Métodos para súper-resolución de imágenes

Madrigal-Custodio Jesús A., Tevera-Ruiz Alejandro, Torres-Martínez Luis Á.
Departamento: Robótica y Manufactura Avanzada
Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional

Resumen

En el presente documento se explican los fundamentos, metodología y proceso de implementación para el desarrollo de algoritmos de súper resolución bajo diversos enfoques con el objetivo...

Palabras Clave

Súper Resolución, Redes Convolucionales, Inteligencia Artificial

1. Introducción

1-A. SRGAN

2. Antecedentes

Bajo un enfoque *clásico*, existen tres formas de mejorar la resolución de una imagen:

- Amplificación de detalles existentes
- Suma de múltiples frames
- Único frame

Para el primero de ellos, se realiza una amplificación de las frecuencias altas (donde se encuentran los detalles existentes de la imagen) dada la variación local entre los pixeles vecinos. La amplificación de detalles existentes resulta bastante sencillo de aplicar. Sin embargo, ante imágenes con una cantidad considerable de ruido puede no ser la mejor opción a tomar. Además, al potencializar las frecuencias ya existentes de la imagen, el resultado estará definido por el detalle previo en la imagen de entrada.

El segundo de los métodos considera que el frame de alta resolución es el resultado de una secuencia de frames de baja resolución que permiten obtener las frecuencias altas de la imagen resultante para mejorar su resolución. Esto es conveniente cuando ya se cuenta con el conjunto de imágenes requeridas y se planea realizar una reconstrucción de la imagen en una mejor resolución.

Por otro lado, el tercer método basado en un único frame o imagen busca aproximar las frecuencias altas (detalles) que no se encuentra en la entrada del algoritmo y que evidentemente no puede obtenerse sólo amplificando las frecuencias altas como lo que ocurre con el primero de los métodos.

2-A. Interpolación

Para mejorar la resolución se busca aumentar la densidad de pixeles de la imagen con el objetivo de hacer la imagen más grande y mejorar sus detalles a partir de la predicción de pixeles que no se encuentran en la imagen visiblemente, pero que podrían aproximarse al buscar que se mantenga una consistencia en la imagen modificada de acuerdo a la vecindad de los pixeles.

Esto permite proponer el uso de algoritmos de interpolación que buscan predecir los 3 pixeles vecinos y con ello aumentar la densidad de pixeles de la imagen de entrada. Con base en [1], dichos algoritmos pueden agruparse en dos categorías: adaptativos y no adaptativos. Los primeros cambian dependiendo de lo que se está interpolando (bordes o texturas suaves) pixel por pixel con el objetivo de minimizar los errores antiestéticos de los algoritmos de interpolación como el desenfoque o pérdida de detalles en regiones evidentes. Ejemplos de ellos pueden ser los softwares de licencia como *Qimage, PhotoZoom Pro, Genuine Fractals, etc*.

Mientras que los métodos no adaptativos tratan todos los pixeles por igual dada la predicción de un pixel central de acuerdo a sus pixeles adyacentes. Esto involucra que entre más vecinos se consideren en la interpolación, una mejor aproximación se tendrá del pixel a predecir, pero de manera proporcional aumentarán los recursos computacionales necesarios. Dentro de los algoritmos se incluyen: *vecino más cercano, bilineal, bicúbica, spline, entre otros.*

A continuación se describirán algunos de los algoritmos no adaptativos para interpolación que serán utilizados en los diferentes métodos para *Súper Resolución*:

- **Vecino más cercano** - Dado un pixel considera sólo un pixel adyacente para la interpolación, lo que resulta en un menor tiempo de procesamiento pero resultados poco consistentes al observar al conjunto de pixeles interpolados.
- **Bilineal** - Considera una vecindad 2x2 correspondiente al pixel a predecir con su correspondiente promedio ponderado de acuerdo a la distancia del pixel desconocido. Esto da como resultado un aspecto más suave que el vecino más cercano.

- **Bicúbica** - Valora una vecindad 4x4 de pixeles conocidos para la predicción del pixel central considerando el mismo procedimiento de la interpolación bilineal. Como resultado, se alcanzan imágenes más nítidas que los métodos anteriores. Logrando así un equilibrio entre la resolución de salida y el tiempo de procesamiento. Lo anterior promueve que sea un estándar en muchos programas de edición de imágenes, controladores de impresoras e interpolación en cámaras.

