Incrementa un índice (idx) en 1 en cada iteración del bucle. Este índice se utiliza para numerar los archivos procesados.

- **base = osp.splitext(osp.basename(path))[0]:** Utiliza las funciones del módulo *'os.path'* para obtener el nombre base del archivo sin la extensión. *'osp.basename(path)'* obtiene el nombre de archivo de la ruta completa, y *'osp.splitext(...)'* divide el nombre de archivo y su extensión, y [0] toma solo el nombre de archivo.
- **print(idx, base):** Muestra por pantalla el índice y el nombre base del archivo para realizar un seguimiento del progreso del procesamiento.
- **img = cv2.imread(path, cv2.IMREAD_COLOR):** Lee la imagen del archivo utilizando la librería ya explicada en este trabajo *OpenCV* y la almacena en la variable *'img'*.
- **img = img * 1.0 / 255:** Normaliza los valores de píxeles de la imagen para que estén en el rango [0, 1].
- **img = torch.from_numpy(np.transpose(img[:, :, [2, 1, 0]], (2, 0, 1))).float():** Convierte la imagen en un tensor de *'PyTorch'* y realiza operaciones de transposición para cambiar el orden de los canales de la imagen.
- **img_LR = img.unsqueeze(0):** Añade una dimensión adicional al tensor para representar el tamaño del lote (*batch size*). Esto es necesario para alimentar la imagen a un modelo de *'PyTorch'* que espera un lote de imágenes.
- **img_LR = img_LR.to(device):** Mueve el tensor al dispositivo especificado, *'gpu'* en nuestro caso, para la inferencia del modelo.
- **with torch.no_grad():** Se inicia un bloque de código en el que las operaciones de gradiente no se registran. Esto se utiliza para la inferencia, donde no se necesita realizar el retroceso para ajustar los pesos del modelo.
- **output = model(img_LR).data.squeeze().float().cpu().clamp_(0, 1).numpy():** Realiza la inferencia en el modelo con la imagen de baja resolución. Después, realiza operaciones para obtener una imagen de salida procesada y la convierte nuevamente a un arreglo *'NumPy'*.
- **output = np.transpose(output[[2, 1, 0], :, :], (1, 2, 0)):** Realiza operaciones de transposición para cambiar el orden de los canales de la imagen de salida.
- **output = (output * 255.0).round():** Realiza ajustes para convertir los valores de píxeles nuevamente al rango [0, 255] y redondea los valores.
- **cv2.imwrite('results/{:s}_rlt.png'.format(base), output):** Guarda la imagen de salida en el directorio *'results'* con un nombre de archivo que incluye el nombre base original y *"_rlt.png"*. Esto guarda los resultados procesados para cada imagen en el directorio de resultados.

Tras ejecutar el programa con las imágenes de 'baja resolución' se obtendrán en la carpeta de *'results'* las imágenes mejoradas. A continuación, en la **Figura 23**, se

pueden observar, a la izquierda nuevamente las imágenes originales, y a la derecha las imágenes mejoradas que ha obtenido el programa:

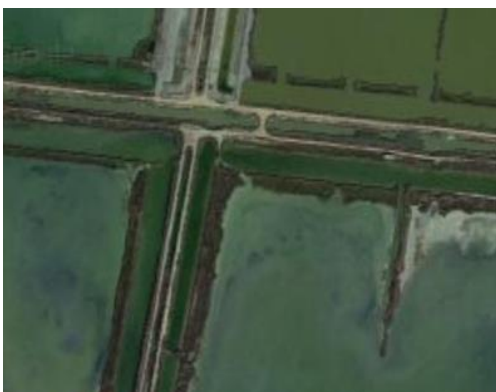
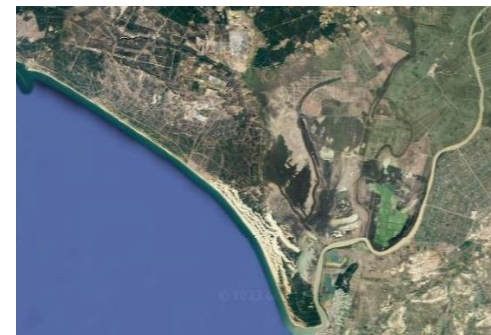
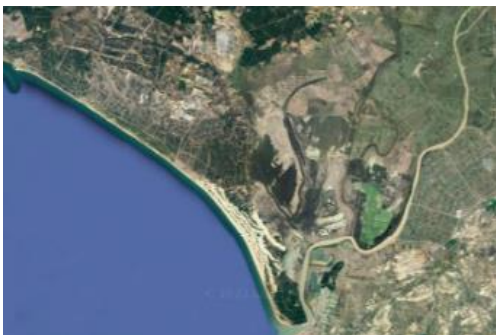


