

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 píxeles 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 calidad.

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 calidad se busca aumentar la densidad de píxeles de la imagen con el objetivo de hacer la imagen más grande y mejorar sus detalles a partir de la predicción de píxeles 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 píxeles.

Esto permite proponer el uso de algoritmos de interpolación que buscan predecir los 3 píxeles vecinos y con ello aumentar la densidad de píxeles 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 píxeles por igual dada la predicción de un pixel central de acuerdo a sus píxeles 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 píxeles 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 píxeles conocidos para la predicción del píxel 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 calidad 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.

NOTA: Agregar comparativo de interpolaciones con un parche y escalado x2

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

NOTA: Imagen entre los tres efectos comentados al realizar 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 píxeles 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 píxeles 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 calidad 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 2.1 puede observarse de manera específica el algoritmo propuesto basado en el parchado de la imagen de entrada mediante un algoritmo de predicción.

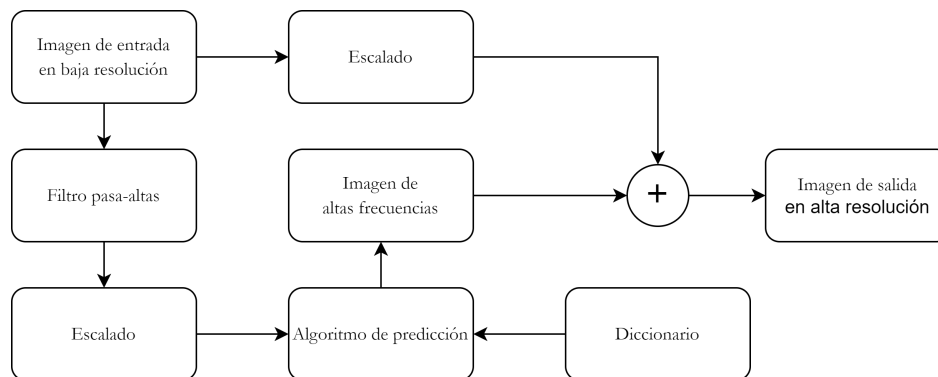


Figura 2.1. Algoritmo de súper resolución

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 calidad 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 píxeles mientras que los de alta calidad serán de 5x5 todos con centro en el mismo píxel para mantener la relación. 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 donde se tendrá toda la información del algoritmo es necesario establecer el algoritmo de predicción para generar esos detalles no visibles en la imagen original.

Para ello, es necesario pre-procesar la imagen de entrada (en baja resolución) mediante un filtrado pasa-altas para eliminar información innecesaria y posteriormente se realiza un proceso de escalado mediante algún algoritmo de interpolación para aumentar sus dimensiones y con ello la densidad de los píxeles tal como se presenta en la Figura 2.1. Observe que para ese punto, la imagen de entrada ya sólo 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.

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],

3. IMPLEMENTACIÓN

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.