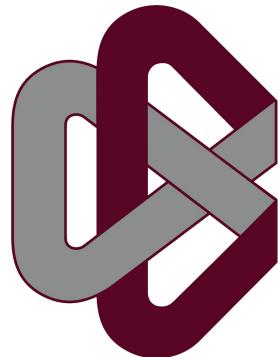


SUPER-RESOLUCIÓN USANDO UNA ÚNICA IMAGEN



CIMAT

CENTRO DE INVESTIGACIÓN EN MATEMÁTICAS

PRESENTA: RUBÉN RODRÍGUEZ RODRÍGUEZ

ASESOR: OSCAR SUSANO DALMAU CEDEÑO

Trabajo de tesis presentado para obtener el grado de

Maestría en Ciencias con Especialidad en Computación y Matemáticas Industriales.

Marzo 2016

HOJA DE FIRMAS DEL JURADO

Dr. Johan Jozef Lode Van Horebeek

Dr. Jean Bernard Hayet

Dr. Oscar Susano Dalmau Cedeño

Este trabajo de tesis está dedicado a todas las personas que me apoyaron de alguna manera para cumplir esta meta, entre ellas están mi novia y futura esposa, mi madre y gran parte de mi familia, quienes con sus acciones y palabras de aliento me apoyaron de manera significativa para concluir mi grado de maestría.

Agradecimientos

Me gustaría agradecer a mi asesor el Dr. Oscar S. Dalmau Cedeño por su asesoramiento, apoyo y guía intelectual durante estos meses de trabajo. También me gustaría agradecer al Centro de Investigación en Matemáticas (CIMAT) y a todos mis profesores durante mi grado de maestría por todas las cosas increíbles, divertidas y útiles que aprendí.

Un agradecimiento especial a mi asesor José Manjón Herrera y a la Universitat Politècnica de València por su apoyo y asesoría durante mi estancia de investigación en Valencia, España.

Finalmente, pero no menos importante quiero agradecer a CONACyT por el apoyo económico que hizo posible que llevara a cabo mis estudios de postgrado y mi estancia académica.

Resumen

En este trabajo de tesis se presentan dos propuestas para resolver el problema de Single Image Super Resolution o Super Resolución Basada en una Única Imagen, enfocadas para imágenes naturales. La solución a este problema involucra eliminación de ruido (no abordado en esta tesis), deblurring (eliminar borrosidad de la imagen) e incremento en la resolución espacial. Los métodos tradicionales de interpolación como lineal, bilineal o bicúbica funcionan bien en zonas sin frecuencias altas (texturas, bordes, etc.) al usarlos para incrementar la resolución espacial de una imagen digital, pero en zonas con gradientes altos estos algoritmos fallan al recuperar poca información de altas frecuencias y al introducir blur en la imagen resultante. El objetivo del trabajo presentado es mejorar la calidad de los métodos tradicionales de interpolación infiriendo la información de altas frecuencias perdida durante el proceso de adquisición de una imagen digital.

En base a investigaciones recientes que demuestran que la relación entre baja y alta resolución a nivel de parches puede representarse bien mediante técnicas de aprendizaje maquina, como maquinas de soporte vectorial, redes neuronales convolucionales, diccionarios y representación “sparse” y además que mediante el uso de técnicas de agrupamiento (como K-Means) se pueden mejorar los resultados de algunos algoritmos de Single Image Super Resolution, se proponen dos modelos para resolver este problema; Un modelo de estimación de parches de alta resolución basado en un conjunto de redes neuronales artificiales y un modelo que hace uso de un conjunto de pares de diccionarios que representan información de baja y alta resolución, tanto las redes neuronales artificiales como los pares de diccionarios son usados para inferir las altas frecuencias de un parche de baja resolución.

El proceso descrito en este trabajo se divide en dos partes. La primera consiste en la estimación de los parámetros de los modelos propuestos y la segunda en utilizar los modelos con sus parámetros bien estimados para reconstruir una imagen de alta resolución dada una imagen de baja resolución. El proceso de estimación de parámetros se define como entrenamiento y el proceso de reconstruir una imagen de alta resolución se define como reconstrucción, ambas etapas son explicadas en este trabajo.

Los resultados conseguidos resultan ser competitivos con técnicas del estado del arte e incluso mejoran los resultados de algunas de ellas en varias de las pruebas realizadas.

Índice general

| | |
|--|----------|
| 1. Introducción | 1 |
| 1.1. Background | 2 |
| 1.2. Objetivos | 4 |
| 1.3. Aplicaciones | 4 |
| 1.4. Estructura de la Tesis | 5 |
| 2. Trabajo Previo | 7 |
| 2.1. Modelo de Adquisición de una Imagen Digital Monocromática | 8 |
| 2.1.1. Clasificación de los Algoritmos de Super Resolution | 10 |
| 2.1.1.1. Algoritmos Basados en Secuencias | 10 |
| 2.1.1.2. Algoritmos Basados en una Única Imagen | 10 |
| 2.2. Explotando la Redundancia de Parches | 12 |
| 2.2.1. Super Resolution Usando la Información de la Imagen de Baja Resolución Únicamente | 12 |
| 2.2.2. Propiedad de Similaridad Local (Local Self-Similarity) | 14 |
| 2.2.3. Conclusión Sobre los Algoritmos que Usan la Redundancia de Parches | 16 |
| 2.3. Algoritmos Basados en Aprendizaje por Computadora | 17 |
| 2.3.1. Algoritmos Basados en Diccionarios | 17 |
| 2.3.1.1. Single Image Super Resolution Usando Representación Sparse | 17 |
| 2.3.1.2. Usando Aprendizaje de Diccionarios en los problemas de Zooming y Deblurring | 20 |
| 2.3.2. Enfoques Previos que Usan Técnicas de Agrupamiento | 21 |
| 2.3.2.1. Diccionarios Semi-Acoplados Usados en la Solución de Single Image Super Resolution y Photo-Sketch Synthesis | 21 |
| 2.3.2.2. Super Resolution Usando Funciones Simples | 23 |
| 2.3.2.3. Uso de PCA para la Estimación de Diccionarios | 24 |
| 2.3.3. Algoritmos Basados en Redes Convolucionales | 25 |

| | |
|---|-----------|
| 2.3.3.1. Single Image Super-Resolution Usando Redes Convolucionales | 25 |
| 2.3.4. Conclusión Sobre los Algoritmos que Usan el Aprendizaje Maquina | 26 |
| 3. Metodología de las Soluciones Propuestas | 27 |
| 3.1. Panorama General | 28 |
| 3.2. Modelos | 33 |
| 3.2.1. Modelo Basado en Redes Neuronales Artificiales | 34 |
| 3.2.2. Modelo Basado en Diccionarios | 36 |
| 3.3. Estimación de los Parámetros de los Modelos | 38 |
| 3.3.1. Entrenamiento de las Redes Neuronales Artificiales | 39 |
| 3.3.1.1. Topología de la Red Neuronal Artificial | 40 |
| 3.3.2. Entrenamiento de los Diccionarios | 41 |
| 3.3.2.1. Entrenamiento del Diccionario de Baja Resolución | 42 |
| 3.3.2.2. Restricciones Sobre la Representación Entre Diccionarios | 44 |
| 3.3.3. Usando Agrupamiento en los Datos de Entrenamiento | 46 |
| 3.4. Reconstrucción de una Imagen de Alta Resolución | 50 |
| 3.4.1. Estimando los Parches de Alta Resolución | 50 |
| 3.4.2. Reconstrucción de la Imagen de Alta Resolución | 52 |
| 3.4.3. Restricción Global de Reconstrucción | 52 |
| 3.4.4. Resumen del Proceso de Reconstrucción | 55 |
| 3.5. Factores de Escala | 56 |
| 4. Experimentos y Resultados | 58 |
| 4.1. Métricas Usadas en los Experimentos | 59 |
| 4.2. Construcción de la Base de Datos de Entrenamiento | 60 |
| 4.3. Tamaño de Parche | 61 |
| 4.4. Número Óptimo de Grupos, K -Means | 67 |
| 4.5. Estimación de los Parámetros para el Modelo Basado en Redes Neuronales Artificiales | 68 |
| 4.6. Estimación de los Parámetros para el Modelo Basado en Diccionarios | 73 |
| 4.6.1. Cantidad de Átomos en el Diccionario | 74 |
| 4.6.2. Estimación del Factor Sparse en el Modelo Basado en Diccionarios - ¿Solución Sparse? | 77 |
| 4.7. Estimación del Parámetro de la Función de Pesos | 80 |
| 4.8. Comparación con Otros Algoritmos | 85 |
| 4.8.1. Pruebas con factor de escala 2x | 86 |
| 4.8.1.1. Base de Datos de Imágenes Naturales | 86 |
| 4.8.1.2. Base de Datos de Imágenes de Texto | 97 |

| | |
|---|------------|
| 4.8.2. Factor de escala 4x | 101 |
| 5. Conclusiones y Trabajo Futuro | 110 |
| 5.1. Conclusiones | 110 |
| 5.2. Trabajo Futuro | 112 |
| Bibliografía | 115 |

Índice de figuras

| | | |
|------|---|----|
| 1.1. | Adquisición de una imagen digital. | 2 |
| 1.2. | Artefactos no deseados dentro del proceso de adquisición de imágenes. | 3 |
| 2.1. | Incremento de la resolución espacial usando técnicas de interpolación. | 8 |
| 2.2. | Esquema de Super Resolution propuesto por Glasner et al. (2009). | 13 |
| 2.3. | Artefactos no deseados introducidos por el método de Glasner et al. (2009). | 14 |
| 2.4. | Propiedad de Local Self-Similarity. | 15 |
| 2.5. | Comparación entre dos enfoques de Single Image Super Resolution. | 15 |
| 2.6. | Fallo de la propiedad de Local Self-Similarity. | 16 |
| 3.1. | Representación de una imagen natural usando pequeños parches. | 29 |
| 3.2. | Diagrama del Proceso de Entrenamiento. | 30 |
| 3.3. | Diagrama de Flujo para el Proceso de Reconstrucción. | 32 |
| 3.4. | Single Image Super Resolution: un problema mal condicionado. | 34 |
| 3.5. | Estructura de una Red Neuronal Artificial. | 41 |
| 3.6. | Diccionarios de Alta Resolución y sus Correspondientes en Baja Resolución. | 46 |
| 3.7. | Incremento de PSNR vs Cantidad de grupos. | 48 |
| 3.8. | 1024 grupos obtenidos de un conjunto de 577700 parches. | 49 |
| 4.1. | Diagrama de creación de la base de entrenamiento. | 60 |
| 4.2. | Incremento del PSNR obtenido con diccionarios respecto al PSNR obtenido con interpolación bicubica. | 69 |
| 4.3. | Incremento del PSNR obtenido con diccionarios respecto al PSNR obtenido con interpolación bicubica. Continuación de la Fig. 4.2 | 70 |
| 4.4. | Incremento del PSNR obtenido con redes neuronales artificiales respecto al PSNR obtenido con interpolación bicubica. | 71 |
| 4.5. | Incremento del PSNR obtenido con redes neuronales artificiales respecto al PSNR obtenido con interpolación bicubica. | 72 |
| 4.6. | PSNR vs Número de Átomos - 64 Grupos. | 75 |
| 4.7. | PSNR vs Número de Átomos - 300 Grupos. | 76 |
| 4.8. | PSNR vs Número de Átomos - 1024 Grupos. | 76 |
| 4.9. | Efecto al incrementar λ | 78 |

| | |
|---|-----|
| 4.10. Resultado para $\lambda = 0.1$ | 79 |
| 4.11. Comportamiento del factor λ | 80 |
| 4.12. Cantidad de evaluaciones de redes neuronales artificiales. | 82 |
| 4.13. PSNR Obtenido variando los valores del parámetro h | 83 |
| 4.14. SSIM Obtenido variando los valores del parámetro h | 84 |
| 4.15. Comparación Entre Métodos del Estado del Arte - Microchip | 87 |
| 4.16. Comparación de Resultados Entre Algoritmos. | 88 |
| 4.17. Comparación de Resultados Entre Algoritmos. | 89 |
| 4.18. Comparación de Resultados Entre Algoritmos. | 90 |
| 4.19. Comparación de Resultados Entre Algoritmos. | 91 |
| 4.20. Comparación de Resultados Entre Algoritmos. | 92 |
| 4.21. Comparación de Resultados Entre Algoritmos. | 93 |
| 4.22. Comparación de imágenes de texto obtenidas por diferentes algoritmos. . . | 98 |
| 4.23. Comparación de imágenes de texto obtenidas por diferentes algoritmos. . . | 98 |
| 4.24. Comparación entre diferentes algoritmos. Imagen de un ojo de un bebe. . . | 102 |
| 4.25. Comparación entre diferentes algoritmos. Imagen de una construcción en- tre árboles. | 104 |
| 4.26. Comparación entre diferentes algoritmos. Imagen de un microchip. | 105 |
| 4.27. Comparación entre diferentes algoritmos. Imagen de la cara de un adoles- cente. | 106 |
| 4.28. Comparación entre diferentes algoritmos. Imagen de una mariposa. | 107 |
| 4.29. Comparación entre diferentes algoritmos. Imagen del ojo de un adulto mayor. | 108 |
| 4.30. Comparación entre diferentes algoritmos. Imagen de un edificio. | 109 |