Figura 23 Conjunto de imágenes de baja resolución (izquierda), mejoradas con ESRGAN (derecha)

Sin hacer demasiado *zoom* se puede apreciar una mejora en ellas, la cual se hace más notable cuanto más *zoom* se haga sobre las imágenes. También se aprecia que las imágenes mejoradas guardan bastante fidelidad con la imagen original. El modelo SRGAN devuelve una imagen con el cuádruple de tamaño y una profundidad de bits inferior (24 bits de profundidad frente a los 32 bits de la imagen original), lo que hace que sea menos preciso en cuanto a gama de colores.

La profundidad de bits de una imagen, también conocida como bits por píxel (*bpp*), es la cantidad de información que se utiliza para representar el color de cada píxel en la imagen. Cuantos más bits se utilicen para representar cada píxel, mayor es la gama de colores que se puede representar y mayor es la calidad de la imagen.

5.2.2. Upscale.media

Ahora se va a analizar el resultado de utilizar el modelo que provee Upscale.media, tanto la versión de superresolución básica, como el modelo mejorado (*enhanced quality*). En ambos modelos la propia web nos muestra una comparación entre la imagen original y la imagen escalada con inteligencia artificial en la parte derecha, así como una comparación entre la imagen original y un simple escalado de esta en la izquierda. Esto se puede observar en la **Figura 24** para la versión mejorada (*enhanced quality*) y en la **Figura 25** para la versión normal.

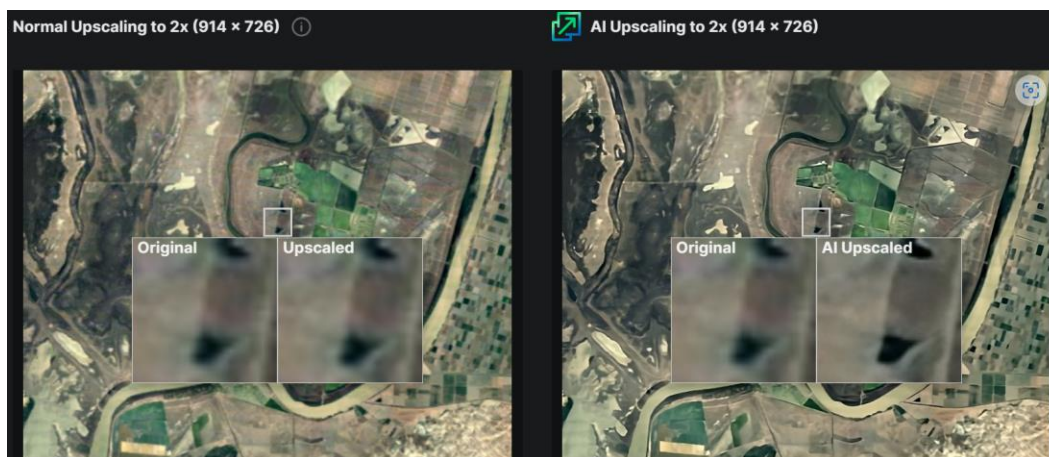


Figura 24 Comparación entre la imagen original y la imagen escalada de forma tradicional (izquierda), y la imagen original y la imagen escalada con inteligencia artificial (derecha) con la opción *Enhance Quality* activada

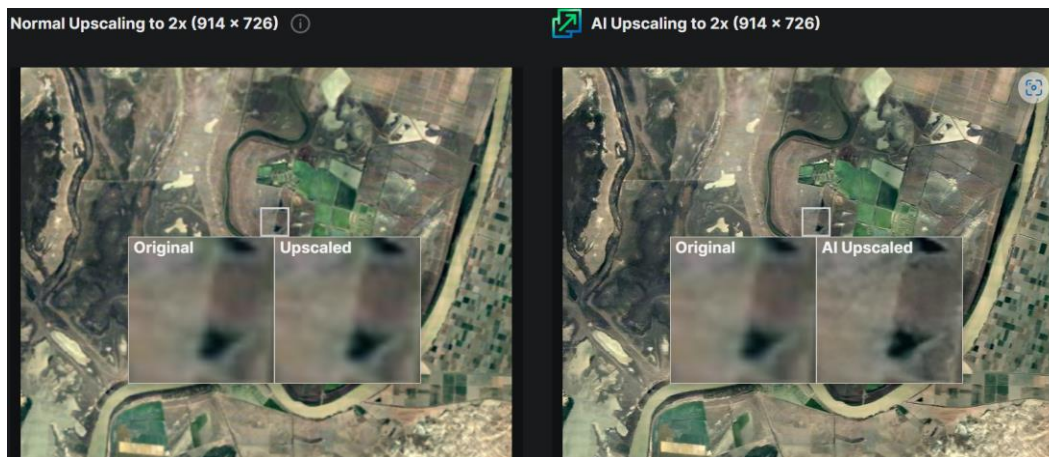


Figura 25 Comparación entre la imagen original y la imagen escalada de forma tradicional (izquierda), y la imagen original y la imagen escalada con inteligencia artificial (derecha) con la opción *Enhance Quality* desactivada

