

# Superresolución: descripción técnica y comparativa entre los diferentes métodos.

Mohsin Riaz

**Abstract**— El objetivo de este proyecto es explorar el panorama actual de la superresolución, describir de manera técnica su funcionamiento. En este artículo, se intenta explorar las diferentes técnicas de la superresolución, investigar y analizar las técnicas del estado del arte. Se empezará por la parte de la investigación para aprender el funcionamiento, la matemática y la lógica detrás de la superresolución. Del conjunto de diferentes técnicas se escogerán las más interesantes y se procederá a su análisis computacional y posteriormente una comparativa entre los resultados obtenidos.

**Keywords**—Visión por computador, súper resolución (SR), restauración, Deep Learning, Redes neuronales convolucionales (CNN), U-Net, análisis y comparativa.

## 1. INTRODUCCIÓN

La superresolución (SR) es una de las técnicas y algoritmos de procesamiento de imágenes que ha recibido un gran interés por parte de la comunidad investigadora en los últimos años. La superresolución tiene como objetivo incrementar la resolución espacial<sup>1</sup> a partir de una imagen o una serie de imágenes de baja resolución (LR) mejorando su calidad visual e incorporando detalles de alta frecuencia (véase Fig.1). A esta técnica también se le puede llamar interpolación, muestreo superior, ampliación o zoom. La idea básica detrás de SR es combinar la información no redundante contenida en múltiples *frames* de baja resolución para generar una imagen de alta resolución.

La superresolución también tiene múltiples aplicaciones importantes en diferentes dominios, por ejemplo: Reconocimiento facial en cámaras de videovigilancia, imágenes astronómicas, contrarrestar los errores en las cámaras de fotografía, imágenes médicas y *remastering* de videojuegos antiguos para mejorar las texturas y muchos más [2].

Para aumentar la resolución espacial de una imagen o un vídeo, una forma sencilla es aumentar la densidad del sensor reduciendo el tamaño del sensor. Sin embargo, a medida que disminuye el tamaño del sensor, la cantidad de luz que incide en cada sensor también disminuye, lo que provoca el llamado ruido de disparo<sup>2</sup>. Además, cuanto más densidad tiene un sensor más elevado es su coste, por ello, la limitación de hardware es la que restringe la resolución espacial de una imagen. Ante las limitaciones del hardware, las diferentes técnicas de SR se presentan como una alternativa muy atractiva. [1]

<sup>1</sup>La resolución espacial se refiere a la densidad de píxeles en una imagen y se mide en píxeles por unidad de área.

<sup>2</sup>El ruido de disparo en los dispositivos electrónicos consiste en fluctuaciones aleatorias de la corriente, causadas por el hecho de que la corriente se transporta en cargas discretas (electrones).

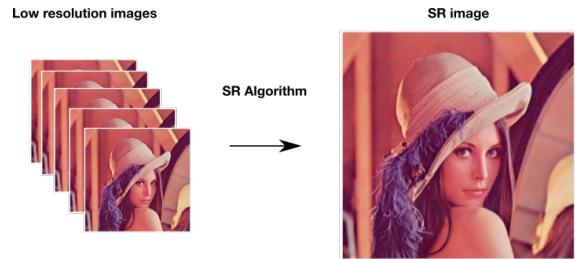


Fig. 1: Incremento de la resolución espacial a partir de una serie de imágenes de baja resolución para obtener una imagen de alta resolución de alta calidad.

### 1.1. SISR: superresolución de imagen única

El problema de SISR se ha estudiado ampliamente en la literatura utilizando una variedad de técnicas basadas en el *deep learning*. Se pueden clasificar los métodos existentes en diferentes grupos de acuerdo con las características más distintivas de los diseños de sus modelos. La taxonomía general de las SISR se muestra en la Fig.2. En el siguiente apartado se discutirán unos de los modelos del estado del arte.