En la Figura 2.1 se presentan los tres algoritmos no adaptativos más utilizados. Observe que las definiciones dadas anteriormente coinciden con los resultados expuestos donde el *vecino más cercano* resulta poco útil para predecir los pixeles intermedios mientras que *bilineal* o *bicúbica* predicen de mejor manera los pixeles intermedios con un poco más de detalle para en el caso del último algoritmo.

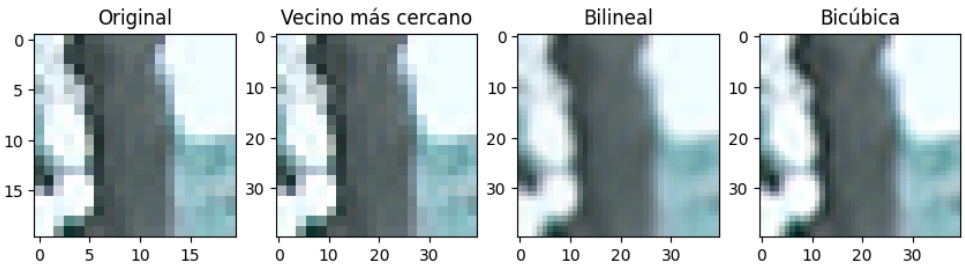

Figura 2.1. Algoritmos de interpolación no adaptativos

Todos los interpoladores no adaptativos intentan encontrar un equilibrio óptimo entre tres efectos no deseados: halos de borde, desenfoque y *aliasing*. En la Figura 2.2 puede observarse el efecto para cada caso.

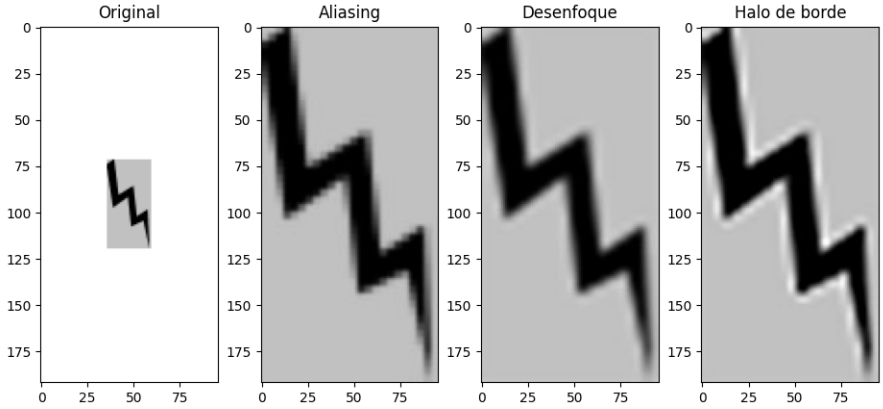

Figura 2.2. Efectos de interpolación

Incluso los interpoladores no adaptativos más avanzados siempre tienden a aumentar o disminuir algunos de los efectos a expensas de los otros dos, por lo tanto uno será más evidente.

En contraste, los interpoladores adaptativos pueden o no producir los efectos mencionados aunque generalmente inducen texturas que no son de la imagen o pixeles extraños a pequeña escala.

2-B. Example Based Super Resolution

Como puede observarse la interpolación soluciona parcialmente el problema de *Súper Resolución*, pero tiene como consecuencia los efectos mencionados. En particular, el desenfoque resulta contraproducente al intentar mejorar los detalles de una imagen. Por lo mismo, en los algoritmos clásicos de *Súper Resolución* se utiliza la interpolación únicamente para aumentar la densidad de los pixeles y aproximar la imagen de salida como una imagen más grande con un determinado factor de escalado, pero con los detalles de desenfoque que producen los algoritmos de interpolación no adaptativos.

Para solucionarlo, algunos autores proponen realizar un postprocesado a la imagen interpolada para incluir los detalles faltantes y con ello mejorar visiblemente la resolución de los bordes de la imagen.