En ambas imágenes se puede ver como el escalado simple de una imagen, es decir, el multiplicar el número de píxeles de una imagen, no es para nada efectivo, ya que la imagen escalada se ve igual de borrosa que la original.

Por otro lado, empezando por la superresolución sin la opción *enhance quality*, se puede apreciar cierta mejora en la imagen, con algo de ruido todavía, pero mejorando bastante la calidad de la original.

Lo mismo se puede decir de la superresolución con la opción *enhanced quality* activada, siendo en este caso la mejora de la imagen original considerablemente mejor.

Esta vez el modelo de Upscale.media devuelve una imagen mejorada de 32 bits de profundidad, al igual que la imagen original, pero con solo el doble de tamaño que la imagen original, frente al cuádruple de tamaño que ofrece el modelo ESRGAN preentrenado o Cutout.pro.

En la **Figura 26** se puede observar la imagen mejorada con ambas opciones normal (derecha) y con la opción *enhanced quality* (izquierda), frente a la imagen de baja resolución original (abajo).

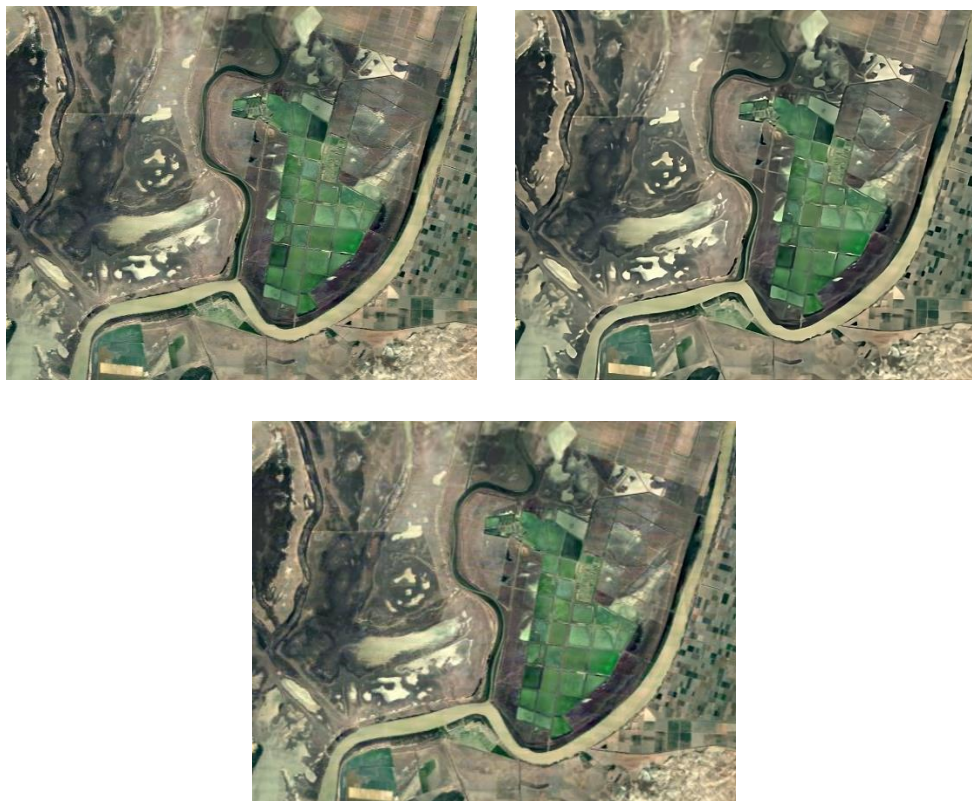
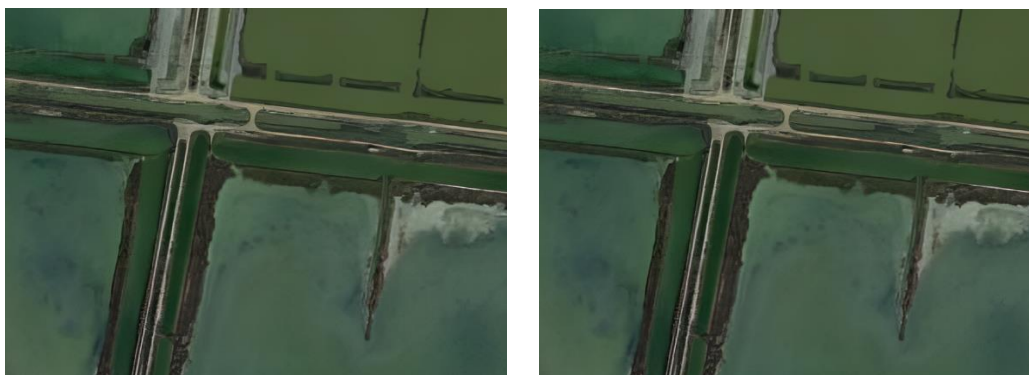