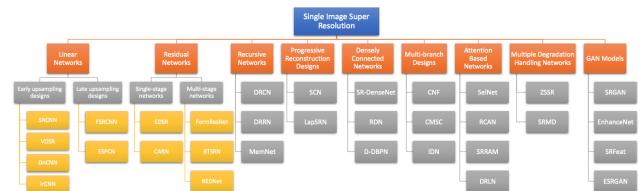


Fig. 2: La taxonomía de las técnicas de superresolución de imagen única existentes basadas en las características más distintivas.

## 2. ESTADO DEL ARTE

Las diferentes técnicas de la superresolución se pueden categorizar de la siguiente manera: (i) Técnicas de dominio de frecuencia. (ii) Métodos basados en la reconstrucción teniendo información previa sobre la apariencia del objeto. (iii) Algoritmos de aprendizaje que mapean entre las imágenes LR y SR. En los últimos años, gracias a los avances en la rama de la investigación de las redes neuronales, los modelos de aprendizaje SR basados en una sola imagen han experimentado increíbles mejoras al aplicar las *deep convolutional neural networks*(CNNs) [2].

*Super-Resolution Convolutional Neural Network* (SRCNN) es uno de los primeros modelos que consiguió emplear solamente las capas neuronales para poder obtener imágenes SR. Este trabajo puede considerarse como el trabajo pionero en la SR basada en el *deep learning* que inspiró varios intentos posteriores en esta dirección. Hay un total de tres capas convolucionales y dos ReLU, apiladas juntas linealmente. La primera capa convolucional se denomina extracción de parches o extracción de características que crea los mapas de características a partir de las imágenes de entrada. La segunda capa convolucional se llama mapeo no lineal que convierte los mapas de características en vectores de características de alta dimensión. La última capa convolucional agrega los mapas de características para generar la imagen final de alta resolución. El SRCNN es una red entrenable end-to-end que minimiza la diferencia entre las imágenes SR y las imágenes HR

de validación utilizando la función de pérdida de error cuadrático medio (MSE).

*Enhanced Super-Resolution Generative Adversarial Networks* (ESRGAN) Como las *Generative Adversarial Networks* (GAN) en general, ESRGAN se compone de dos algoritmos: uno que genera una imagen y otro que determina si la imagen es real o falsa. ESRGAN compara la imagen generada con una imagen real e intenta determinar cuál es más real. El enfoque de la competitividad entre las dos redes obliga al algoritmo a generar eventualmente detalles con bordes más nítidos y texturas más realistas. El modelo ESRGAN busca la variaciones de color entre los píxeles de una imagen e intenta incorporar detalles que cree que debería existir en función de las imágenes con las que se ha entrenado previamente.

### 3. DESARROLLO Y EXPERIMENTACIÓN

#### 3.1. Dataset

En este proyecto se pretende realizar una comparativa cualitativa y cuantitativa entre la interpolación bicúbica, SISR mediante una red neuronal U-Net y un modelo de ESRGAN ya pre-entrenado. En este proyecto se hará servir el *dataset Set5*, el modelo ESRGAN utilizado viene entrenado con los datos **General100** que contiene el *dataset*. Para evaluar tanto la interpolación bicúbica, la red U-Net o el modelo ESRGAN se utilizarán las imágenes del conjunto **Urban100**.



Fig. 3: Unas de las imágenes del *dataset Urban100*.

#### 3.2. Downsampling

El *dataset* seleccionado contiene imágenes con composiciones variadas que presentan al modelo escenarios suficientemente variados para generar imágenes de alta resolución más naturales con texturas más ricas. El modelo ESRGAN previamente se ha entrenado con canales RGB y aumentando el conjunto de datos de entrenamiento con *flips* horizontales aleatorias y rotaciones de 90 grados. Para poder evaluar las diferentes técnicas de SR hace falta primero reducir el tamaño de las imágenes para luego intentar aplicar la técnica deseada y comparar los resultados. Las imágenes del *dataset Urban100* contiene imágenes HR del tamaño de media de  $1024 \times 1024$  píxeles, por ello se ha reducido el tamaño de las imágenes con un factor de  $\times 4$  y para obtener estas imágenes se han eliminado las filas y columnas impares dos veces (Fig.4).