En particular, [2] propone un parchado de la imagen reescalada a partir de un conjunto de entrenamiento o diccionario de parches en pares de alta y baja resolución. Dichos parches permiten construir una imagen con frecuencias altas que no están en la imagen de entrada con el objetivo de sumar la imagen original interpolada con las frecuencias altas que buscan mejorar

su resolución al realzar sus detalles. En la Figura 2.3 puede observarse de manera específica el algoritmo propuesto basado en el parchado de la imagen de entrada mediante un algoritmo de predicción.

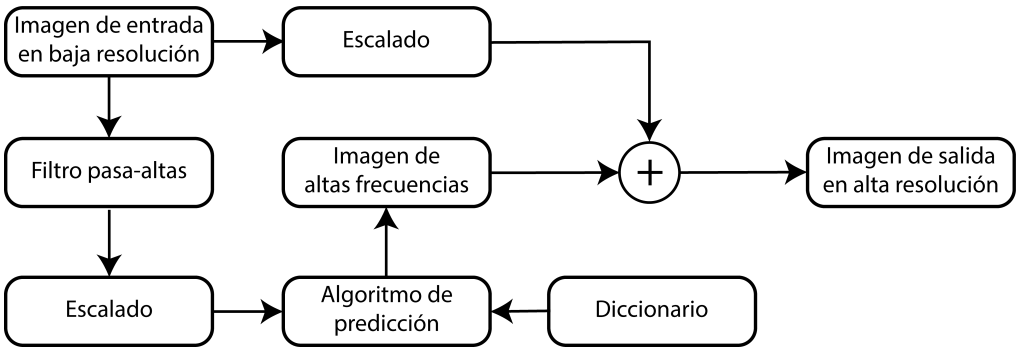

Figura 2.3. Algoritmo de súper resolución

2-B1. Diccionario

El algoritmo de *Súper Resolución* [2] opera bajo la premisa que la relación de predicción entre los parches de alta y baja resolución es independiente del contraste de la imagen. Esto también resulta ventajoso, ya que el diccionario no necesita ser de imágenes similares a las que se van a reconstruir para mejorar la calidad de resolución tal como comenta [3]. Esto resulta en un algoritmo general aplicable a cualquier tipo de imagen y escalable respecto al tamaño de la base de entrenamiento.

Desde el punto de vista de almacenamiento, los parches de baja resolución carecen de detalle y por lo tanto predominan las frecuencias bajas, las cuales son irrelevantes en su uso para la predicción de detalles y por lo tanto resulta información necesaria dentro del proceso. Por lo mismo, es aconsejable aplicar un filtro pasa-altas a cada parche con el objetivo de dejar sólo la información útil para el algoritmo (detalles).

Por otra parte, para que el diccionario sea funcional sin importar el tipo de imagen a reconstruir, se busca normalizar cada pareja de parche con el objetivo de mantener su relación intrínseca. De acuerdo con [2], los parches de baja resolución se recomiendan con un tamaño de 7x7 pixeles mientras que los de alta resolución serán de 5x5 todos con centro en el mismo pixel para mantener la relación tal como se presenta en la Figura 2.4.

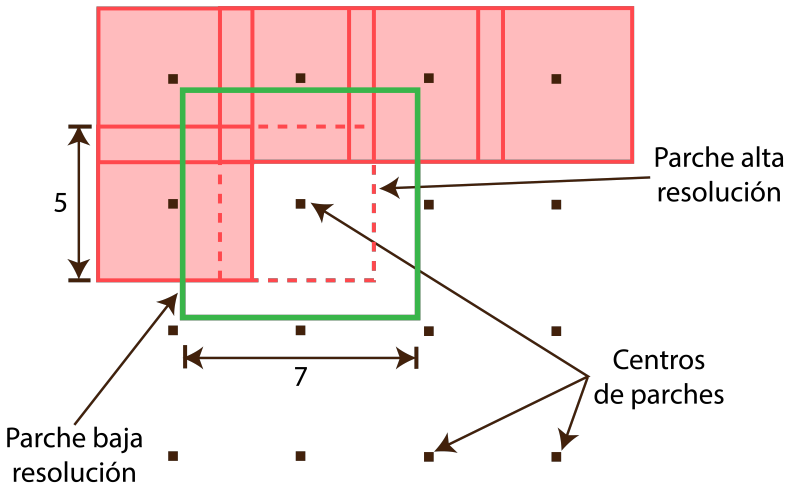

Figura 2.4. Adquisición de parches de cada imagenes

En específico, los parches de baja resolución deben reordenarse como un vector en $\mathbb{R}^{1\times 49}$ concatenado con la primera fila y primera columna del parche de alta resolución, resultando en un vector en $\mathbb{R}^{1\times 59}$. Lo último es necesario para considerar la superposición de los parches de alta resolución de la imagen a reconstruir.