Figura 26 Imagen de baja resolución (abajo), mejorada con *upscale.media* y la opción *enhanced* activada (izquierda) y desactivada (derecha)

5.2.3. Cutout.pro

Ahora se va a analizar el resultado de aplicar los modelos que ofrece Cutout.pro, el modelo de *Quality* y el modelo de *Fidelity*. Como sus propios nombres en inglés indican, el primer modelo se centra en obtener la mayor calidad posible, mientras que el segundo también intentará obtener la mayor calidad posible, pero centrándose en mantener lo máximo posible las características de la imagen original.

En la **Figura 27**, **Figura 28** y **Figura 29** se pueden observar los resultados.



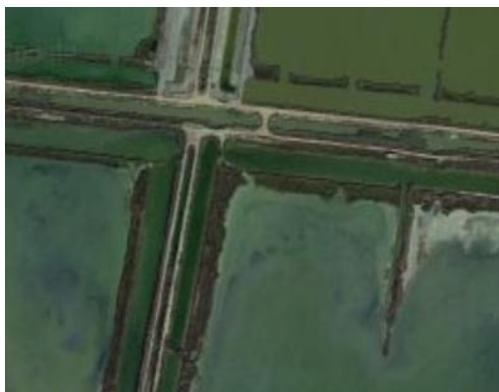


Figura 27 Imagen de baja resolución (abajo), mejorada con cutout.pro en la opción Quality (izquierda) y la opción Fidelity (derecha)



Figura 28 Imagen de baja resolución (abajo), mejorada con cutout.pro en la opción Quality (izquierda) y la opción Fidelity (derecha)



Figura 29 Imagen de baja resolución (abajo), mejorada con cutout.pro en la opción *Quality* (izquierda) y la opción *Fidelity* (derecha)

Se puede apreciar que ambos modelos son bastante eficientes a la hora de mejorar la calidad de la imagen original. No se observa gran diferencia, sin embargo, entre el modelo *Quality* y el modelo *Fidelity*, ya que el primero de ellos ya es bastante fiel a la imagen original, y por lo general parece que ofrece algo más de calidad que el modelo *Fidelity*, aunque nada que se aprecie si no se hace el suficiente *zoom* en las imágenes.

Ambos modelos devuelven una imagen mejorada de 32 bits de profundidad, al igual que la imagen original, y del cuádruple de tamaño que esta.

5.2.4. Implementación de métricas

Hasta ahora se ha podido medir la efectividad de los modelos simplemente analizando las imágenes mejoradas, pero solo desde un punto de vista subjetivo. Para poder analizar objetivamente la efectividad de la superresolución de imágenes llevada a cabo en este trabajo, es necesario aplicar las métricas ya explicadas y analizar los resultados, dichas métricas se dividen en métricas de bajo y alto nivel:

- Métricas de bajo nivel: Se centran en medidas numéricas y estadísticas directas de la señal de la imagen, como la diferencia promedio de intensidad de píxeles, la relación señal-ruido, la similitud estructural, etc. Son objetivas y proporcionan una evaluación cuantitativa de la calidad de la imagen,

además pueden no reflejar completamente la percepción visual humana, ya que no consideran ciertos aspectos subjetivos de la calidad de la imagen.

- **Métricas de alto nivel:** Se centran en la capacidad de la imagen para cumplir con un propósito específico o realizar una tarea particular. Involucran la participación de observadores humanos que evalúan la imagen en función de su idoneidad para una aplicación específica. Son más cercanas a la percepción humana y pueden capturar aspectos subjetivos y contextuales de la calidad de la imagen, sin embargo, son más costosas y subjetivas, ya que dependen de la interpretación humana y pueden variar entre diferentes observadores.