Abreviaciones y Símbolos

| Símbolo | Descripción |
|---|---|
| I_l | Imagen de baja resolución. |
| I_h | Imagen estimada de alta resolución. |
| Y | Versión interpolada de I_l . Se define como $Y = I_l \uparrow_s$. |
| y | Parche de baja resolución de tamaño $P = (2a + 1)^2$ tomado de Y , $y \in \mathbb{R}^P$. |
| X | Imagen real de alta resolución (Ground Truth). |
| x | Parche de alta resolución de tamaño $P = (2a + 1)^2$ tomado de X , $x \in \mathbb{R}^P$. |
| $x = f(y \Theta)$ | Se usa para denotar una función que transforma un parche de baja resolución a su versión en alta resolución. Θ son los parámetros que recibe la función. |
| H_σ | Kernel gaussiano con un cierto valor σ . |
| B | Kernel de Back-Projection. |
| * | Operador de convolución. |
| \uparrow_s | Operador de upsampling. Incrementa la resolución espacial de una imagen por un factor de escala s . |
| \downarrow_s | Operador de downsampling. Decrementa la resolución espacial de una imagen por un factor de escala s . |
| \hat{x} | Parche estimado de alta resolución de tamaño $P = (2a + 1)^2$. |
| \hat{X} | Matriz formada por parches estimados de alta resolución, $\hat{X} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N)$. |
| J | Tamaño de un conjunto de pares de parches de entrenamiento. |
| K | Cantidad de grupos de pares de parches con estructura similar. |
| N | Cantidad de píxeles en una imagen. |
| M | Número de átomos o elementos de un diccionario. |
| D_k^y | Diccionario entrenado con parches de baja resolución del conjunto k -ésimo. |
| D_k^x | Diccionario entrenado con parches de alta resolución del conjunto k -ésimo. |
| $\text{ANN}_k(\cdot \Theta_k^{\text{ANN}})$ | Red neuronal artificial entrenada con parches de baja resolución como entrada y alta resolución como salida, del conjunto k -ésimo. |

| | |
|----------------------|--|
| G^J | Conjunto de pares de parches de entrenamiento, definido como $G^J = \{(\mathbf{y}_j, \mathbf{x}_j) \mathbf{y}_j \in \mathbb{R}^p, \mathbf{x}_j \in \mathbb{R}^p, j = 1, 2, \dots, J\}$ |
| G_k^Q | k -ésimo conjunto de pares de parches de entrenamiento, definido como $G_k^Q = \{(\mathbf{y}_q, \mathbf{x}_q) \mathbf{y}_q \in \mathbb{R}^P, \mathbf{x}_q \in \mathbb{R}^P, q = 1, 2, \dots, Q_k\}, G_k^Q \subset G^J$. |
| G^N | Conjunto de parches de baja resolución tomados de \mathcal{I}_l , definido como $G^N = \{\mathbf{y}_n \mathbf{y}_n \in \mathbb{R}^P, n = 1, 2, \dots, N\}$. |
| G^{ANN} | Conjunto de redes neuronales artificiales, definido como $G^{\text{ANN}} = \{\text{ANN}_k(\mathbf{y} \Theta_k^{\text{ANN}}) k = 1, 2, \dots, K\}$. |
| G^D | Conjunto de pares de parches de diccionarios , definido como $G^D = \{(\mathbf{D}_k^y, \mathbf{D}_k^x) \mathbf{D}_k^y \in \mathbb{R}^{P \times M}, \mathbf{D}_k^x \in \mathbb{R}^{P \times M}, k = 1, 2, \dots, K\}$. |
| G^μ | Conjunto de centroides, definido como $G^\mu = \{\boldsymbol{\mu}_k k = 1, 2, \dots, K\}$. Estos centroides representan el promedio de cada conjunto G_k^Q . |
| $\bar{\mathbf{Y}}_k$ | Matriz compuesta de parches de baja resolución, $\bar{\mathbf{Y}}_k = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{Q_k})$; |
| $\bar{\mathbf{X}}_k$ | Matriz compuesta de parches de alta resolución, $\bar{\mathbf{X}}_k = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{Q_k})$; |
| $\ \cdot\ _1$ | Representa la norma $_1$. |
| $\ \cdot\ _2$ | Representa la norma $_2$. |
| $\ \cdot\ _F$ | Representa la norma de Frobenius. |
| λ | Factor de regularización. |
| η | Ruido gaussiano con ciertos parámetros. |
| MEAN | Función que calcula la media de una imagen o vector. |
| MSE | Mean Square Error, se define como $(MSE) = \frac{1}{n} \sum_{i=1}^L (\hat{\Phi}_i - \Phi_i)^2$. |
| RMSE | Root Mean Square Error, se define como \sqrt{MSE} . |
| PSNR | Peak Signal to Noise Ratio, se define como $PSNR = 10 \log_{10} \left(\frac{\text{MAX}_I^2}{MSE} \right)$, donde MAX representa el valor máximo de intensidad que puede tomar la imagen. |
| SSIM | Structural Similarity, se define como $SSIM(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$ donde, μ_x es el promedio de \mathbf{x} , μ_y es el promedio de \mathbf{y} , σ_x^2 es la varianza de \mathbf{x} , σ_y^2 es la varianza de \mathbf{y} , $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ son dos variables que estabilizan la división, L es el rango dinámico de los valores de los píxeles, $k_1 = 0.01$ y $k_2 = 0.03$ por defecto. |

Capítulo 1

Introducción

Hoy en día, las imágenes digitales aparecen en muchas partes de nuestras vidas, como en los televisores, en nuestros teléfonos celulares, en los hospitales con maquinas de rayos x o de resonancia magnética, en laboratorios con microscopios o telescopios, etc. Muchos algoritmos son usados para procesar toda esta información y muchos otros se están desarrollando para ser utilizados en un futuro. Esto es un ejemplo de la importancia que tienen las imágenes digitales en nuestra vida diaria. Suponiendo en el contexto de un hospital que una maquina de rayos x no funciona bien y genera imágenes borrosas, el doctor no sería capaz de determinar si el paciente tiene un hueso roto. Otro ejemplo, si un satélite toma una imagen borrosa de una ciudad y esa imagen es usada por un sistema de GPS entonces este sistema podría dar instrucciones incorrectas al conductor. Estos son algunos ejemplos de como se podría afectar la vida de las personas si los algoritmos de procesamiento de imágenes no funcionaran correctamente.

En esta tesis se presenta un trabajo enfocado en resolver un problema de procesamiento de imágenes llamado **Single Image Super Resolution**. Este problema se compone de varios sub-problemas como lo son eliminación de ruido, la reducción de blur, y el incremento de la resolución espacial de imágenes digitales. El objetivo de este algoritmo es mejorar la calidad de las imágenes digitales de manera que pueda usarse como un pre-proceso para algoritmos de procesamiento de imágenes que usan como datos de entrada precisamente, imágenes digitales.

En general, cuando una imagen es adquirida, la escena real se ve afectada por problemas como blur, ruido y perdida de información de altas frecuencias (texturas y detalles finos). Estos problemas se deben al proceso que existe al momento de adquirir la imagen digital. Los niveles de blur y ruido pueden ser diferentes dependiendo del hardware y las condiciones del entorno donde fueron adquiridas las imágenes. La calidad de las imágenes se puede mejorar si se cuenta con hardware más sofisticado y con condiciones de entorno

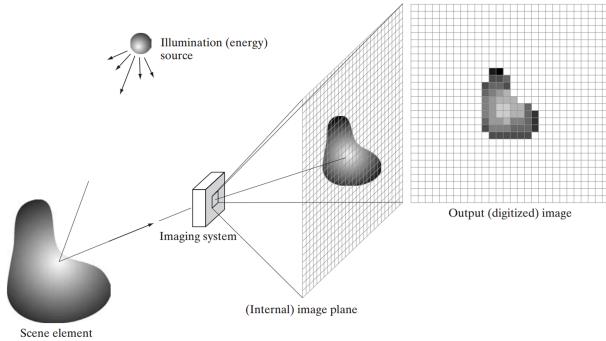


Figura 1.1: Adquisición de una imagen digital.
Ejemplo de la adquisición de una imagen digital tomado de Gonzalez and Woods (2006).

óptimas. Sin embargo, el hecho de adquirir o desarrollar mejor hardware implica una mayor inversión de dinero lo cual puede no ser factible en muchos casos. El software se puede tomar como alternativa al hardware, y con este es posible mejorar imágenes digitales y puede resultar menos costoso.

El objetivo de este trabajo de tesis es proponer y explicar una nueva manera de mejorar las imágenes digitales, resolviendo el problema de Single Image Super Resolution. El resto de este Capítulo se divide en las siguientes Secciones; La Sección llamada **Background** (Sec. 1.1) donde se explica el proceso de adquisición de una imagen digital y los problemas visuales relacionados con este proceso. En la Sección **Objetivos** (Sec. 1.2) se explican cuales son los objetivos de este trabajo de tesis. En la Sección **Aplicaciones** (Sec. 1.3) se mencionan algunas de las posibles aplicaciones en que se podría usar un algoritmo de Single Image Super Resolution. En la Sección **Estructura de la Tesis** (Sec. 1.4) se describen de manera breve los Capítulos que forman este trabajo de tesis.

1.1. Background

Una imagen tomada mediante una cámara digital es generada por la combinación de la luz ambiental (o iluminación de la fuente) y el proceso de reflexión/absorción de la energía sobre los objetos que están dentro de la escena. La energía que es reflejada por los objetos en la escena es medida por uno o más sensores dentro de la cámara. Estos sensores transforman la energía en señales eléctricas que son discretizadas y guardadas en un formato digital (Gonzalez and Woods, 2006). Este proceso de discretización y la óptica causan que la información de frecuencias altas se pierda y que se agregue cierto nivel de blur debido a que varios haces de luz son promediados dentro de un píxel (una unidad de representación

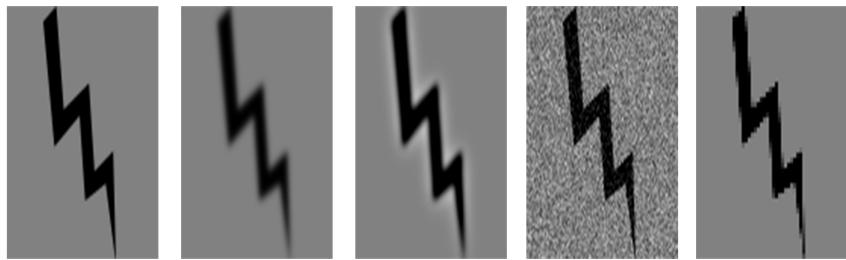


Figura 1.2: Artefactos no deseados dentro del proceso de adquisición de imágenes. De izquierda a derecha 1.- Imagen original, 2.- Bordes con blur, 3.- Edge Halo, 4.- Imagen ruidosa, 5.- Aliasing.

de la intensidad de la luz). Además del blur y de la perdida de frecuencias altas, dentro de las imágenes digitales existen otros tipos de artefactos no deseados. Dentro de los artefactos no deseados en imágenes digitales y que pueden ser generados por el proceso de adquisición de la imagen (mostrado en la Fig. 1.1) o bien por algún algoritmo de procesamiento de imágenes, se encuentran aliasing, blurring, edge halo y ruido. En la Fig. 1.2 se puede observar un ejemplo de estos artefactos.

Como se mencionó anteriormente, construir mejores sensores ayudaría a mejorar la calidad de las imágenes digitales, pero esto resulta en muchos casos costoso y poco práctico. Otra manera de mejorar la calidad es mediante el uso de un software que pueda aplicar un proceso de restauración a la imagen de baja resolución. Un hecho importante de mencionar es que actualmente las cámaras digitales y otros equipos incorporan software interno que intenta mejorar la calidad de las imágenes obtenidas. Además, para reducir el tamaño en término de bytes que ocupan las imágenes capturadas se usan algoritmos de compresión de imágenes como JPEG, PNG, GIF, etc. Este software interno puede generar otro tipo de artefactos no deseados en las imágenes adquiridas, y el trabajo de esta tesis no supone mejorar estos artefactos.

Dentro de las opciones para crear software que pueda ser usado para mejorar la calidad de las imágenes digitales, se encuentran algunas técnicas de restauración de imágenes que usando un modelo matemático (que representa el proceso de adquisición de una imagen) tratan de “de-convolucionar” desde la imagen de baja resolución, para recuperar las altas frecuencias perdidas y así estimar la imagen de alta resolución. Una de las primeras técnicas propuestas fue desarrollada por Irani and Peleg (1991). Esta técnica sugiere usar un conjunto de imágenes digitales tomadas de la misma escena con pequeños desplazamientos

tos entre píxeles. Cada una de estas imágenes agrega restricciones al problema de Super Resolution que hacen posible obtener una buena estimación de la escena real. El problema con esta propuesta es que en gran cantidad de entornos no es posible obtener un conjunto de imágenes de la misma escena. Es por esta razón que algoritmos de Single Image Super Resolution (aquellos que sólo necesitan una única imagen digital) se han presentado en años recientes. Estas técnicas se explican con más detalle en el Capítulo 2.