Cabe destacar que la base de entrenamiento está en RGB por lo que bastará con el procedimiento antes mencionado para guardar los pares de parches en algún archivo de fácil acceso.

2-B2. Algoritmo de predicción

Una vez teniendo el diccionario o base de entrenamiento es necesario establecer el algoritmo de predicción para generar esos detalles no visibles en la imagen original. Para ello, la imagen de entrada (en baja resolución) debe ser pre-procesada mediante

un filtro pasa-altas para eliminar información innecesaria y posteriormente realizar un proceso de escalado mediante algún algoritmo de interpolación para aumentar sus dimensiones y densidad de los pixeles tal como se presenta en la Figura 2.3. Observe que para ese punto, la imagen de entrada cuenta con los píxeles que representan sus bordes o detalles que han sido escalados con el objetivo de tener una base a partir de la cual se va a reconstruir con más detalle la imagen de salida.

Puesto que [2] propone que la reconstrucción sea con parches, se debe realizar una búsqueda para elegir al parche de la base de datos que se aproxime más al parche de entrada con el objetivo de realizar la reconstrucción. En la Figura 2.5 se observa el algoritmo propuesto por *Freeman et al* para la predicción de parches.

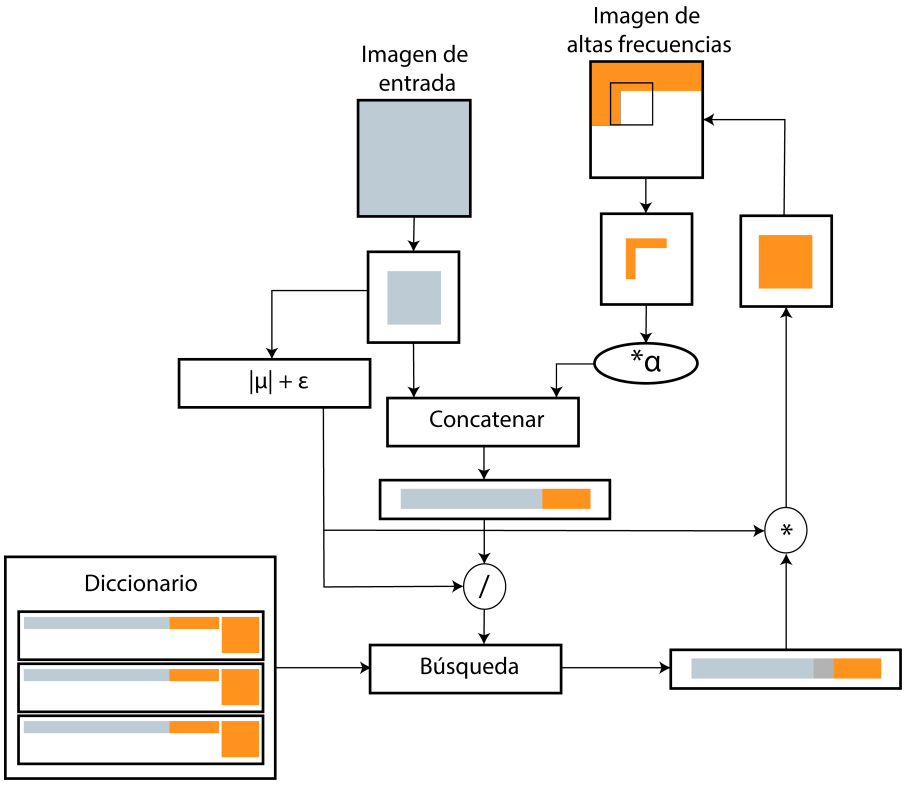

Figura 2.5. Algoritmo de predicción de imagen de frecuencias altas

Observe que la imagen de entrada se irá segmentando en los parches de baja resolución (color gris-verdoso). Estos parches se calculará su media absoluta $|\mu|$ más un ϵ (para evitar indeterminación ante parches con poco contraste), posteriormente se concatena con el producto del factor de control α que considera la *superposición* entre los parches de alta resolución (naranjas). Dicha concatenación se divide entre el promedio absoluto más ϵ para convertirse en el vector a buscar en el diccionario.