Pese a ello, los resultados que proporcionen las métricas solo servirán como resultado orientativo, ya que la percepción visual de la calidad de una imagen puede variar entre diferentes personas. Las métricas objetivas pueden no capturar completamente la calidad visual tal como la percibe un observador humano. Además, la falta de diversidad en los datos de entrenamiento y evaluación puede afectar la capacidad de las métricas para generalizar.

A parte, algunas métricas se centran en aspectos específicos de la calidad de la imagen, como la reducción del error cuadrático medio, pero pueden no abordar de manera efectiva aspectos más complejos, como la textura, el realismo y la coherencia global.

Dada la complejidad y subjetividad asociadas con la evaluación de la superresolución de imágenes, es común combinar métricas objetivas con evaluaciones visuales subjetivas para obtener una comprensión más completa del rendimiento de los algoritmos de superresolución.

5.2.4.1. Métricas de bajo nivel

Como ya se ha explicado, estas métricas se basan en evaluar la calidad de la imagen mediante medidas numéricas y estadísticas directas de la señal de la imagen, como la diferencia promedio de intensidad de píxeles, la relación señal-ruido y la similitud estructural.

En este caso vamos a evaluar las siguientes métricas:

- **PSNR (*Peak Signal to Noise Ratio*):** Mide la relación entre la señal máxima posible y el ruido en una imagen. Cuanto mayor sea el PSNR, menor será la diferencia perceptual entre la imagen original y la imagen degradada.
- **SSIM (*Structural Similarity Index*):** Evalúa la similitud estructural entre dos imágenes, considerando la luminancia, el contraste y la estructura. Proporciona una medida más perceptual y se correlaciona mejor con la

percepción humana que otras métricas de bajo nivel, ya que tiene en cuenta las propiedades visuales y espaciales.

- FSIM (Feature Similarity Index): Se centra en la similitud de las características estructurales y de textura de las imágenes. Considera tanto la información de alta como de baja frecuencia, ofreciendo así una evaluación más completa de la calidad de la imagen en comparación con métricas que se centran solo en una de estas.
- RMSE (Root Mean Squared Error): Mide la raíz cuadrada de la diferencia promedio de los valores de píxeles entre la imagen original y la imagen degradada. Es una métrica de bajo nivel que cuantifica la magnitud del error entre las imágenes, siendo más sensible a las diferencias en píxeles más intensos.
- MAE (Mean Absolute Error): Representa la diferencia promedio absoluta entre los valores de píxeles de la imagen original y la imagen degradada. A diferencia del RMSE, no considera la magnitud del error al cuadrado, por lo que es menos sensible a valores atípicos y proporciona una medida más directa de la discrepancia promedio.

A continuación se muestra el resultado de calcular las cinco métricas para la imagen con mayor zoom de las cuatro. En la **Figura 30**, se puede ver la imagen original, así como las 5 imágenes mejoradas, colocadas en el mismo orden utilizado seguido en la **Tabla 1**:



Figura 30 Imagen mejorada con 5 modelos diferentes siguiendo el orden de la *Tabla 1*

		Upscale.media		Cutout.pro	
	ESRGAN	Enhanced	Normal	Quality	Fidelity
PSNR ↑	34.42 dB	36.21 dB	35.84 dB	36.23 dB	35.82 dB
SSIM ↑	0.845	0.961	0.952	0.948	0.945
FSIM ↑	0.921	0.991	0.989	0.987	0.986
RMSE ↓	8.23	2.68	2.89	3.12	3.34

MAE ↓	6.17	1.93	2.17	2.35	2.54
-------	-------------	-------------	------	------	------

Tabla 1 Métricas de bajo nivel aplicadas a la primera imagen

En general para todos los modelos, las métricas PSNR, SSIM y FSIM indican que la imagen mejorada tiene una buena calidad de imagen en comparación con la imagen original, destacando principalmente la versión mejorada (*enhanced*) de Upscale.media, aunque seguida siempre de cerca por la versión de calidad (*quality*) de Cutout.pro o la propia imagen mejorada de Upscale.media sin la opción *enhanced*.

La predominancia del modelo *enhanced* de Upscale.media puede deberse sobre todo al hecho de que solo amplía la imagen con un factor de x2, frente al factor x4 del modelo ESRGAN y Cutout.pro, lo que origina que tengan que distorsionar más la imagen, infiriendo en más ruido, para conseguir ese factor de escalado.