1.2. Objetivos

El objetivo principal de este trabajo es proponer un algoritmo que pueda resolver el problema de Single Image Super Resolution. Algunas de las dificultades más comunes de propuestas previas a este trabajo son la mala calidad de los resultados al tratar de recuperar frecuencias altas en zonas con textura, artefactos no deseados introducidos por los mismos algoritmos, la lentitud de algunos algoritmos que tardan incluso días para generar un resultado (dependiendo del tamaño de la imagen de baja resolución y de la imagen deseada de alta resolución) o bien los modelos propuestos son complicados y requieren que se definan varios parámetros, etc.

Debido a lo anterior se busca que la propuesta de esta tesis sea simple, es decir, que no requiera demasiados parámetros y que el costo computacional no sea excesivo para poder de esta forma generar resultados en menor tiempo en comparación con otros algoritmos.

Otro objetivo es que la calidad en términos de PSNR y SSIM (medidas de similitud entre imágenes) de las imágenes estimadas con los métodos propuestos en este trabajo sea competitiva con los algoritmos del estado del arte.

1.3. Aplicaciones

Single Image Super Resolution, como se mencionó anteriormente, tiene varias aplicaciones. En este trabajo de tesis, se presentan dos algoritmos que se aplicaron sobre imágenes naturales e imágenes de texto. Otra clase de imágenes digitales en las que se podrían usar este tipo de algoritmos son imágenes médicas como imágenes de resonancia magnética (MRI).

El objetivo de aplicar algoritmos de Single Image Super Resolution en una base de datos compuesta de imágenes de texto supone un paso de pre-procesamiento en algoritmos de, por ejemplo, reconocimiento de caracteres (OCR). Se puede suponer que si las

imágenes que recibe el algoritmo de OCR fueron mejoradas entonces el algoritmo generará mejores resultados, reconociendo una mayor cantidad de caracteres de forma correcta. Una de las mejoras que se puede aportar al usar una base de datos compuesta de imágenes de resonancia magnética es mejorar las imágenes para que los algoritmos de segmentación (comúnmente usados en este tipo de imágenes) den resultados de mejor calidad. Por último, el objetivo de trabajar con una base de datos compuesta de imágenes naturales específicas puede tener varias aplicaciones como lo es segmentación, reconocimiento de objetos, entre otras.

Otra posible aplicación para los algoritmos de Super Resolution es la compresión de vídeo/imagen. De esta manera sería posible reducir el tamaño físico de la imagen (menos columnas y filas) y posteriormente recuperar una imagen similar a la imagen inicial. Esto significaría una perdida de calidad pero podría usarse en aplicaciones donde esta perdida no afecte de forma significativa la experiencia de la visualización del vídeo.

1.4. Estructura de la Tesis

En el Capítulo 2 se discuten las técnicas más comunes de interpolación y los problemas relacionados con usar estas técnicas para incrementar la resolución espacial de una imagen digital. Se introduce el modelo de adquisición de imágenes también llamado modelo de degradación. Se discuten algunos algoritmos del estado del arte, específicamente 2 enfoques, aquellos que explotan cierta información redundante a nivel de parches tomados de diversas imágenes digitales naturales y los que usan aprendizaje maquina para aprender la relación que existe entre los parches de alta resolución y los parches de baja resolución.

En el Capítulo 3 se detallan las dos propuestas de este trabajo de tesis para resolver el problema de Single Image Super Resolution. En la primera propuesta se usa un conjunto de redes neuronales artificiales que tratan de aprender la relación que existe entre las imágenes de alta resolución y las de baja resolución. El segundo enfoque propuesto se basa en el uso de un conjunto de pares de diccionarios entrenados de tal manera que también logren representar la transformación que existe entre una imagen de alta resolución y una de baja resolución. En el mismo capítulo se detalla el proceso de entrenamiento y de reconstrucción de una imagen de alta resolución. También se argumenta la idea del uso de técnicas de agrupamiento como parte de los modelos de Single Image Super Resolution.

En el Capítulo 4 se presentan los resultados de este trabajo y algunos experimentos. Basados en estos experimentos se explica el uso de ciertos parámetros usados en los algo-

ritmos propuestos. También en este Capítulo, se detalla como se construye la base de datos usada en la parte de entrenamiento de las redes neuronales artificiales y diccionarios. Finalmente se comparan los resultados de otros algoritmos del estado del arte con las propuestas presentadas en esta tesis.

Capítulo 2

Trabajo Previo

Las técnicas más comunes para incrementar la resolución espacial mediante el uso de software son los métodos de interpolación. Dentro de los métodos más conocidos y usados se encuentran Nearest Neighbor, Linear, Bilinear, Bicúbica, B-Spline y Lanczos. Estos métodos de interpolación trabajan bien en regiones planas donde hay poca información de altas frecuencias, pero funcionan mal en regiones donde sí hay información de altas frecuencias. En base a esto, se puede hacer la siguiente observación, los métodos de interpolación pueden ser usados para incrementar la resolución espacial de alguna imagen, pero la imagen resultante habrá perdido detalles de alta frecuencia como lo son, bordes, texturas, etc. y aunado a esto, según sea el método podrá añadir artefactos no deseados como aliasing, edge halo y blur.

Como se puede observar en la Fig.2.1 las técnicas de interpolación generan bordes borrosos y no logran recuperar detalles de altas frecuencias. La mejor aproximación es realizada por la interpolación bicúbica (imagen D). En la imagen B podemos observar bordes sin blur pero con problemas de aliasing.

Los métodos de interpolación han sido ampliamente usados, debido a su fácil implementación y a su eficiencia computacional. Sin embargo, debido al avance tecnológico, imágenes digitales de mejor calidad son cada vez más requeridas, calidad que los métodos tradicionales de interpolación no pueden lograr por si mismos.

En el resto del Capítulo se introduce el modelo de adquisición de una imagen monocromática (Sec. 2.1) y se explican brevemente algunos de los algoritmos de Single Image Super Resolution más populares dentro del área de procesamiento de imágenes. La estructura del Capítulo es la siguiente; En la Sección **Clasificación de los Algoritmos de Super Resolution** (Sec. 2.1.1) se clasifican de manera general los enfoques de Super Resolution que se han desarrollado a lo largo del tiempo. Dentro de estos se encuentran dos grupos, los algoritmos basados en secuencias de imágenes (Sec. 2.1.1.1) y los algoritmos basados en

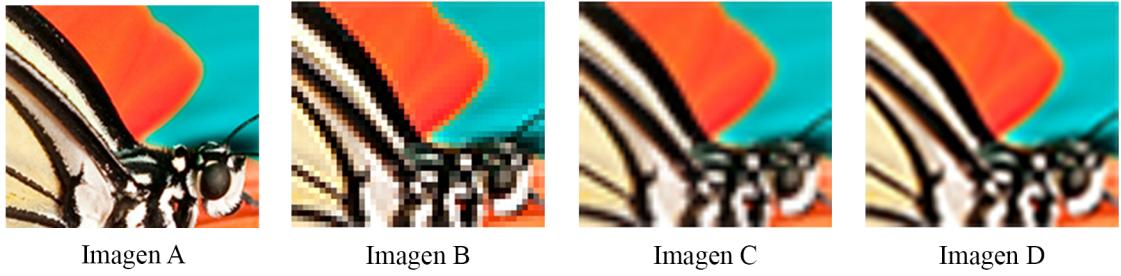


Figura 2.1: Incremento de la resolución espacial usando técnicas de interpolación. En esta imagen se muestran 3 de los métodos de interpolación más populares. A)Imagen Original, B)Nearest Neighbor, C)Bilinear, D)Bicúbica. La imagen A se redujo de tamaño usando el modelo de adquisición con un factor de escala $s = 2$ y un valor de $\sigma = 0.8$. Posteriormente se usaron los métodos de interpolación mencionados para restaurar las dimensiones a la imagen original.

una única imagen (Sec. 2.1.1.2). En la sección **Explotando la Redundancia de Parches** (Sec. 2.2) se explican dos de las propuestas más conocidas de Single Image Super Resolution, presentadas por Glasner et al. (2009) y Freedman and Fattal (2010). En la Sección **Algoritmos Basados en Aprendizaje por Computadora** (Sec. 2.3) se presentan algunas de las propuestas basadas en aprendizaje de diccionarios y representación sparse. Estos algoritmos son de especial interés debido a los buenos resultados que logran generar y además, una de las propuestas de esta tesis usa un enfoque basado en diccionarios. Dentro de esta misma Sección (Sec. ??) se mencionan de manera breve algunas propuestas basadas en otras técnicas de aprendizaje maquina, esto con el fin de dar un panorama general de los trabajos que se han realizado en esta área. Finalmente en la Sección **Conclusión Sobre los Algoritmos que Usan el Aprendizaje Maquina** (Sec. 2.3.4) se presenta una opinión personal sobre los algoritmos previamente mencionados.

2.1. Modelo de Adquisición de una Imagen Digital Monocromática

En el trabajo de Irani and Peleg (1991) se presenta un modelo matemático que es usado por la mayoría de los algoritmos de Super Resolution y Single Image Super Resolution. Este modelo trata de representar el proceso de adquisición de una imagen digital (explicado en el Capítulo 1):

$$\mathbf{I}_j = \mathbf{D}\mathbf{H}_j\mathbf{F}_j\mathbf{X} + \boldsymbol{\eta}_j, \quad (2.1)$$

donde,

\mathbf{I}_j es la j -esima imagen de baja resolución en la secuencia.

\mathbf{D} es un operador de downsampling.

\mathbf{H}_j es un operador de blurring.

\mathbf{F}_j es un operador de warpping, que contiene información sobre las transformaciones (pequeños desplazamientos y rotaciones) entre imágenes de la secuencia.

\mathbf{X} es la imagen ideal.

$\boldsymbol{\eta}_j$ es un factor de ruido Gaussiano.

La Eq. (2.1) representa el modelo de adquisición para una secuencia de imágenes tomadas de la misma escena y en este se puede observar el origen de algunos de los artefactos no deseados en las imágenes digitales.

El modelo de la Eq. (2.1) está definido para una secuencia de imágenes. En caso de tener una única imagen (como lo es Single Image Super Resolution) el modelo se define como:

$$\mathbf{I} = \mathbf{D}\mathbf{H}\mathbf{X} + \boldsymbol{\eta}, \quad (2.2)$$

donde, se eliminó el operador de warpping debido a que no tiene sentido dejarlo si la secuencia está formada sólo por un elemento, ya que este factor representa información entre dos imágenes de la secuencia y en este caso sólo existe una única imagen.

Por notación se define \mathbf{I}_l como una imagen de baja resolución, \downarrow_s como el operador de downsampling (en lugar de \mathbf{D}), s es un factor de escala, $*$ es un operador de convolución y \mathbf{H}_σ es un kernel Gaussiano con valor σ . Podemos reescribir la Eq.(2.2) como:

$$\mathbf{I}_l = (\mathbf{X} * \mathbf{H}_\sigma) \downarrow_s + \boldsymbol{\eta}, \quad (2.3)$$

donde, \mathbf{I}_l es una matriz que representa una imagen de baja resolución, \mathbf{X} es una matriz que representa una imagen de alta resolución, $*$ es el operador de convolución, \mathbf{H} es un kernel gausiano con cierto valor σ , \downarrow_s es un operador de downsampling con valor de escala s y $\boldsymbol{\eta}$ es una matriz con ruido Gaussiano generado con cierta media y varianza. Este modelo es usado por los métodos propuestos en este trabajo de tesis y es continuamente

referenciado a lo largo de este documento.

2.1.1. Clasificación de los Algoritmos de Super Resolution

Varios algoritmos de Super Resolution propuestos en años recientes pueden mejorar la calidad de las imágenes de baja resolución al ser ampliadas (en resolución espacial) en comparación con los métodos tradicionales de interpolación. Estos algoritmos se pueden clasificar en dos grandes grupos; El primer grupo es aquel que usa una secuencia de imágenes de baja resolución como base para estimar una imagen de alta resolución y el segundo grupo es aquel que únicamente se basa en una imagen de baja resolución como entrada para generar una imagen de alta resolución. El primer grupo se denomina **Basado en Secuencias** y el segundo grupo **Basado en una Única Imagen** (Single Image Super Resolution).