Dada la gran combinación de valores, resulta bastante complicado encontrar exactamente el mismo vector en el diccionario previamente construido y por lo mismo se utilizan algoritmos de búsqueda aproximados como el del *vecino más cercano*. Esto puede observarse en la salida del algoritmo al notar que el vector emparejado tiene un pixel de color diferente al solicitado, pero como es muy cercano, resulta la salida del algoritmo de búsqueda.

Dicho vector de búsqueda permite emparejar al parche de alta resolución asociado en el diccionario, el cual será multiplicado por el factor $|\mu|+\epsilon$ para retomar las tonalidades del parche de baja resolución e incrustado en la *imagen de altas frecuencias* para ejecutar nuevamente el algoritmo de manera iterativa.

De acuerdo con [2], el *algoritmo de un paso* evita considerar los parches como variables aleatorias y relacionarlas con *Redes de Markov* considerando el algoritmo *belief propagation*. Esto es una enorme ventaja, ya que se simplifica el proceso de selección del parche considerando únicamente la *superposición* futura en los parches de alta resolución.

2-B3. Imagen de alta resolución

Finalmente, como resultado del algoritmo propuesto en [2] se obtiene una imagen de mejor resolución que la entrada considerando un factor de escalado deseado para el proceso de interpolación. Note que el motor central del algoritmo está basado en el algoritmo de predicción descrito anteriormente y el diccionario construido a partir de un conjunto de imágenes donde se realizará la búsqueda mediante el algoritmo del *vecino más cercano*.

Como producto del algoritmo de predicción se obtendrá una imagen con sólo frecuencias altas que no se encontraban explícitamente en la imagen de entrada y que se sumará con la imagen escalada para obtener así la imagen de alta resolución respecto la imagen de entrada tal como se observa en la Figura 2.3.

2-C. Redes Convolucionales

Las Redes neuronales convolucionales son un tipo de redes neuronales artificiales donde las *neuronas* corresponden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria (V1) de un cerebro biológico. Este tipo de red es una variación de un perceptrón multicapa, sin embargo, debido a que su aplicación es realizada en matrices bidimensionales, son muy efectivas para tareas de visión artificial, como en la clasificación y segmentación de imágenes, entre otras aplicaciones.

2-D. SRCNN

Como se menciona en [2],

2-E. SRGAN

El SISR (Single Image Super Resolution) es un problema inverso, *it est*, que para una imagen de baja resolución puede haber muchas imágenes diferentes de alta resolución que le correspondan, esto basado en la interpretación del metodo utilizado, ya que el principio basico es añadir información para obtener imágenes de alta resolución. Las CNN presentan un gran avance en la reconstrucción de imágenes de baja resolución a alta resolución, sin embargo, debido al escalado de la imagen o el hecho de que la imagen que se busca mejorar presenta grandes variaciones con respecto a las del dataset (*Data Augmentation*) los resultados podrían ser no satisfactorios.

Una alternativa que propone un nuevo paradigma son las GAN´s (Generative Adversarial Networks) cuyo funcionamiento está basado en la estimación de modelos generadores, como mencionan Goodfellow et al. [4], esto es posible gracias al entrenamiento simultáneo de dos modelos, uno *generador (G)* que obtiene la distribución de la entrada para generar datos falsos y el otro *discriminador (D)* el cual se encarga de estimar la probabilidad de que la muestra provenga del dataset de entrenamiento y discernir así entre estos datos y los del modelo *generador (G).*

El término *antagónicas* como se menciona en [5], se refiere a la dinámica competitiva que se mantiene entre los dos modelos. Por un lado, el generador tiene por objetivo crear nuevos datos que sean indistinguibles del conjunto de entrenamiento, mientras que el discriminador debe poder ser capaz de distinguir cuáles son los datos creados y los reales, siendo los últimos los que corresponden al conjunto de entrenamiento. Esto resulta en un proceso iterativo donde estos dos modelos se desafían uno a otro, logrando un ajuste de parámetros que logran producir datos que se parezcan con gran acierto a los reales.