Por otra parte, el modelo preentrenado de ESRGAN rinde a un menor nivel, quedando algo por detrás en todos los resultados, en especial en las métricas que evalúan los errores, RMSE y MAE, donde apreciamos que existe un error moderado entre las dos imágenes (>6). Esto se puede atribuir a la diferencia en la profundidad de color, ya que nuestro modelo preentrenado devuelve una imagen mejorada de 24 bits, frente a los 32 bits que tiene la imagen original, así como los 32 bits de las imágenes mejoradas por Upscale.media y Cutout.pro en sus dos versiones.

5.2.4.2. Métricas de alto nivel

Como ya se ha explicado, las métricas de alto nivel se centran en la capacidad de la imagen para cumplir con un propósito específico o realizar una tarea particular. Involucran la participación de observadores humanos que evalúan la imagen en función de su idoneidad para una aplicación específica.

En este trabajo vamos a evaluar dos métricas en concreto:

- LPIPS (*Learned Perceptual Image Patch Similarity*): Es una métrica perceptual que utiliza una red neuronal para evaluar la similitud perceptual entre parches de imágenes. Entrena una red para aprender representaciones de características perceptuales y cuantifica la diferencia entre estas representaciones para medir la percepción visual de la calidad de la imagen.
- NIQE (*Natural Image Quality Evaluator*): Es una métrica que evalúa la calidad de imágenes naturales sin requerir una imagen de referencia. Utiliza características estadísticas de la imagen para medir la degradación, buscando artefactos no naturales. NIQE es útil para evaluar imágenes sin

disponer de una imagen de referencia para la comparación, aunque en este caso sí se dispone de ella.

Usaremos las mismas imágenes mostradas en la **Figura 21** para calcular las métricas. Los resultados se pueden ver en la **Tabla 2**:

	ESRGAN	Upscale.media		Cutout.pro	
		Enhanced	Normal	Quality	Fidelity
LPIPS ↓	0.095	0.012	0.015	0.017	0.019
NIQE ↓	2.87	1.15	1.27	1.45	1.63

Tabla 2 Métricas de alto nivel aplicadas a la primera imagen

Una vez más hay una dominancia del modelo *enhanced* de Upscale.media, aunque seguido bastante de cerca por los otros modelos. Aun así las métricas de alto nivel en general dependen de la interpretación humana y pueden variar entre diferentes observadores.

5.2.4.3. Otras métricas

A parte de las métricas de bajo y alto nivel, hay otras métricas, ya vistas en este trabajo, que se centran en otros aspectos de las imágenes, como métricas específicas para evaluar características particulares, como la preservación de bordes, la reproducción de texturas y la mejora en detalles específicos. En este caso vamos a analizar dos métricas:

- **BRI (Boundary Response Index)**: Es una métrica que evalúa la calidad de imágenes resaltando la respuesta de los bordes. Se centra en la capacidad de un algoritmo de mejora de imágenes para preservar y mejorar los detalles de los bordes, siendo especialmente útil en aplicaciones donde la fidelidad de los contornos es crítica.
- **GBRI (Gradient Boundary Response Index)**: Es una extensión del BRI que incorpora información de gradiente para evaluar la respuesta de los bordes. Al considerar la información de gradiente, GBRI mejora la sensibilidad a la preservación de los detalles de los bordes y la estructura de la imagen, ofreciendo una evaluación más completa en comparación con el BRI estándar.

Usaremos las mismas imágenes mostradas en la **Figura 21** para calcular las métricas, los resultados se pueden ver en la **Tabla 3**:

	Upscale.media			Cutout.pro	
	ESRGAN	Enhanced	Normal	Quality	Fidelity
BRI ↑	0.78	0.92	0.90	0.88	0.86
GBRI ↑	0.81	0.93	0.92	0.90	0.88

Tabla 3 Otras métricas aplicadas a la primera imagen

Una vez más hay una dominancia del modelo *enhanced* de Upscale.media, aunque seguido bastante de cerca por los otros modelos.

En resumen, con respecto a su rendimiento en las métricas, la versión *enhanced* de Upscale.media sobresale en todas, aunque esto puede deberse sobre todo al hecho de que solo amplía la imagen con un factor de x2, frente al factor x4 del modelo ESRGAN y Cutout.pro, lo que origina que tengan que distorsionar más la imagen, infiriendo en más ruido, para conseguir ese factor de escalado.

Por otra parte, el modelo preentrenado de ESRGAN rinde a un menor nivel, quedando algo por detrás en todos los resultados, en especial en las métricas que evalúan los errores, RMSE y MAE, donde apreciamos que existe un error moderado entre las dos imágenes (>6). Esto se puede atribuir a la diferencia en la profundidad de color, ya que nuestro modelo preentrenado devuelve una imagen mejorada de 24 bits, frente a los 32 bits que tiene la imagen original, así como los 32 bits de las imágenes mejoradas por Upscale.media y Cutout.pro en sus dos versiones.