#### 3.3. Experimentación

Para entrenar la red U-net y poder evaluar la eficacia del modelo ESRGAN se ha tenido que implementar el método de SR más básico que es la interpolación bicúbica. A continuación, se presentan las técnicas que se han utilizado en este proyecto y se explica como funcionan estos algoritmos.

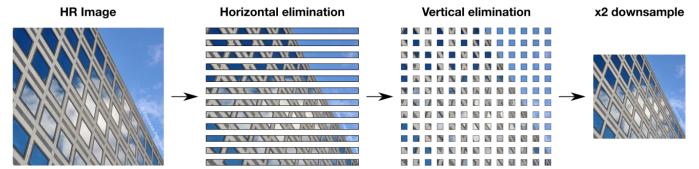


Fig. 4: Downsampling eliminando filas y columnas impares de la imagen.

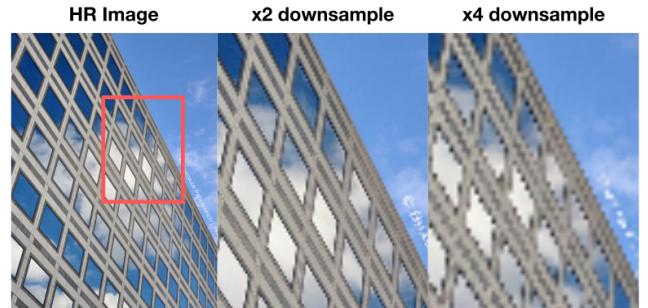


Fig. 5: Resultados obtenidos apartir de la aplicación del algoritmos de downsampling.

##### 3.3.1. Interpolación bicúbica

La interpolación bicúbica es una técnica de aumento de resolución de imágenes que intenta ajustar una superficie entre cuatro puntos de esquina utilizando una función polinomial de tercer grado. Para poder calcular la interpolación bicúbica de una imagen, hace falta especificar los valores de intensidad y la derivada horizontal, vertical y diagonal en los cuatro puntos de las esquinas. La superficie interpolada descrita por el polinomio de tercer grado se presenta en la ecuación 1 y su representación visual en la Fig.6.

$$f(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j \quad (1)$$

En este proyecto, se intentó implementar la función de la interpolación bicúbica de manera manual, pero tras muchos intentos de optimizaciones no se logró obtener una versión del algoritmo que pudiera ejecutar su tarea en un tiempo razonable. Por ello, se ha tenido que utilizar la implementación de la función *resize* con el resampler bicúbico de la librería *skimage*. Para poder calcular la interpolación bicúbica el proceso es relativamente sencillo. Para empezar se carga imagen del disco, se utiliza el OpenCV para convertir el color de BGR a RGB, a continuación se llama la función *resize* de *skimage* para aumentar la resolución utilizando el resampler bicúbico y se guarda la imagen en el disco.

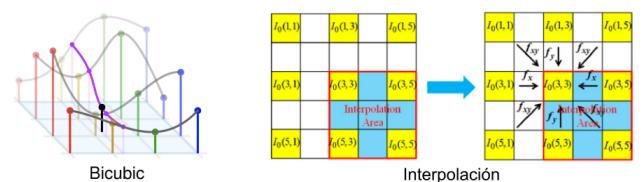


Fig. 6: Representación del proceso de la interpolación bicúbica.