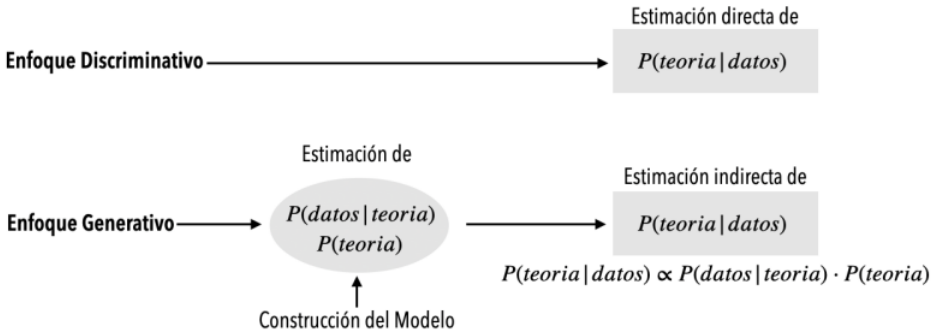

Figura 2.6. Modelo Generador y Discriminador

2-E1. Componentes.

Profundizando un poco más en los componentes del algoritmo, el discriminador es una red neuronal convolucional que consta de muchas capas ocultas y una capa de salida, la principal diferencia aquí es que la capa de salida de las GAN puede tener solo dos salidas, a diferencia de las CNN, que pueden tener un número diferente de salidas con respecto a la cantidad de etiquetas en las que está entrenado. La salida del discriminador puede ser 1 o 0 dependiendo de la función de activación que se aplique. Si la salida es 1, entonces los datos proporcionados son reales y si la salida es 0, entonces se refiere a ellos como datos falsos.

El discriminador está capacitado con los datos del dataset, con estos aprende a reconocer cómo se ven y qué características deben clasificarse como reales, formalmente, discrimina entre $\tilde{x}$, la muestra falsa, y x, los datos muestreados de la distribución real de datos $P_{datos}(x)$.

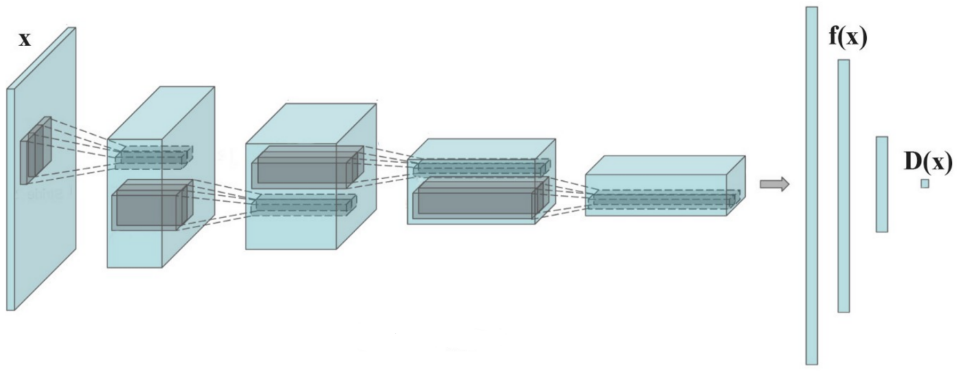

Figura 2.7. Modelo Discriminador

Por el contrario, el generador es una red neuronal convolucional inversa, hace exactamente lo opuesto de lo que hace una CNN, ya que a estas se les da una imagen real como entrada y se espera una etiqueta clasificada como salida, pero en el generador, un vector de ruido aleatorio*(z)* se da como señal de entrada y se espera una imagen falsa como salida, esta imagen debera aproximarse a la real a partir de una distribución pz(z) (en general una distribución Gaussiana) que produce una muestra de datos falsos,$\tilde{x}$ es decir:

$$G(z) = \tilde{x} \tag{2.1}$$

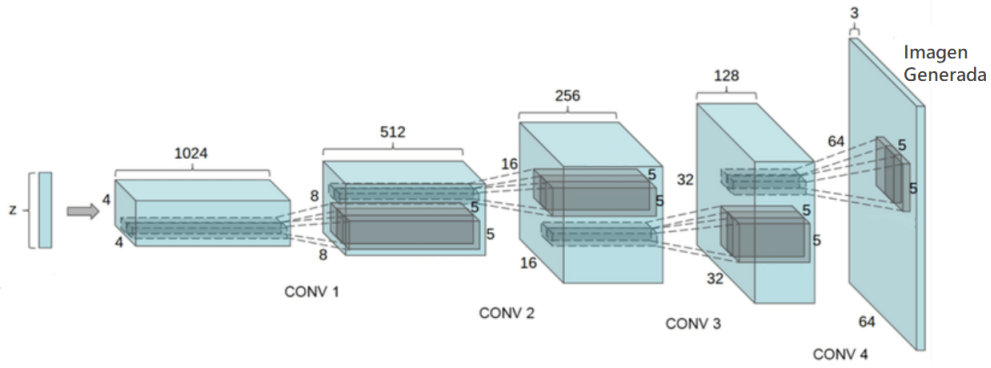

Figura 2.8. Modelo Generador

2-E2. Funciones de perdida.

2-E3. Función de activación.

3. Implementación

Dado que el objetivo del proyecto es comparar tres diferentes métodos de súper resolución, en esta sección se presentarán las actividades requeridas para la implementación de los métodos. Considerando desde los requisitos de software, hardware y datos de entrenamiento en caso sean necesarios.

3-A. Example-Based Super-Resolution

Para el primero de ellos, se retoma la literatura expuesta en [2] comenzando con la generación del diccionario donde se encuentran relacionados los parches de baja y alta resolución. Previo a esto, es necesario pre-procesar el conjunto de imágenes para su posterior segmentación.

De acuerdo a las indicaciones, se debe tener en pares imágenes en alta y baja resolución. En particular, se ha considerado las primeras 13 imágenes del *dataset* disponible en [6].

Realizando un método iterativo que realiza un barrido bidimensional en cada imagen resulta bastante sencillo obtener los parches correspondientes de alta y baja resolución.

3-B. Implementación SRGAN

3-B1. Entrenamiento

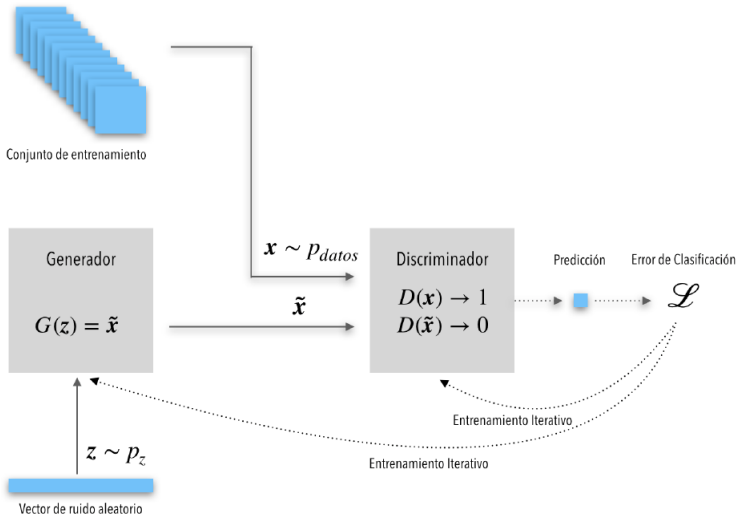

Figura 3.1. Proceso de entrenamiento

4. Resultados

5. Discusión

6. Conclusiones

buenas buenas

Referencias

[1] S. McHugh, “Digital image interpolation,” 2005. [Online]. Available: https://www.cambridgeincolour.com/tutorials/image-interpolation.htm
[2] W. T. Freeman, T. R. Jones, and E. C. Pasztor, “Example-based super-resolution,” *Image-Based Modeling, Rendering, and Lighting*, 2002.
[3] S. Senda, T. Shibata, and A. Iketani, “Example-based super resolution to achieve fine magnification of low-resolution images,” *NEC TECHNICAL JOURNAL*, 2012.
[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *ArXiv*, 06 2014.
[5] L. Calcagni, “Redes generativas antagónicas y sus aplicaciones,” Ph.D. dissertation, Universidad Nacional de la Plata, 04 2020.
[6] M. J. Huiskes and M. S. Lew, “The mir flickr retrieval evaluation,” in *MIR ’08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval*. New York, NY, USA: ACM, 2008.