2.1.1.1. Algoritmos Basados en Secuencias

Algunas técnicas basadas en secuencias de imágenes fueron propuestas en los trabajos de Irani and Peleg (1993) y Farsiu et al. (2004). Estas técnicas usan como información de entrada una secuencia de imágenes de baja resolución tomadas de la misma escena pero con pequeños desalineamientos en los píxeles. Cada imagen de baja resolución provee un conjunto de restricciones lineales al problema de Super Resolution. Si suficientes imágenes de baja resolución son dadas entonces se formará un sistema de ecuaciones lineales sobre-determinado que podrá ser evaluado para obtener la imagen de alta resolución. El problema con este tipo de algoritmos es precisamente el requerimiento de un conjunto de imágenes de baja resolución de una misma escena, ya que esto no es siempre posible de obtener. Por ejemplo, en un ambiente clínico significaría someter a un paciente a un proceso que podría ser largo y requeriría que el paciente estuviera estático por mucho tiempo, lo cual no sería adecuado en muchos casos. Sin embargo, este tipo de técnicas pueden ser útiles para obtener fotografías de alta resolución tomadas de un vídeo, dado que un vídeo es por si mismo una secuencia de imágenes.

2.1.1.2. Algoritmos Basados en una Única Imagen

Por las razones anteriores, los métodos de Single Image Super Resolution se han vuelto populares en años recientes y han comenzado a adquirir importancia en diferentes aplicaciones. Algunos de los métodos más conocidos y que son tomados frecuentemente como referencia en nuevas investigaciones, son los trabajos presentados por Glasner et al. (2009), Yang et al. (2010), Freedman and Fattal (2010), Dong et al. (2011) y Wang et al. (2012).

En los trabajos de Freedman and Fattal (2010) y Glasner et al. (2009) los autores explotan la redundancia a nivel de parches que se encuentra en imágenes naturales. En los trabajos de Yang et al. (2010), Wang et al. (2012), Dong et al. (2011) y Dong et al. (2014) se usan técnicas de aprendizaje máquina para aprender la relación entre las imágenes de baja resolución y las de alta resolución.

Generalizando, los algoritmos de Single Image Super Resolution se pueden clasificar en 2 categorías; Los **modelos basados en predicción** (prediction models) y los **modelos basados en ejemplos** (exampled based).

1.- **Modelos basados en predicción** (Prediction Models). Estos algoritmos usan una expresión matemática explícita para estimar la imagen de alta resolución deseada. Por ejemplo, las técnicas de interpolación usan expresiones explícitas para estimar una nueva imagen. Como ya se mencionó anteriormente, estos enfoques trabajan bien en regiones planas de la imagen pero tienen grandes problemas al tratar de estimar regiones con frecuencias altas y gradientes significativos a lo largo de los bordes.

2.- **Modelos basados en ejemplos** (Exampled Based). La principal idea de este tipo de algoritmos es que dada una base de datos de pares de parches $G^J = \{\mathbf{y}_j, \mathbf{x}_j | j = 1, 2, \dots, J\}$ (de baja y alta resolución respectivamente) se debe aprender una función $\mathbf{x}_j = f(\mathbf{y}_j | \Theta)$ que transforme un parche de baja resolución a su versión en alta resolución, con Θ un conjunto de parámetros de la función. Varios métodos han sido usados con el fin de aprender la función $f(\mathbf{y}_n | \Theta)$. Algunos de estos métodos que son comúnmente usados en la literatura son Maquina de Soporte Vectorial (An and Bhanu, 2011), Redes Neuronales Convolucionales (Dong et al., 2015), Representación Sparse y Aprendizaje de Diccionarios, etc. Estos algoritmos pueden usarse tanto en el enfoque para secuencias de imágenes de baja resolución como en el enfoque para una única imagen de baja resolución.

El trabajo presentado en esta tesis puede ubicarse en la categoría de modelos basados en ejemplos. La propuesta presentada en este trabajo trata de aprender la relación entre parches de baja y alta resolución. Algunos de los trabajos que funcionan con un enfoque similar son los presentados por Yang et al. (2010), Wang et al. (2012) y Dong et al. (2011). En las siguientes secciones de este Capítulo se explican brevemente algunos trabajos previos desarrollados para resolver el problema de Single Image Super Resolution.

2.2. Explotando la Redundancia de Parches

2.2.1. Super Resolution Usando la Información de la Imagen de Baja Resolución Únicamente

Glasner et al. (2009) proponen un algoritmo híbrido que resulta de la combinación de enfoques clásicos de técnicas de Super Resolution basadas en secuencias y las técnicas basadas en ejemplos. A continuación se explican ambos enfoques.

En los métodos clásicos de Super Resolution, usando un conjunto de imágenes de baja resolución $\{\mathbf{L}_1, \dots, \mathbf{L}_J\}$ de la misma escena (con pequeñas transformaciones a nivel de sub-píxel entre sí), se busca recuperar la imagen de alta resolución \mathbf{X} de la escena original. Se asume que cada imagen de baja resolución $\{\mathbf{L}_j | j = 1, 2, \dots, J\}$ fue generada a partir \mathbf{X} usando el modelo de degradación Eq. (2.1) para un conjunto de imágenes de baja resolución. Entonces, cada píxel p de cada imagen de baja resolución \mathbf{L}_j introduce una restricción lineal sobre los valores desconocidos de las intensidades de los píxeles en el vecindario local alrededor del píxel correspondiente de alta resolución. El tamaño del vecindario esta determinado por el tamaño del kernel Gaussiano H_j (o soporte del kernel) del modelo de degradación:

$$\mathbf{L}_j(p) = (\mathbf{X} * H_j)(q) = \sum_{q_i \in \text{support}(H_j)} X(q_i) H_j(q_i - q) \quad (2.4)$$

donde, $\{X(q_i)\}$ es el píxel de alta resolución que se quiere estimar. Si se cuenta con suficientes imágenes, el número de ecuaciones independientes puede exceder el número de incógnitas. Entonces, se obtiene un sistema de ecuaciones sobre-determinado y la solución al problema es razonablemente estable.

El esquema anterior es una breve descripción del enfoque clásico para algoritmos de Super Resolution basados en un conjunto de imágenes de baja resolución. En el caso de una sola imagen, no se cuenta con suficiente información para obtener las restricciones lineales necesarias para este enfoque. Los autores de este trabajo, proponen buscar para cada parche en la imagen de baja resolución \mathbf{L} , un conjunto de parches similares y usar este conjunto de parches similares para introducir el conjunto de restricciones lineales necesarias para el enfoque anterior. De esta manera se esta aplicando el enfoque clásico utilizando una única imagen de baja resolución, pero como se mencionó anteriormente esta propuesta también hace uso de un enfoque basado en ejemplos, a continuación se explica brevemente esta parte.

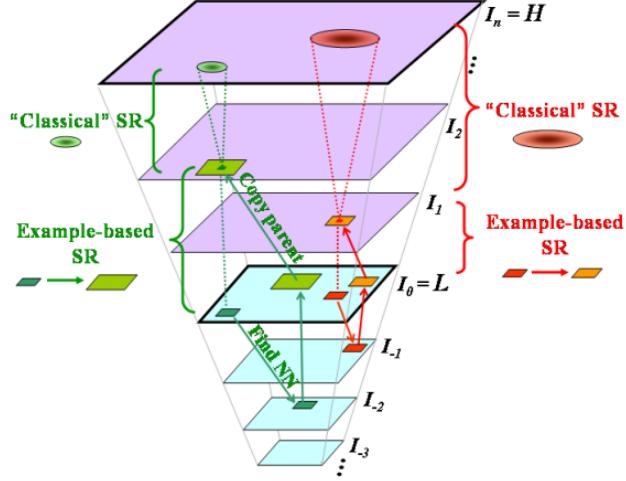


Figura 2.2: Esquema de Super Resolution propuesto por Glasner et al. (2009). Imagen tomada del trabajo de Glasner et al. (2009) donde se describen las dos etapas del algoritmo de Single Image Super Resolution.

Para el enfoque basado en ejemplos se realiza lo siguiente. Se crea una pirámide de escalas donde la imagen de baja resolución L es convolucionada con un kernel Gaussiano H_j y reducida de tamaño por un operador de downsampling \downarrow_s de manera que $\mathbf{L}_0 = \mathbf{L}, \mathbf{L}_1 = (\mathbf{L}_0 * H_1) \downarrow_s, \dots, \mathbf{L}_j = (\mathbf{L}_{j-1} * H_J) \downarrow_s$. Para cada parche de baja resolución en \mathbf{L}_0 se busca el vecino más cercano en todas las imágenes de la pirámide. Una vez encontrado el parche más cercano se toma este parche como un parche de alta resolución del cual se conoce la versión de alta resolución tomado de L_0 . Este parche de alta resolución es pegado directamente en la imagen de alta resolución que se está estimando.

En la Fig. 2.2 se pueden observar las dos etapas del algoritmo de Single Image Super Resolution propuesto por Glasner et al. (2009). Este enfoque puede recuperar información de alta frecuencia, sin embargo, introduce bordes sumamente marcados, artefactos no deseados y no es capaz de realizar una buena estimación en zonas con texturas de alta frecuencia.

El artefacto no deseado generado en la Fig. 2.3 se puede observar frecuentemente en varios algoritmos de Super Resolution. La interpolación Bicúbica es usada comúnmente como aproximación inicial de la solución al problema de Super Resolution por muchos algoritmos. Este artefacto es introducido por esta interpolación y es magnificado por los algoritmos de Super Resolution.

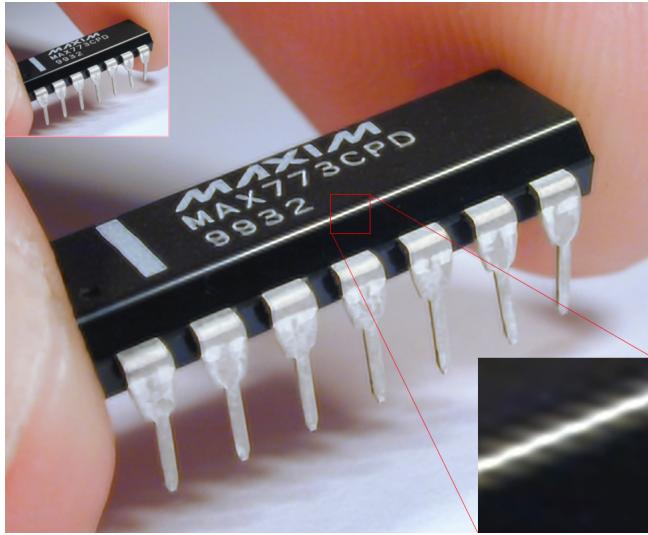


Figura 2.3: Artefactos no deseados introducidos por el método de Glasner et al. (2009). En esta imagen se puede observar un ejemplo de los resultados presentados en el trabajo de Glasner et al. (2009). Se puede notar que se generan líneas con cierta inclinación al rededor de la línea blanca a lo largo del microchip, también se puede notar que los bordes están resaltados de manera significativa.

2.2.2. Propiedad de Similaridad Local (Local Self-Similarity)

En el trabajo de Freedman and Fattal (2010) se propone un algoritmo que resuelve el problema de *image and video upscaling*, que es otro nombre que se da al problema de Super Resolution. Los autores de este trabajo también usan la redundancia a nivel de parches que se encuentra en imágenes naturales para resolver el problema de Single Image Super Resolution. Ellos asumen que para parches suficientemente pequeños existen otros parches muy similares a ellos mismos en imágenes de tamaño mayor usando pequeños factores de escala. Sugieren que esta relación a través de escalas pequeñas se mantiene para varias estructuras de imagen como lo son las líneas rectas, esquinas, etc. La Fig.2.4 muestra un ejemplo visual de esta relación.

Se argumenta en este trabajo que la propiedad de “local self-similarity” (ver Fig. 2.4) es valida solamente para imágenes vecinas a una escala relativamente pequeña. Con el objetivo de preservar esta propiedad al hacer un upsampling de la imagen de baja resolución, se presentan dos operadores de blurring e interpolación. Estos operadores están basados en un banco de filtros que también se proponen en este trabajo. La idea básica usando estos operadores es la siguiente; Dada una imagen de baja resolución \mathbf{I}_l se estima una nueva imagen usando el operador de interpolación propuesto $\hat{\mathbf{I}}_l = \mathbf{I}_l \uparrow_s$ con s pequeño. Este proceso de interpolación provoca que información de altas frecuencias contenida en la imagen de baja resolución se pierda, entonces haciendo uso de la propiedad de ”local self-similarity.”^{es} po-

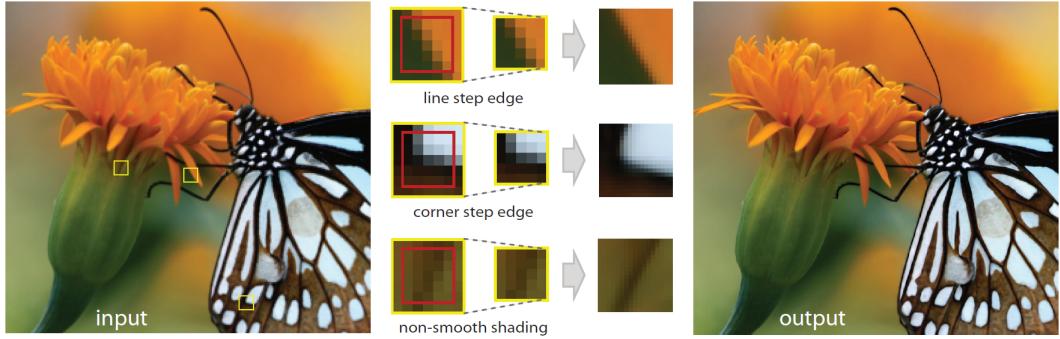


Figura 2.4: Propiedad de Local Self-Similarity.