### 3.3.2. Modelo UNET

En este proyecto, como segunda técnica para la SR que se ha utilizado es una red neuronal U-Net. Este tipo de redes suelen ser redes completamente convolucionales y se adaptan muy bien al tamaño de la entrada. Normalmente se suelen utilizar para problemas de segmentación pero con los cambios apropiados se pueden utilizar para añadir detalles de baja frecuencia en la imagen de entrada, así aumentando su calidad. Se llaman U-Net por la forma que tiene la red (véase Fig.7). Una red U-Net consta de operación de convolución, *maxpooling*, activación de ReLU, concatenación y capas de *upsampling* y todo esto dividido en tres secciones: *Encoder*, *bottleneck* y *Decoder* [3][4].

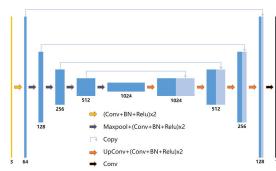


Fig. 7: Arquitectura de la red neuronal U-Net.

Para poder aplicar esta técnica al nuestro *dataset* se ha buscado una implementación de esta red neuronal y a continuación se ha procedido a entrenarla con los datos que previamente se habían generando a partir de interpolación bicúbica. Como entrada a la red, se le ha proporcionado la imagen que previamente ha sido reducida por un factor de  $\times 4$  y luego aumentada mediante la interpolación bicúbica. Como *loss function* se ha utilizado el *MSE* entre la salida y la imagen *groundtruth* y como optimizador se ha utilizado el *Adam* que ya viene implementado en *Keras*. Para obtener la imagen de salida con la calidad deseada se ha procedido a ajustar varios hiperparámetros para llegar a resultados óptimos. Estos parámetros han sido el *Batch size*, *Epochs*, *Learning rate* y el número de *Maxpoolings*. Por cuestiones de tiempo se ha tenido que utilizar bloques de  $256 \times 256$  de las imágenes del *dataset* y aparte se ha tenido que utilizar la escala de grises para obtener resultado en un tiempo razonable. Lamentablemente, tras muchos intentos, no se llegó a poder entrenar la red para que pudiera producir buenos resultados. Sin embargo, se han utilizado las mismas métricas de calidad para evaluar los resultados que la interpolación bucuática y el modelo ESRGAN.

### 3.3.3. Modelo ESRGAN

Como se ha visto en la sección de la introducción, el ESRGAN es un *Generative Adversarial Networks* que básicamente son dos redes neuronales compitiendo entre sí constantemente para ‘engaños’ a la otra. Estos modelos cuando bien entrenados, tienen la capacidad de producir resultados espectaculares pero requieren de mucho tiempo, recursos y datos para llegar a ese nivel. En este proyecto se ha procedido a buscar un modelo ya pre-entrenado con el *dataset set5*. Se ha utilizado la plataforma *Colaboratory* para probar el modelo y obtener las imágenes SR. A continuación se han calculado las métricas de calidad para posteriormente compararlas con el resto de técnicas [5].

### 3.3.4. Desarrollo

Para poder realizar el proyecto se ha utilizado la plataforma *Colaboratory* de Google. Los algoritmos y los *scripts* se han implementado en *Colab Notebooks*. Otro motivo por el que se ha

optado por utilizar esta plataforma es la existencia de *Colab Notebooks* con los modelos ESRGAN ya implementados y entrenados.

Antes de empezar con las implementaciones de los algoritmos se ha escogido el *dataset* que se ha mencionado en los apartados anteriores. A continuación, se ha creado una nueva libreta *Colab* para empezar a programar los algoritmos para calcular las diferentes métricas y algoritmos para calcular el interpolación bicubica.

## 3.4. Métricas

Los índices utilizados para la fase de la experimentación se han obtenido mediante su aplicación a los canales RGB de las imágenes. Generalmente, en los algoritmos SR los resultados se evalúan mediante varias métricas de evaluación de la calidad de las imágenes (*IQA: Image quality assessment scores*). Estas métricas se pueden clasificar en dos categorías: métricas subjetivas y métricas objetivas. Las métricas subjetivas se basan en la evaluación perceptiva por el humano y las métricas objetivas se basan en modelos computacionales que intentan evaluar la calidad de la imagen. En este proyecto se ha experimentado con las métricas ampliamente utilizadas que son las SSIM y PSNR que se explícan a continuación.