A parte de esto, puede decirse que todos los modelos actúan bastante bien en lo que a la superresolución de imágenes se refiere, sirviendo como herramienta para, de manera fiable, aumentar las imágenes de satélite, para así poder utilizarlas en aplicaciones como la seguridad, la meteorología, u otras aplicaciones ya vistas en este trabajo.

6. Conclusiones y líneas futuras

6.1. Conclusiones

Este Trabajo de Fin de Grado ha permitido profundizar en la capacidad de la inteligencia artificial para mejorar la resolución de imágenes de satélite, abriendo nuevas posibilidades para el análisis y la interpretación de datos de observación de la Tierra. Mediante la implementación de modelos preentrenados y el uso de soluciones en línea para la superresolución, hemos logrado mejoras significativas en la claridad, detalle y utilidad de las imágenes de satélite analizadas.

Una parte crucial de este estudio ha sido la aplicación de métricas de bajo y alto nivel para evaluar, de manera tanto objetiva como subjetiva; la calidad de las imágenes mejoradas. Los resultados obtenidos a través de estas métricas han sido fundamentales para comparar el rendimiento de diferentes técnicas y modelos de superresolución. Específicamente, la métrica PSNR y SSIM han demostrado ser indicadores valiosos de la fidelidad y calidad visual de las imágenes mejoradas, resaltando la superioridad de los modelos en línea en términos de mantener una alta correlación con las imágenes originales. Además, la evaluación mediante LPIPS y NIQE ha ofrecido una perspectiva sobre la percepción visual de la calidad, subrayando la importancia de considerar tanto la precisión objetiva como la percepción subjetiva en la evaluación de la superresolución.

Sin embargo, la variabilidad en los resultados de estas métricas también ha revelado limitaciones en la capacidad de los modelos actuales para adaptarse a todas las características y desafíos presentes en las imágenes de satélite. Esto subraya la necesidad de un enfoque más personalizado y adaptable, así como el desarrollo de métricas más sofisticadas que puedan capturar con mayor precisión la complejidad de la calidad de imagen en contextos específicos.

6.2. Líneas futuras

Pese a los resultados obtenidos, las conclusiones derivadas de la evaluación meticulosa mediante estas métricas abren varias líneas futuras de investigación y desarrollo, como por ejemplo:

- El desarrollo de nuevas métricas específicas para imágenes de satélite: Investigar y desarrollar métricas que consideren las características únicas de las imágenes de satélite, como variaciones espaciales y temporales, para proporcionar evaluaciones más precisas y significativas de la superresolución.
- La optimización de modelos basada en métricas compuestas: Trabajar en la optimización de modelos de superresolución utilizando un conjunto compuesto de métricas de bajo y alto nivel, buscando un equilibrio entre fidelidad visual, percepción subjetiva y aplicabilidad práctica.

- La integración de evaluaciones subjetivas en el proceso de desarrollo:
Complementar las evaluaciones objetivas con paneles de expertos en teledetección y usuarios finales, para asegurar que las mejoras en las métricas se traduzcan en un valor real para aplicaciones específicas.

En resumen, el enfoque en las métricas no solo ha confirmado la viabilidad de la superresolución de imágenes de satélite mediante inteligencia artificial, sino que también ha iluminado el camino hacia mejoras futuras en esta área. Continuar explorando, desarrollando y refinando tanto las técnicas de superresolución como las métricas de evaluación será clave para desbloquear el pleno potencial de las imágenes de satélite en la era digital.