Los parches amarillos cuando se reducen de tamaño (por un factor de escala pequeño) son muy similares a su versión recortada marcada por el rectángulo rojo. Esta figura fue tomada del trabajo de Freedman and Fattal (2010).

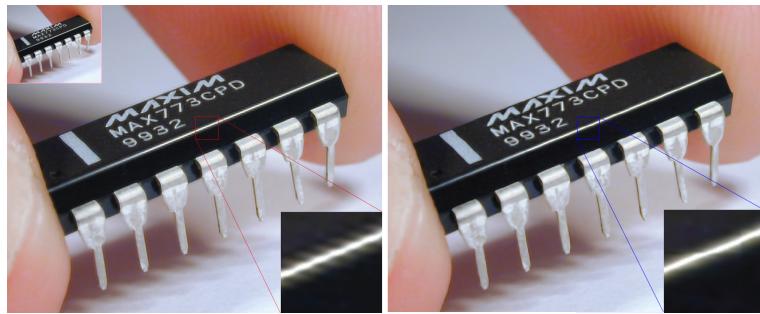


Figura 2.5: Comparación entre dos enfoques de Single Image Super Resolution.

De izquierda a derecha. 1) Resultado de usar el algoritmo propuesto por Glasner et al. (2009). 2) Resultado de usar el algoritmo propuesto por Freedman and Fattal (2010). Puede observarse que el segundo resultado no presenta artefactos a lo largo de la línea blanca en el microchip mientras que el primer resultado sí presenta este problema.

sible recuperar parte de esta información en la imagen de mayor resolución espacial. Para mayores detalles revisar la sección de referencias.

Una ventaja de esta propuesta es que no es necesaria una base de datos externa. Sin embargo, existen grandes desventajas de este método. La desventaja principal es que este algoritmo funciona bien únicamente en zonas con poca textura o bien con texturas planas. Por ejemplo, podría ser un buen algoritmo para usarse en imágenes de texto que contienen poca información de textura. En la Fig. 2.5 se muestra un ejemplo de una imagen obtenida mediante este algoritmo y el propuesto por Glasner et al. (2009).

Como se mencionó, la desventaja de este algoritmo es la calidad de los resultados cuando se usa en imágenes que tienen texturas con altas frecuencias. Este enfoque está basado en la propiedad de “local self-similarity” la cual sólo es razonable para estructuras como



Figura 2.6: Fallo de la propiedad de Local Self-Similarity.

De izquierda a derecha, el resultado obtenido por interpolación bicúbica y el resultado obtenido por el algoritmo propuesto por Freedman and Fattal (2010).

líneas rectas o esquinas, pero el conjunto de estructuras que se pueden encontrar en los parches obtenidos de imágenes naturales está compuesto por más elementos, de tal manera que esta propiedad no se aplica para gran parte de estos elementos. En la Fig. 2.6 se puede observar el problema de este método en una imagen con texturas difíciles de estimar.

2.2.3. Conclusión Sobre los Algoritmos que Usan la Redundancia de Parches

Como conclusión se puede decir que es posible recuperar información de altas frecuencias usando la redundancia de parches a través de imágenes naturales. De esta manera se mejora la calidad de los métodos tradicionales de interpolación. El problema más grande de la mayor parte de estos enfoques es la mala calidad de los resultados en zonas con texturas que contienen gran cantidad de detalles, por lo que estos algoritmos no pueden usarse en cualquier imagen natural. En la siguiente sección se explica otro enfoque para resolver el problema de Single Image Super Resolution. Los enfoques que se presentan en su mayoría superan los resultados obtenidos por los trabajos presentados en esta sección.

2.3. Algoritmos Basados en Aprendizaje por Computadora

2.3.1. Algoritmos Basados en Diccionarios

Varios de los enfoques recientes en algoritmos de restauración de imagen (incluyendo Super Resolution) están basados en una descomposición sparse de parches. Dada una matriz (diccionario) $\mathbf{D} = (\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_M)$ en $\mathbb{R}^{P \times M}$, una señal \mathbf{x} en \mathbb{R}^P es reconstruida como una combinación lineal de unas pocas columnas de \mathbf{D} . Estas columnas son llamadas átomos o elementos del diccionario. Si existe una estimación tal que $\hat{\mathbf{x}} = \mathbf{D}\boldsymbol{\alpha} = \mathbf{x}$ con $\boldsymbol{\alpha}$ como un vector sparse en \mathbb{R}^M , se dice que \mathbf{D} es bien adaptado (aprendido). En procesamiento de imágenes un parche en forma de vector es procesado como una señal \mathbf{x} .

Para conseguir una buena estimación de un parche \mathbf{x} es necesario tener un diccionario bien aprendido/estimado \mathbf{D} . En la literatura se han propuesto varias opciones para definir \mathbf{D} . En el trabajo de Dong et al. (2011) se propone usar el resultado de PCA sobre un conjunto de parches como elementos del diccionario, pero otras técnicas como Wavelets, Transformada de Fourier, parches escogidos aleatoriamente, etc., son comúnmente usadas. Otra forma de aprender el diccionario \mathbf{D} es mediante un proceso de optimización sobre un conjunto de parches. Algunas técnicas para resolver este proceso son Lasso (Tibshirani, 1994) o Basis Pursuit (Chen et al., 2001). El problema de optimización a resolver es el definido en la Eq. 2.5.

$$\min_{\boldsymbol{\alpha}} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1, \quad (2.5)$$

donde, $\boldsymbol{\alpha}$ es un vector de números reales comúnmente llamado “código” y su objetivo es representar los coeficientes que permitan recuperar la señal \mathbf{x} sobre un diccionario \mathbf{D} , λ es un factor de regularización que controla que tan sparse es la solución.

2.3.1.1. Single Image Super Resolution Usando Representación Sparse

En el trabajo de Yang et al. (2010) los autores proponen aprender un par de diccionarios, \mathbf{D}^x para representar información de alta resolución y \mathbf{D}^y para representar parches de baja resolución. Los diccionarios deben tener la misma cantidad de columnas. Aprendiendo la representación sparse de los parches de baja resolución en términos de \mathbf{D}^y el parche de alta resolución correspondiente a cada parche de baja resolución puede ser recuperado

usando el diccionario de alta resolución \mathbf{D}^x . Cada uno de los diccionarios esta compuesto por parches de baja resolución en el caso de \mathbf{D}^y y por parches de alta resolución en el caso de \mathbf{D}^x , cada diccionario tiene alrededor de 100000 parches.

Para cada parche \mathbf{y} extraído de una imagen de baja resolución, se busca encontrar una representación sparse respecto al diccionario \mathbf{D}^y . Los parches de alta resolución correspondientes se obtendrán usando estas representaciones sparse respecto al diccionario \mathbf{D}^x . El problema que se propone resolver para encontrar esta representación sparse es el siguiente:

$$\min \frac{1}{2} \|\mathbf{F}\mathbf{D}^y\boldsymbol{\alpha} - \mathbf{F}\mathbf{y}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1, \quad (2.6)$$

donde, \mathbf{F} es un operador de extracción de características. El objetivo principal de \mathbf{F} es proporcionar una restricción perceptualmente significativa de como los coeficientes $\boldsymbol{\alpha}$ deben aproximar al parche de baja resolución \mathbf{y} . Los autores de este trabajo usan 4 filtros en 1 dimensión (derivadas de primer y segundo orden) para extraer las características de altas frecuencias, los filtros usados son:

$$\begin{aligned} \mathbf{f}_1 &= [-1, 0, 1], & \mathbf{f}_2 &= \mathbf{f}_1, \\ \mathbf{f}_3 &= [1, 0, -2, 0, 1], & \mathbf{f}_4 &= \mathbf{f}_3, \end{aligned} \quad (2.7)$$

donde, T es la versión transpuesta del filtro correspondiente. Aplicando estos 4 filtros a cada parche, se consiguen 4 vectores de características para cada parche, los cuales son concatenados en un sólo vector y es este vector el que es usado como versión final del parche de baja resolución.

Resolver la Eq. (2.6) individualmente no garantiza compatibilidad entre parches adyacentes. Para reforzar la compatibilidad entre parches adyacentes los autores usan un algoritmo de “un-paso”. Los parches de una imagen de baja resolución son procesados en orden, comenzando de izquierda a derecha y de arriba hacia abajo (el problema de optimización Eq. (2.8) se resuelve para cada parche de baja resolución). Modificando la Eq. (2.6) de manera que la reconstrucción de alta resolución $\mathbf{D}^x\boldsymbol{\alpha}$ de un parche \mathbf{y} esté restringida a ser muy cercana con los parches de alta resolución (parches en el vecindario de \mathbf{y}) previamente reconstruidos. El problema de optimización resultante es:

$$\min \|\boldsymbol{\alpha}\|_1 \quad \text{s.t.} \|\mathbf{F} \mathbf{D}^y \boldsymbol{\alpha} - \mathbf{F} \mathbf{y}\|_2^2 \leq \epsilon_1, \|\mathbf{P} \mathbf{D}^x \boldsymbol{\alpha} - \mathbf{w}\|_2^2 \leq \epsilon_2, \quad (2.8)$$

donde, $\boldsymbol{\alpha}$ representa un vector de coeficientes o código sparse, \mathbf{F} es una matriz de extracción de características (definida anteriormente), \mathbf{P} es una matriz que extrae la región de solapamiento entre en el parche actual y la imagen reconstruida previamente (parches vecinos de alta resolución reconstruidos previamente), \mathbf{w} contiene los valores de la región de solapamiento entre el parche actual y la imagen previamente reconstruida, ϵ_1 y ϵ_2 son valores que definen que tanto error es aceptable en la solución.

Una vez que se obtiene el vector de coeficientes óptimo $\boldsymbol{\alpha}^*$ (resolviendo Eq. (2.8)) la estimación de un nuevo parche de baja resolución se obtiene evaluando la siguiente ecuación:

$$\mathbf{x} = \mathbf{D}^x \boldsymbol{\alpha}^*, \quad (2.9)$$

El trabajo de Yang et al. (2010) aborda el mal condicionamiento del problema de Single Image Super Resolution agregando una restricción global de reconstrucción aplicando el algoritmo presentado por Irani and Peleg (1991). Esta restricción se aborta en el Capítulo 3 de este trabajo de tesis. Es importante resaltar que los algoritmos presentados por Glasner et al. (2009) y Freedman and Fattal (2010) no aseguran de ninguna manera que exista relación entre la imagen generada por estos algoritmos y la imagen de baja resolución.

Una ventaja de esta propuesta es que este algoritmo sólo necesita 2 diccionarios pequeños para procesar los parches de alta resolución, esto hace más simple el proceso de entrenamiento. Otra ventaja es que el uso de representación sparse sobre un diccionario sobre-completo (que contiene muchas más columnas que las necesarias para construir una base) ha demostrado que puede representar bien la relación entre parches de baja resolución y los de alta. Sin embargo, algunas de las observaciones que pueden hacerse sobre esta propuesta es que no se menciona como abordar imágenes con ruido, ya que para poder aplicar la restricción global de reconstrucción es necesario que las imágenes de baja resolución no contengan ruido.

2.3.1.2. Usando Aprendizaje de Diccionarios en los problemas de Zooming y Deblurring

Los autores de este trabajo, Couzinie-Devy et al. (2011), usan de manera similar al trabajo de Yang et al. (2010) un par de diccionarios para resolver los problemas de deblurring y zoom de una imagen digital. Como en los métodos basados en ejemplos, se usa un conjunto de pares de parches (obtenidos de pares de imágenes correspondientes con y sin blur) para estimar los parámetros del modelo propuesto en este trabajo. El método propuesto trata de predecir parches sin blur a partir de parches con blur.

Para predecir un parche sin blur, es necesario tomar en cuenta los valores en el vecindario del píxel correspondiente en la imagen con blur. Los parches sin blur y los parches con blur pueden ser de tamaños diferentes, m_s y m_b denotan los tamaños de los parches sin y con blur respectivamente, con $m_s < m_b$.

Se propone aprender un par de diccionarios $\mathbf{D}^s \in \mathbb{R}^{m_s \times k}$ y $\mathbf{D}^b \in \mathbb{R}^{m_b \times k}$ y usarlos en un modelo no lineal (propuesto en el trabajo) para reconstruir patrones que ciertos modelos lineales no lograrían.