### 3.4.1. PSNR: Peak signal-to-noise ratio

La relación pico-señal-ruido (PSNR)(2) es una métrica objetiva de uso común para medir la calidad de reconstrucción de una transformación con pérdidas. PSNR es inversamente proporcional al logaritmo del error cuadrático medio (MSE) entre la imagen original y la imagen generada.

$$PSNR = 10 * \log_{10} \left( \frac{L^2}{MSE} \right) \quad (2)$$

En la ecuación 2, L es el valor de píxel máximo posible para imágenes RGB de 8 bits, es 255. Dado que PSNR solo se preocupa por la diferencia entre los valores de los píxeles, no representa tan bien la calidad de percepción.

### 3.4.2. SSIM: Structural similarity

La similitud estructural (SSIM) es una métrica subjetiva utilizada para medir la similitud estructural entre imágenes, basada en tres comparaciones relativamente independientes, luminancia, contraste y estructura. Dado que SSIM evalúa la calidad de la reconstrucción desde la perspectiva del sistema visual humano, cumple mejor con los requisitos de la evaluación perceptual. La representación comúnmente utilizada de la ecuación SSIM es la que se muestra a continuación:

$$SSIM(I, \hat{I}) = \frac{(2\mu_I\mu_{\hat{I}} + C_1)(\sigma_{I\hat{I}} + C_2)}{(\mu_I^2 + \mu_{\hat{I}}^2 + C_1)(\sigma_I^2 + \sigma_{\hat{I}}^2 + C_2)} \quad (3)$$

## 4. RESULTADOS

En este apartado se presentan los resultados y diferentes índices y métricas para poder efectuar una comparación para determinar el rendimiento de cada técnica utilizada para la SR.

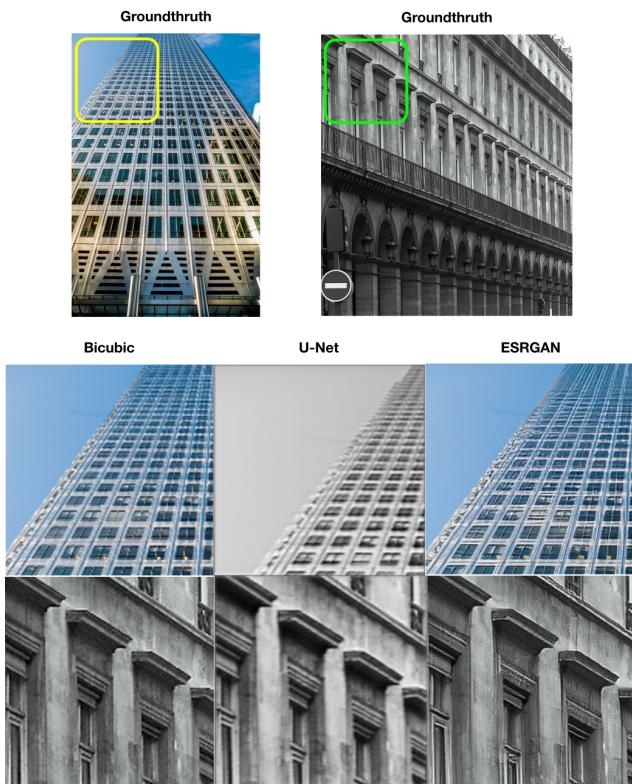


Fig. 8: Los resultados obtenidos a partir de aplicar las diferentes técnicas de SR.

Resultados (PSNR / SSIM)		
Modelo	PSNR	SSIM
Bicubic	18.33	0.53
U-Net	18.85	0.65
ESRGAN	21.49	0.65

Taula 1: Relación PSNR / SSIM de los resultados obtenidos.