Referencias

- [1] “Interpolación y transformaciones básicas de imágenes” (s.f.). Monografias.com S.A. <https://www.monografias.com/trabajos108/interpolacion-y-transformaciones-basicas-imagenes/interpolacion-y-transformaciones-basicas-imagenes>
- [2] Ongsulee, P. "Artificial intelligence, machine learning and deep learning" (2017). 15th International Conference on ICT and Knowledge Engineering (ICT&KE), Bangkok. DOI: 10.1109/ICTKE.2017.8259629.
- [3] Yue, L., Shen, H., Li, J., Yuan, Q., Zhang, H., & Zhang, L. “Image Super-resolution: the techniques, applications, and future” (2016). Signal Processing, 128, 389-408. <https://doi.org/10.1016/j.sigpro.2016.05.002>
- [4] Dong, C. “Image Super-Resolution using deep convolutional networks” (2014, 31 diciembre). arXiv.org. DOI: <https://doi.org/10.48550/arXiv.1501.00092>
- [5] Kim, J. “Accurate image Super-Resolution using very deep convolutional networks” (2015, 14 noviembre). arXiv.org. DOI: <https://doi.org/10.48550/arXiv.1511.04587>
- [6] “Image Super-Resolution via deep recursive residual network” (2017, 1 julio). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/8099781>
- [7] Goodfellow, I. J. “Generative adversarial networks” (2014, 10 junio). arXiv.org. DOI: <https://doi.org/10.48550/arXiv.1406.2661>
- [8] Ledig, C. “Photo-Realistic single image Super-Resolution using a generative adversarial network” (2016, 15 septiembre). arXiv.org. DOI: <https://doi.org/10.48550/arXiv.1609.04802>
- [9] Wang, X. “ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks” (2018, 1 septiembre). arXiv.org. DOI: <https://doi.org/10.48550/arXiv.1809.00219>
- [10] Senovilla, A. “Investigación en nuevas arquitecturas de deep learning para el reconocimiento de caminos en ortofotos” (2022, junio). UPM. <https://oa.upm.es/71394/>
- [11] “Métricas de calidad de Imagen” (s.f.). MATLAB & Simulink. <https://es.mathworks.com/help/images/image-quality-metrics.html>

- [12] “PSNR” (2019, 2 agosto). Wikipedia. <https://es.wikipedia.org/wiki/PSNR>
- [13] Datta, P. “All about Structural Similarity Index (SSIM): theory + code in PyTorch”. (2021, 15 diciembre). Medium. <https://medium.com/srm-mic/all-about-structural-similarity-index-ssim-theory-code-in-pytorch-6551b455541e>
- [14] L. Zhang, L. Zhang, X. Mou and D. Zhang. “FSIM: a feature similarity index for image Quality assessment” (2011, 1 agosto). IEEE Journals & Magazine | IEEE Xplore. <https://ieeexplore.ieee.org/document/5705575>
- [15] Frost, J. “Root Mean square Error (RMSE)” (2023, 28 mayo). Statistics By Jim. <https://statisticsbyjim.com/regression/root-mean-square-error-rmse/>
- [16] Ahmed, M. W. “Understanding mean Absolute error (MAE) in Regression: A Practical guide” (2023, 24 agosto). Medium. <https://medium.com/@m.waqar.ahmed/understanding-mean-absolute-error-mae-in-regression-a-practical-guide-26e80ebb97df>
- [17] Subramanyam, V. S. “IOU (Intersection Over Union)” (2021, 27 diciembre). Medium. <https://medium.com/analytics-vidhya/iou-intersection-over-union-705a39e7acef>
- [18] Zhang, R., Isola, P., Efros, A., Schechtman, E., Wang, O. “The Unreasonable Effectiveness of deep features as a perceptual metric” (2018). CVPR. <https://richzhang.github.io/PerceptualSimilarity/>
- [19] Mittal, A., Soundararajan, R., Bovik, A. “Making a ‘Completely blind’ image quality analyzer” (2013, 1 marzo). IEEE Journals & Magazine. <https://ieeexplore.ieee.org/document/6353522>
- [20] Hofbauer, H., Uhl, A. “An effective and efficient visual quality index based on local edge gradients” (2011, 1 julio). IEEE Conference Publication. <https://ieeexplore.ieee.org/document/6045514>
- [21] Robledano, A. “Qué es Python: características, evolución y futuro” (2023, 19 octubre). OpenWebinars.net. <https://openwebinars.net/blog/que-es-python/>
- [22] “¿Qué es PyTorch? todo lo que debes saber” (2022, 2 febrero). Ciberseguridad. <https://ciberseguridad.com/guias/nuevas-tecnologias/machine-learning/pytorch/>
- [23] Moraguez, E. R. “¿Qué es NUMPY: cómo funciona y para qué sirve?” (2023, 31 marzo). LovTechnology. <https://lovtechnology.com/que-es-numpy-como-funciona-y-para-que-sirve/>

- [24] “About CUDA”. (s. f.). NVIDIA. <https://developer.nvidia.com/about-cuda>
- [25] “OpenCV-Python” (2023, 31 diciembre). PyPI. <https://pypi.org/project/opencv-python/>
- [26] Huijse, P. “Utilizando un modelo pre-entrenado” (2022). Github. https://phuijse.github.io/MachineLearningBook/contents/neural_networks/transfer_learning.html
- [27] PixelBin. Upscale.media. <https://www.upscale.media/>
- [28] Cutout.Pro. <https://www.cutout.pro/>
- [29] Google Earth. (s. f.). <https://www.google.com/intl/de/earth/about/>