Dado un conjunto de pares de parches de entrenamiento $(\mathbf{b}_i, \mathbf{s}_i), i = 1, \dots, n$ donde, \mathbf{b}_i es un parche con blur y \mathbf{s}_i es el parche correspondiente a \mathbf{b}_i sin blur. Este conjunto se usa para resolver el siguiente problema de optimización:

$$\min_{\mathbf{D}^b, \mathbf{D}^s, \mathbf{W}} \frac{1}{n} \sum_{i=1}^n \|\mathbf{s}_i - \mathbf{W}\hat{\mathbf{b}}_i - \mathbf{D}_s \boldsymbol{\alpha}^*(\mathbf{b}_i, \mathbf{D}^b)\|_2^2, \quad (2.10)$$

donde $\boldsymbol{\alpha}^*(\mathbf{b}_i, \mathbf{D}^b)$ es la solución al siguiente problema de optimización:

$$\boldsymbol{\alpha}^*(\mathbf{b}_i, \mathbf{D}^b) = \underset{\boldsymbol{\alpha} \in \mathbb{R}^M}{\operatorname{argmín}} \|\mathbf{b}_i - \mathbf{D}^b \boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1. \quad (2.11)$$

El parche $\hat{\mathbf{b}}_i$ es una versión sin ruido del parche \mathbf{b}_i , esta versión es obtenida aplicando previamente un algoritmo de eliminación de ruido a la imagen con blur de donde posteriormente \mathbf{b}_i es extraído. Las matrices \mathbf{D}^b y \mathbf{D}^s son dos diccionarios aprendidos conjuntamente de manera que para todo i , $\mathbf{W}\hat{\mathbf{b}}_i + \mathbf{D}_s \boldsymbol{\alpha}^*(\mathbf{b}_i, \mathbf{D}^b)$ es un buen estimador para el parche sin blur \mathbf{s}_i . La matriz \mathbf{W} es un operador de deblurring que transforma $\hat{\mathbf{b}}_i$ a su versión sin blur,

por lo tanto, el término que depende de los diccionarios en la Eq. (2.11) pretende recuperar información de altas frecuencias que no es posible utilizando únicamente \mathbf{W} .

Una de las aportaciones que se realizan en este trabajo es la introducción de la matriz \mathbf{W} que representa los parámetros de transformación entre los parches con blur y los que no tienen blur, además de que se promueve a que ambos diccionarios compartan la misma representación sparse α^* .

Los resultados presentados en este trabajo mejoran la calidad visual y el PSNR en comparación con el método de Yang et al. (2010). Sin embargo, recientes enfoques sugieren que el uso de técnicas de agrupamiento en combinación con otras técnicas de aprendizaje maquina pueden generar mejores resultados. Es importante señalar que parte del trabajo presentado en esta tesis es inspirado en este artículo.

2.3.2. Enfoques Previos que Usan Técnicas de Agrupamiento

2.3.2.1. Diccionarios Semi-Acoplados Usados en la Solución de Single Image Super Resolution y Photo-Sketch Synthesis

El trabajo presentado por Wang et al. (2012) fue desarrollado para resolver el problema de *Photo-Sketch Synthesis*, este problema consiste en convertir una imagen en un estilo a otra imagen de estilo diferente, con el fin de obtener una mejor visualización o facilitar el reconocimiento de algún objeto en la imagen. Debido a la similitud entre los problemas de *Photo-Sketch Synthesis* y Single Image Super Resolution los autores aplicaron su propuesta en resolver el problema de Single Image Super Resolution. Este trabajo está basado en uso de diccionarios con un enfoque similar a al trabajo de Yang et al. (2010). El modelo que se presenta en este trabajo es el siguiente:

$$\begin{aligned} & \min_{\{\mathbf{D}^x, \mathbf{D}^y, \mathbf{W}\}} \|\mathbf{X} - \mathbf{D}^x \mathbf{A}^x\|_F^2 + \|\mathbf{Y} - \mathbf{D}^y \mathbf{A}^y\|_F^2 + \\ & \gamma \|\mathbf{A}^y - \mathbf{W} \mathbf{A}^x\|_F^2 + \lambda^x \|\mathbf{A}^x\|_1 + \lambda^y \|\mathbf{A}^y\|_1 + \lambda^w \|\mathbf{W}\|_F^2 \\ & \text{s.t. } \|\mathbf{d}_i^x\|_2 \leq 1, \|\mathbf{d}_i^y\|_2 \leq 1, \forall i, \end{aligned} \quad (2.12)$$

donde, $\mathbf{A}^x = (\alpha_1^x, \alpha_2^x, \dots, \alpha_N^x)$ y $\mathbf{A}^y = (\alpha_1^y, \alpha_2^y, \dots, \alpha_N^y)$ son matrices que contienen los coeficientes o códigos de la representación sparse a través de los diccionarios de alta y baja resolución respectivamente, la matriz \mathbf{W} representa una relación entre los códigos \mathbf{A}^x y \mathbf{A}^y , y $\gamma, \lambda^x, \lambda^y, \lambda^w$ son parámetros que controlan que tan sparse son los coeficientes

\mathbf{A}^x , \mathbf{A}^y y la matriz \mathbf{W} .

En una propuesta anterior (Yang et al., 2010) la matriz \mathbf{W} (diferente que en el método de Couzinie-Devy et al. (2011)) se asume como la identidad y los coeficientes \mathbf{A}^x y \mathbf{A}^y se asumen como iguales. En este trabajo los autores decidieron usar diferentes matrices de coeficientes e introducir una matriz \mathbf{W} que contendrá la relación entre estos coeficientes. De manera que para el proceso de estimación de un nuevo parche de alta resolución (o de otro estilo) se resuelve el siguiente problema de optimización:

$$\hat{\boldsymbol{\alpha}}_y = \min_{\boldsymbol{\alpha}_y} \|\mathbf{y} - \mathbf{D}^y \boldsymbol{\alpha}^y\|_2^2 + \|\boldsymbol{\alpha}^y\|_1, \quad (2.13)$$

luego, se estima el vector de coeficientes sobre el diccionario de alta resolución \mathbf{D}^x resolviendo el siguiente problema:

$$\hat{\boldsymbol{\alpha}}^x = \min_{\boldsymbol{\alpha}_x} \|\boldsymbol{\alpha}^y - \mathbf{W} \boldsymbol{\alpha}^x\|_F^2, \quad (2.14)$$

finalmente, una vez que se evaluó $\hat{\boldsymbol{\alpha}}^x$ el parche de alta resolución es estimado de la siguiente manera:

$$\hat{\mathbf{x}} = \mathbf{D}^x \hat{\boldsymbol{\alpha}}^x. \quad (2.15)$$

Para el proceso de optimización se divide la Eq. (2.9) en dos partes. La primer parte se usa para optimizar los valores de los diccionarios usando una estrategia de optimización de uno-por-uno (descenso coordinado) y la matriz de coeficientes usando LARS (Efron et al., 2004). En este primer parte la matriz \mathbf{W} se considera inicialmente como la identidad y se optimiza únicamente sobre los diccionarios y las matrices de coeficientes, dejando únicamente una incógnita (ya sea un diccionario o un vector de coeficientes) a la vez. De esta manera se puede optimizar usando las técnicas mencionadas anteriormente. Una vez que se optimizaron los diccionarios y los vectores de coeficientes se puede estimar de manera analítica el valor óptimo para la matriz \mathbf{W} . El proceso de optimización es iterativo primero se optimiza sobre los diccionarios y vectores de coeficientes fijando la matriz \mathbf{W} y como segundo paso se optimiza la matriz \mathbf{W} , esto se repite hasta convergencia.

En este trabajo también se introduce una técnica de agrupamiento. Basados en la suposición de que las estructuras en los parches de diferentes estilos son complejas, los autores sugieren crear diferentes grupos formado por parches con estructuras similares, tratando de

abrir una mayor cantidad de variantes dentro de los parches y aprender el mismo volumen de pares de diccionarios como cantidad de grupos se tengan.

2.3.2.2. Super Resolution Usando Funciones Simples

Uno de los principales objetivos del método propuesto por Yang and Yang (2013) es proponer un algoritmo que sea eficiente en términos computacionales. La forma en que se busca cumplir este objetivo es la siguiente: Dado un conjunto de entrenamiento compuesto por parches de baja resolución y sus parches correspondientes en alta resolución, se divide este conjunto de entrenamiento en K subconjuntos y para cada subconjunto se estiman los parámetros de una función simple (una función de bajo coste computacional al momento de evaluarla) que es capaz de transformar un parche de baja resolución a su versión en alta resolución.

Suponiendo que hay l parches de baja resolución que pertenecen al mismo subconjunto. Define \mathbf{y}_i y \mathbf{x}_i , $i = 1, \dots, l$ como dos vectores de características de los parches de baja resolución (\mathbf{y}_i) y los parches de alta resolución (\mathbf{x}_i), con dimensión m y n respectivamente. Se propone aprender un conjunto de n funciones lineales de regresión para predecir las n características del parche de alta resolución. Estas características son simplemente los valores de las intensidades de cada píxel en cada parche. Define $\hat{\mathbf{Y}} \in \mathbb{R}^{m \times l}$ y $\hat{\mathbf{X}} \in \mathbb{R}^{n \times l}$ como las matrices que contienen al conjunto de l parches \mathbf{y}_i y \mathbf{x}_i . Se evalúan los coeficientes de regresión $C^* \in \mathbb{R}^{n \times (m+1)}$ de la siguiente manera:

$$C^* = \underset{C}{\operatorname{argmín}} \|\hat{\mathbf{X}} - C(\hat{\mathbf{Y}})\|_2^2, \quad (2.16)$$

donde $\mathbf{1}$ es un vector de tamaño $1 \times l$ con todos los valores iguales a 1. Este es un problema lineal de mínimos cuadrados que se resuelve fácilmente.

Una vez estimados la matriz de coeficientes \hat{C}^* , la estimación de un parche de alta resolución $\hat{\mathbf{x}}$ a partir de un parche de baja resolución \mathbf{y} se realiza resolviendo la siguiente expresión:

$$\hat{\mathbf{x}} = \hat{C}^*(\hat{\mathbf{Y}}) \quad (2.17)$$

El proceso de entrenamiento (estimar los coeficientes C^*) se realiza para cada subconjunto del conjunto de pares de parches de entrenamiento, de manera que se tendrán K matrices de coeficientes. Luego, para el proceso de reconstrucción, se usa la matriz de

coeficientes que fue estimada con el subconjunto cuyo centroide (promedio de los parches de baja resolución del subconjunto de entrenamiento) es más parecido al parche de baja resolución.

Finalmente, de manera similar al trabajo de Yang et al. (2010), se aborda el mal condicionamiento del problema de Single Image Super Resolution agregando una restricción global de reconstrucción aplicando el algoritmo presentado por Irani and Peleg (1991). Esta restricción se aborta en el Capítulo 3 de este trabajo de tesis.

2.3.2.3. Uso de PCA para la Estimación de Diccionarios

El trabajo presentado por Dong et al. (2011) también se hace uso de diccionarios y agrupamiento con el objetivo de resolver los problemas de deblurring y Super Resolution. En esta propuesta hay dos diferencias importantes respecto a las propuestas anteriores. La primer diferencia es el uso de un único diccionario (en lugar del par de diccionarios D^y y D^x) para representar la imagen de alta resolución. La segunda diferencia es el uso de PCA para obtener los valores de los átomos del diccionario Φ , bajo el supuesto de que no es necesario evaluar un diccionario sobre-completo si los parches han sido previamente agrupados ya que estos grupos contienen parches con estructuras similares.

En varias propuestas (incluyendo este trabajo de tesis) existe una restricción global de reconstrucción que garantiza que la correcta relación entre la imagen de baja resolución y la imagen de alta resolución estimada. En este trabajo esta relación es absorbida por el modelo matemático propuesto.

En cuanto a la parte del agrupamiento, también se hace uso de PCA. Se aplica PCA a cada parche de tal manera que únicamente las bases más significativas sean usadas para determinar a qué grupo pertenecerá un nuevo parche de baja resolución. Se argumenta que este enfoque ayuda a que se reduzca una mala estimación si el nuevo parche de baja resolución contiene ruido, sin embargo. Esto también podría ocasionar, en caso de que el ruido sea poco significativo o no exista, ignorar información de altas frecuencias que podría quedar en las bases poco "significativas.^{a1} aplicar PCA.

En general este algoritmo da buenos resultados visuales, sin embargo, generalmente imágenes desalineadas con respecto a la imagen real. Este desalineamiento provoca que no sea posible medir la calidad de la imagen en términos de PSNR o SSIM.

2.3.3. Algoritmos Basados en Redes Convolucionales

Aunque las redes convolucionales datan de algunas décadas atrás (LeCun et al., 1989), recientemente se ha popularizado su uso en el área de procesamiento de imágenes. Algunos trabajos en restauración de imágenes son los presentados por Jain and Seung (2009) con un trabajo de eliminación de ruido, Eigen et al. (2013) con su trabajo para restaurar imágenes tomadas a través de una ventana con gotas de agua o sucia y el trabajo en Single Image Super Resolution de Dong et al. (2014).