#### 4.1. Resultados cualitativos

Como se puede observar en la Fig.8, la técnica que mejor resultados presenta es el modelo ESRGAN. Observando la tabla 1 vemos que, efectivamente el modelo ESRGAN tiene un índice SSIM por encima del 50 % y el PSNR más alto. Como ya era de esperar, el modelo bicúbico es el que peor resultados ha presentado ya que el problema que se quiere resolver, el de aumentar la información inexistente, esta mal planteado. Finalmente, el modelo de la U-Net tiene un índice alto de SSIM pero el PSNR es al nivel del bicúbico. Esto nos indica que aunque estructuralmente la imagen se parece más al original, la información contenida es muy diferente o ruidosa.

##### 4.1.1. Resultados de la red U-Net

Para poder entrenar la red U-Net se ha optado por utilizar los siguientes hiperparámetros:

- BATCH\_SIZE = 32
- EPOCHS = 10
- LERNING RATE = 0.0015
- MAXPOOLING = 16 → 32 → 64 → 128 → 256

A partir de estos hiperparámetros y el *dataset* seleccionado se han obtenido los siguientes datos sobre el progreso del entrenamiento de la red.

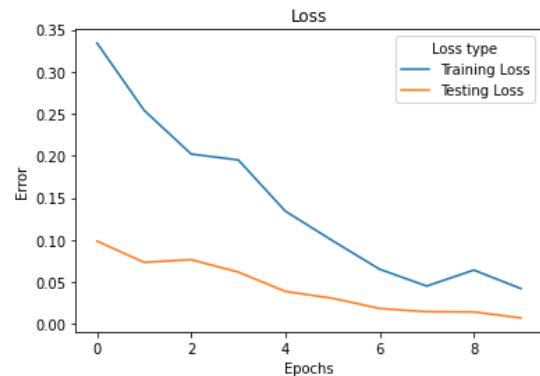


Fig. 9: La evolución del error durante el entrenamiento de la red U-Net.

Tras muchas iteraciones y modificaciones de los hiperparámetros, la versión que representa los datos de la Fig.9 ha sido la mejor de todas. Como ya se ha mencionado anteriormente, la red no pudo llegar al nivel de entrenamiento en el que fuera capaz de producir buenos resultados.

## 5. CONCLUSIONES

Este documento presenta una descripción técnica y una comparativa entre las técnicas de SR por interpolación bicúbica, mediante U-Net y ESRGAN para aumentar la escala y la calidad de imágenes. Se han descrito unas de las técnicas de SR existentes y se ha optado por profundizar en las tres técnicas mencionadas anteriormente. Se ha presentado el *dataset* a usar y a continuación una técnica sencilla para poder reducir el tamaño de las imágenes para poder entrenar las redes.

Se ha visto a más profundidad el funcionamiento de las técnicas de SR y se explicado los pasos seguido para poder implementarlas. Se han expuestos problemas encontrados durante la fase de experimentación. Se ha visto brevemente las métricas existentes para evaluar los resultados de los modelos de SR. Finalmente se han presentado los resultados obtenidos y se ha realizado una comparativa para determinar cual de las tres técnicas ha funcionado la mejor.

## REFERENCIAS

- [1] Sun, W., & Chen, Z. (2020). Learned Image Downscaling for Upscaling Using Content Adaptive Resampler. *IEEE Transactions on Image Processing*, 29, 4027-4040.
- [2] Z. Wang, J. Chen and S. C. H. Hoi, "Deep Learning for Image Super-resolution: A Survey," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, doi: 10.1109/TPAMI.2020.2982166.
- [3] Lu, Z., & Chen, Y. (2020). Dense U-net for super-resolution with shuffle pooling layer. *arXiv preprint arXiv:2011.05490*.
- [4] Vojtekova, Antonia, et al. "Learning to denoise astronomical images with U-nets." *Monthly Notices of the Royal Astronomical Society* 503.3 (2021): 3204-3215.
- [5] Wang, Xintao, et al. "Ergan: Enhanced super-resolution generative adversarial networks." *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 2018.