En esta Sección explicamos de manera breve en que consiste el trabajo presentado por Dong et al. (2014) por ser de especial interés en esta tesis.

2.3.3.1. Single Image Super-Resolution Usando Redes Convolucionales

Aplicar una red convolucional para resolver el problema de Single Image Super Resolution es una de las principales contribuciones de este trabajo. Otra contribución es que los autores relacionan su trabajo con trabajos previos basados en diccionarios y en representación sparse. Como resultado de este trabajo se obtuvo un algoritmo rápido (usando GPU y cuda) que genera buenos resultados visuales.

La propuesta que se plantea en este trabajo se divide en tres partes:

- Extracción y representación de parches.- Esta operación es común a la mayoría de los algoritmos de Single Image Super Resolution y consiste en extraer parches(sobrelapados) de una imagen Y . Se toma la representación de cada parche en forma de vector de alta dimensión (tamaño del parche).
- Mapeo no lineal.- Esta operación consiste en mapear cada parche de alta dimensión a otro parche de alta dimensión. Cada vector resultante del mapeo representa un parche de alta resolución.
- Reconstrucción.- Esta parte consiste en tomar los parches de alta resolución obtenidos de las dos etapas anteriores y usarlos para reconstruir una estimación final de una imagen de alta resolución.

Los 3 pasos anteriores en general se aplican en cualquier algoritmo de Single Image Super Resolution, sin embargo, en este trabajo cada una de esas etapas es vista como una capa de una red convolucional y todas las operaciones se representan como convoluciones entre un banco de filtros y los parches en forma de vector.

2.3.4. Conclusión Sobre los Algoritmos que Usan el Aprendizaje Ma-quina

En general los algoritmos *Exampled Based* o *Basados en Ejemplos* han demostrado generar buenos resultados. Existen diversos enfoques que se han presentado, sin embargo, la representación sparse y el uso de diccionarios han logrado sobresalir entre la mayoría de las propuestas debido a sus buenos resultados. Recientemente el trabajo de Dong et al. (2014) ha demostrado que las redes convolucionales pueden ser el próximo enfoque que logre obtener los mejores resultados con un menor coste computacional.

Capítulo 3

Metodología de las Soluciones Propuestas

De manera general los algoritmos de Single Image Super Resolution clasificados como Exampled-Based pueden dividirse en dos etapas, la etapa de **entrenamiento** y la de **reconstrucción**. Estas dos etapas son independientes entre sí. La primera implica un problema de estimación de parámetros y la segunda es una aplicación donde se usan los parámetros estimados en la primer etapa. Previo a la etapa de entrenamiento, se debe proponer un modelo matemático que represente la relación entre un parche de baja resolución y su versión en alta resolución. Son precisamente los parámetros de este modelo los que son estimados durante la etapa de entrenamiento. Una vez que el modelo es definido y sus parámetros son estimados se lleva a cabo la etapa de reconstrucción. Durante la etapa de reconstrucción, una nueva imagen de baja resolución es convertida en un conjunto de parches de baja resolución y la versión de alta resolución de cada parche de este conjunto es generada usando el modelo con los parámetros previamente estimados. Finalmente, el conjunto estimado de parches de alta resolución se usa para reconstruir la estimación final de la imagen de alta resolución.

El resto de este Capítulo se divide de la siguiente manera; En la Sección **Panorama General** (Sec. 3.1) se argumentan las ideas que motivaron el desarrollo de este trabajo y así mismo se explica la estructura general de los algoritmos de Single Image Super Resolution presentados en esta tesis. En la Sección **Modelos** (Sec. 3.2) se introducen y explican dos modelos matemáticos que posteriormente serán usados en la etapa de reconstrucción para estimar parches de alta resolución. El primero de estos modelos está basado en redes neuronales artificiales y el segundo modelo está basado en diccionarios. En la Sección **Estimación de los Parámetros de los Modelos** (Sec. 3.3) se detalla el proceso para estimar los parámetros de los modelos presentados previamente. Este proceso se divide en dos partes, la primera para explicar como se lleva a cabo el entrenamiento de las redes neuronales

artificiales y la segunda para explicar como estimar los diccionarios. En la Sección **Reconstrucción de una Imagen de Alta Resolución** (Sec. 3.4) se explica como usar los modelos presentados en la Sec. 3.2 para estimar parches de alta resolución que serán usados para reconstruir la imagen de alta resolución. En esta Sección también se dan algunos tips para reducir el tiempo computacional a la hora de reconstruir una imagen de alta resolución. De esta forma el trabajo presentado no genera únicamente buenos resultados, si no que es más rápido que la mayoría de los algoritmos del estado del arte. Finalmente, está la Sección **Factores de Escala** (Sec. 3.5) donde se dan algunos comentarios sobre como usar distintos factores de escala en los algoritmos propuestos.

3.1. Panorama General

Cualquier imagen natural (inclusive de otras clases) puede representarse como un conjunto de parches pequeños. Mientras más pequeños sean estos parches, la probabilidad de encontrar otro parche similar en la misma imagen (o en otra imagen de la misma clase) aumenta. Supongamos que los parches son de tamaño 1×1 (un píxel), entonces existirán solamente 255 parches diferentes, los cuales con seguridad aparecerán repetidas veces a lo largo de la imagen y conforme el tamaño del parche aumente la probabilidad de encontrar parches similares se reduce considerablemente. En trabajos previos como en el presentado por Buades et al. (2005) para eliminar ruido en imágenes naturales, se argumenta que existen estructuras de parches pequeños para las cuales se pueden encontrar otras estructuras muy similares a lo largo de las imágenes naturales. Estos parches no pueden ser de tamaño 1×1 ya que deben incorporar información de píxeles vecinos y no deben ser muy grandes por las razones explicadas anteriormente, es por ello que en la literatura es común encontrar recomendaciones de tamaño que oscilan entre 5×5 y 11×11 píxeles. Dentro de las estructuras o formas más comunes que se pueden encontrar dentro del conjunto de parches pequeños se pueden observar líneas rectas, esquinas, líneas curvas, zonas planas (es decir, sin estructura), entre otras. Algoritmos como los presentados por (Buades et al., 2005), (Freedman and Fattal, 2010), (Glasner et al., 2009), etc. hacen uso de esta redundancia de información a nivel de parches pequeños para llevar a cabo procesos de restauración de imagen. En la Fig. 3.1 se muestra un ejemplo de lo explicado anteriormente.

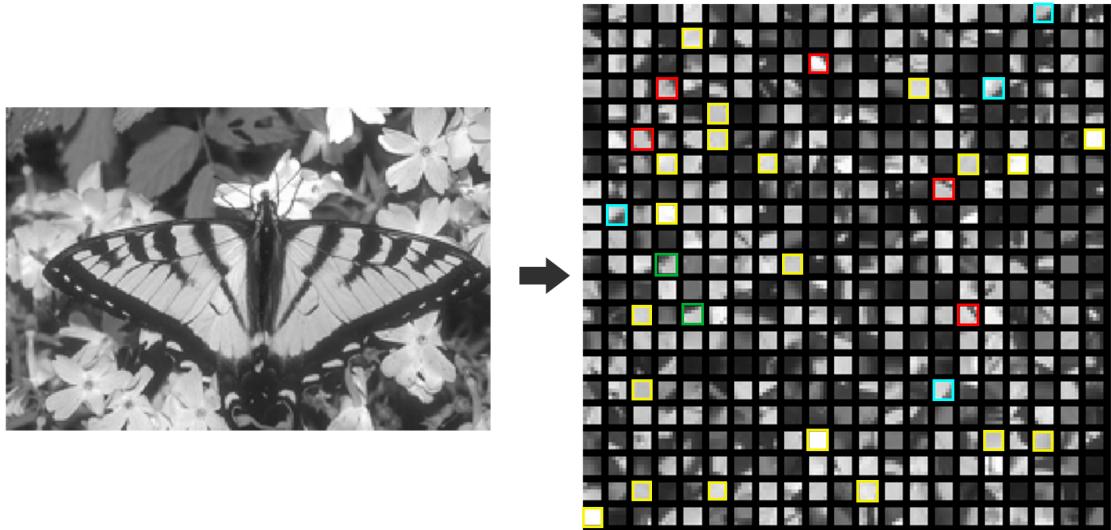


Figura 3.1: Representación de una imagen natural usando pequeños parches.

Se pueden observar 400 parches de tamaño 5×5 muestreados de la imagen de mariposa. En su mayoría se observan parches con zonas planas (sin frecuencias altas), también se pueden observar parches que asemejan líneas ya sean curvas o rectas y esquinas. Es común observar este tipo de estructuras en las imágenes naturales. Los parches marcados con un cuadrado del mismo color representan diferentes parches con estructura similar.

En base a lo anterior se propone la siguiente función para representar la transformación entre un parche de baja resolución a su versión en alta resolución:

$$\mathbf{x} = f(\mathbf{y}|\Theta), \quad (3.1)$$

que representa la transformación entre un parche de baja resolución \mathbf{y} y su versión en alta resolución \mathbf{x} , con $\mathbf{y} \in \mathbb{R}^P$, $\mathbf{x} \in \mathbb{R}^P$.

El proceso de estimación de los parámetros Θ de la Eq. (3.1) se denomina **entrenamiento**. Si la función de la Eq. (3.1) y sus parámetros se estiman correctamente, entonces el problema de estimación de la versión de alta resolución para una imagen de baja resolución dada se reduce a transformar esta imagen en un conjunto de parches, estimar la versión de alta resolución para este conjunto de parches y usarlos para reconstruir la nueva imagen en alta resolución. Este último proceso se conoce como **reconstrucción**.

El proceso de entrenamiento se define de la siguiente manera; Dada una base de datos de entrenamiento formada por pares de parches indizados $G^J = \{(\mathbf{y}_j, \mathbf{x}_j) | \mathbf{y}_j \in G_y^J, \mathbf{x}_j \in G_x^J, j = 1, 2, \dots, J\}$ donde, $\mathbf{y}_j \in \mathbb{R}^P$ es un parche en baja resolución, $\mathbf{x}_j \in \mathbb{R}^P$ es su versión en alta resolución, se puede estimar una función paramétrica $\mathbf{x}_j = f(\mathbf{y}_j|\Theta)$, $f : \mathbb{R}^P \rightarrow \mathbb{R}^P$ que aplica una transformación a un parche de baja resolución \mathbf{y}_j para generar su versión en alta resolución \mathbf{x}_j . G_y^J y G_x^J son conjuntos formados por parches de baja y alta resolución respectivamente. En la Fig. 3.2 se muestra un diagrama que describe el proceso de entrenamiento.

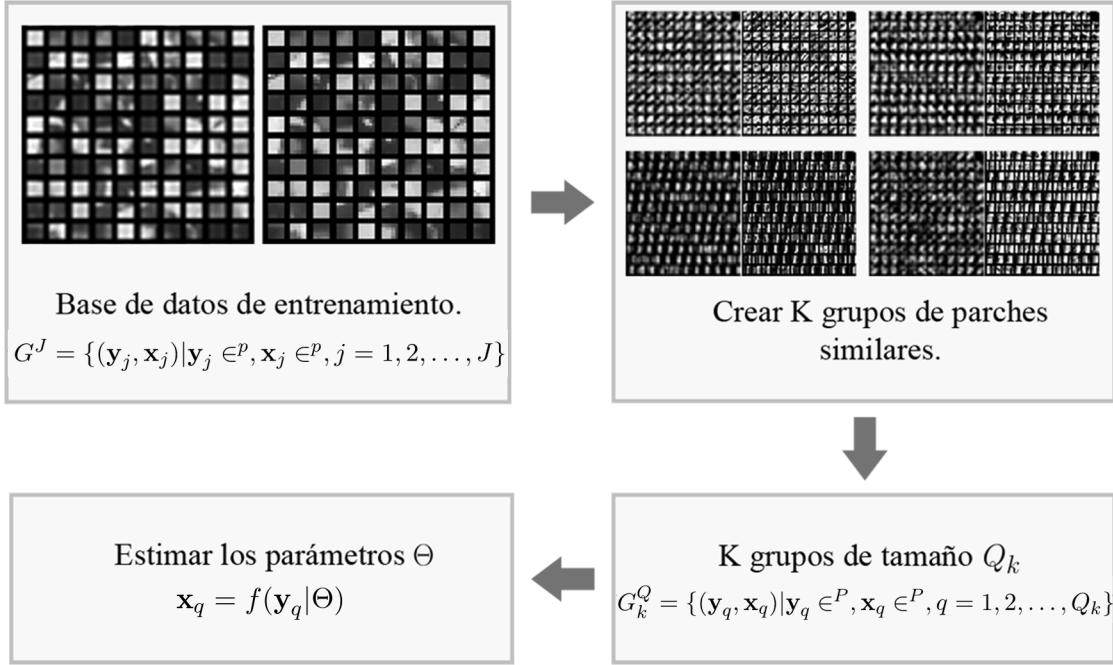


Figura 3.2: Diagrama del Proceso de Entrenamiento.

El proceso de entrenamiento propuesto en esta tesis consta de dos pasos; El primer paso consiste en crear K grupos de pares de parches (baja y alta resolución) donde cada grupo deberá contener parches con estructura similar, con la finalidad de que los modelos propuestos puedan ser entrenados usando cada uno de estos grupos de parches de manera independiente y así obtener una mejor estimación de los parches de alta resolución. El segundo paso se define como la estimación de los parámetros Θ . Este proceso depende del modelo definido para la Eq. (3.1). Por cuestiones de visualización se presentan los parches en forma de matriz pero en el proceso estos parches son transformados en vectores, es decir, cambiamos de dimensión cada parche $\mathbb{R}^{a \times a} \rightarrow \mathbb{R}^P$.

La estimación de la función Eq. (3.1) es la parte más importante de los algoritmos Example-Based. Proponer un modelo matemático tan simple como sea posible y además una buena manera de estimar los parámetros de este modelo son los retos que se plantean en la parte de entrenamiento. Un modelo simple supone una forma fácil y rápida para eva-

luarlo. Esto se refleja en algoritmos con un buen tiempo computacional, lo cual es deseado en mucha aplicaciones.

Por la manera en que se plantea el proceso de entrenamiento es adecuado pensar en usar técnicas de aprendizaje por computadora para estimar de los parámetros Θ . En este trabajo de tesis se proponen dos modelos matemáticos para la Eq. (3.1). El primer modelo hace uso de un conjunto de redes neuronales artificiales, donde cada red aprenderá a priori la transformación de parches con estructuras similares. Esto se logra aplicando una técnica de agrupamiento como pre-proceso a la base de datos de entrenamiento. El segundo modelo propuesto tiene como base un conjunto de pares de diccionarios, donde cada par de diccionarios está formado por un diccionario D^y que representa información de parches de baja resolución y por un diccionario D^x que representa información de parches de alta resolución. De la misma manera que el modelo de redes neuronales artificiales, cada par de diccionarios representa información de parches con estructura similar. En las Secciones 3.2.1 y 3.2.2 de este Capítulo, se explican a detalle los modelos propuestos y en la Sec. 3.3.3 se explica el porque de usar una técnica de agrupamiento.

Una vez que se ha definido el modelo a usar y se han estimado sus parámetros correctamente, el proceso de reconstrucción de la imagen de alta resolución dada una imagen de baja resolución se resume como se muestra en la Fig. 3.3.

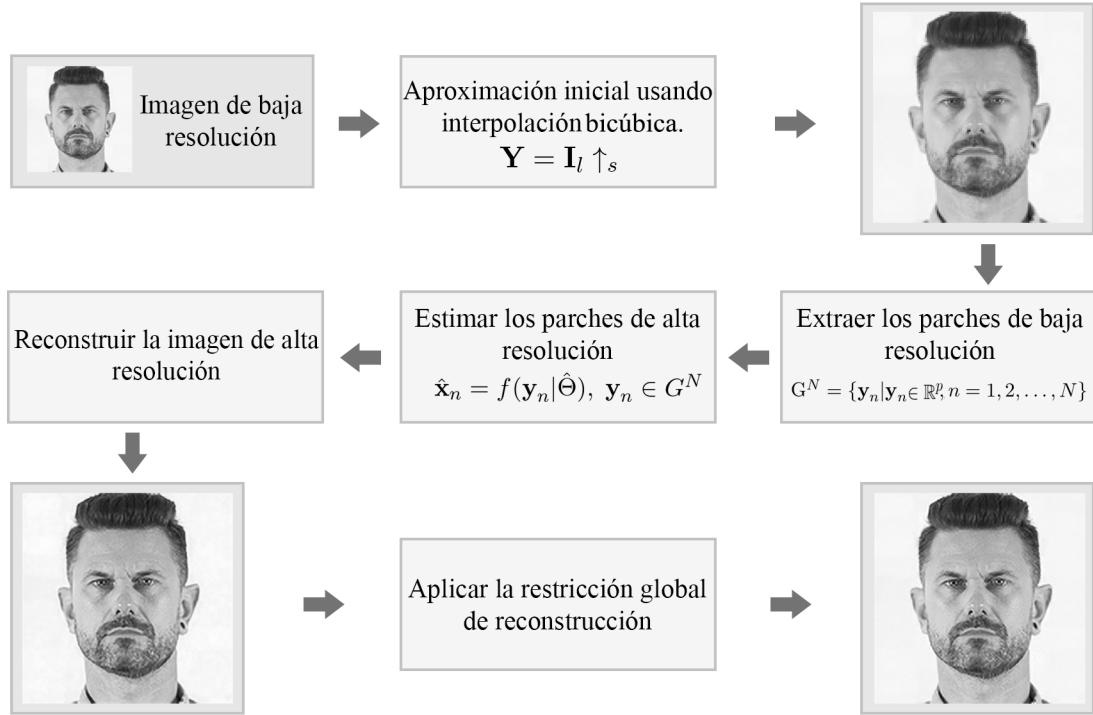


Figura 3.3: Diagrama de Flujo para el Proceso de Reconstrucción.

La Figura 3.3 representa el proceso de reconstrucción. Dada una imagen de baja resolución \mathbf{I}_l y los parámetros $\hat{\Theta}$ del modelo, se siguen una serie de pasos para la estimación de la imagen de alta resolución. El primer paso es incrementar la resolución espacial de la imagen de baja resolución $\mathbf{Y} = (\mathbf{I}_l) \uparrow_s$ por un factor de escala s . El segundo paso es extraer todos los parches de \mathbf{Y} con cantidad de elementos (píxeles) $P = a \times a$. Se propone el uso de condiciones de frontera simétricas para este paso. El conjunto de estos parches de baja resolución se define como $G_y^N = \{\mathbf{y}_n | \mathbf{y}_n \in \mathbb{R}^P, n = 1, 2, \dots, N\}$, donde N es el número total de píxeles en \mathbf{Y} . El tercer paso es estimar los parches de alta resolución \mathbf{x}_n correspondientes a cada \mathbf{y}_n usando la siguiente función:

$$\mathbf{x}_n = f(\mathbf{y}_n | \Theta), \forall n \in N. \quad (3.2)$$

El cuarto paso es reconstruir la imagen de alta resolución usando el conjunto de parches de alta resolución estimados en el tercer paso. Finalmente, se aplica un algoritmo de Back-Projection (Irani and Peleg, 1991) para mantener la restricción global del problema de Super Resolution.

El resto de este Capítulo se enfoca en explicar los pasos generales de entrenamiento y reconstrucción. Primero se presentan y explican los modelos propuestos y la forma de estimar los parámetros para cada modelo. Posteriormente, se detalla el proceso de reconstrucción.

3.2. Modelos

En este trabajo se proponen 2 modelos matemáticos usados para definir la función de la Eq. (3.1), donde Θ dependerá de cada modelo y de los datos usados durante el entrenamiento. El primer modelo usa un conjunto de K redes neuronales artificiales. Este conjunto se define como $G^{\text{ANN}} = \{\text{ANN}_k(\mathbf{y}|\Theta_k^{\text{ANN}}) | k = 1, 2, \dots, K\}$, donde $\text{ANN}_k(\mathbf{y}|\Theta_k^{\text{ANN}})$ representa la k -ésima red neuronal artificial con parámetros Θ_k^{ANN} . El segundo modelo hace uso de un conjunto de K pares de diccionarios indizados. Este conjunto se define como $G^D = \{(\mathbf{D}_k^y, \mathbf{D}_k^x) | \mathbf{D}_k^y \in G_y^D, \mathbf{D}_k^x \in G_x^D, k = 1, 2, \dots, K\}$, donde $\mathbf{D}_k^y \in \mathbb{R}^{P \times M}$ y $\mathbf{D}_k^x \in \mathbb{R}^{P \times M}$ representan diccionarios entrenados con parches de baja resolución y sus correspondientes en alta resolución y G_y^D y G_x^D son conjuntos formados por diccionarios de baja y alta resolución respectivamente.

Cada uno de los modelos propuestos hace uso de un conjunto de tamaño K , ya sea de redes neuronales artificiales o pares de diccionarios. Aplicando previamente una técnica de agrupamiento al conjunto de entrenamiento G^J , se obtienen K grupos con parches similares. Luego cada modelo aprende la relación entre baja resolución y alta resolución para cada una de estas formas básicas. Más delante en este Capítulo se explica a detalle la parte de agrupamiento.

Si los parámetros de cada modelo son bien estimados, los parches de alta resolución se estiman fácilmente evaluando estos modelos. Sin embargo, el problema de Single Image Super Resolution es altamente mal condicionado. Esto significa que para una imagen de baja resolución existen infinitas imágenes de alta resolución que pueden haberla generado. Un ejemplo se muestra en la Fig. 3.4. Entonces, ¿Cómo determinar cuál de todas estas posibles soluciones es la correcta o cuál de ellas se acerca más a la solución real?, el término de **Restricción de Reconstrucción Local** se aplica precisamente para hacer referencia a este problema y además explica el porque los modelos propuestos logran generar una solución cercana a la solución real.

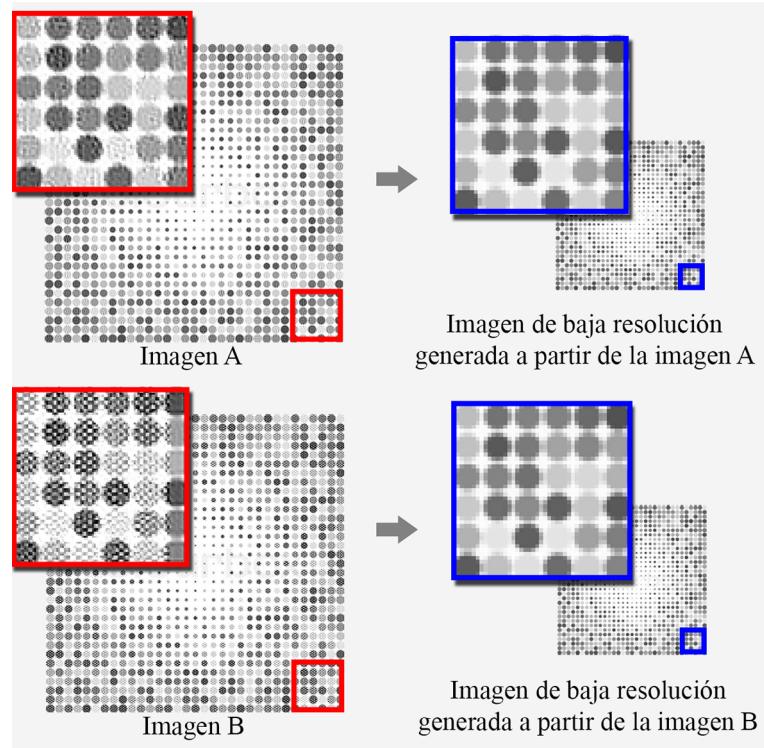


Figura 3.4: Single Image Super Resolution: un problema mal condicionado.
Las imágenes A y B son diferentes, sin embargo, al aplicar el modelo de degradación se obtienen las mismas imágenes de baja resolución.

La restricción de reconstrucción local se introduce para asegurar que la solución obtenida con los modelos propuestos será suficientemente cercana a la solución real. Indirectamente se regulariza el problema al estimar los parámetros de cada modelo usando parches pequeños extraídos de una base de datos de imágenes naturales. De esta manera se está restringiendo a que los parches de alta resolución generados por los modelos contengan formas que se encuentran en imágenes naturales. Además, en el proceso de reconstrucción una vez que sean estimado todos los parches de alta resolución se lleva a cabo una reconstrucción con solapamiento de parches, lo que ayuda a mantener una compatibilidad entre los parches en el mismo vecindario.

3.2.1. Modelo Basado en Redes Neuronales Artificiales

Las redes neuronales artificiales han jugado un papel muy importante en varias aplicaciones, desde su uso en reconocimiento de caracteres (Rajavelu et al., 1989), hasta su uso como agente inteligente en videojuegos (Stanley et al., 2005), además de que su desempeño es eficiente en comparación con otras técnicas, ya que una vez que la red está entrenada la evaluación para una nueva observación se reduce a una serie de multiplicaciones. Por estas

razones se decide proponer el siguiente modelo basado en redes neuronales artificiales:

$$\mathbf{x}_n = f(\mathbf{y}_n | G^{\text{ANN}}) = \frac{1}{W_n} \sum_{k=1}^K w(\boldsymbol{\mu}_k, \mathbf{y}_n) \text{ANN}_k(\mathbf{y}_n | \hat{\Theta}_k^{\text{ANN}}), \quad \forall n \in N, \quad (3.3)$$

donde, $G^{\text{ANN}} = (\text{ANN}_1(\cdot | \hat{\Theta}_1^{\text{ANN}}), \text{ANN}_2(\cdot | \hat{\Theta}_2^{\text{ANN}}), \dots, \text{ANN}_K(\cdot | \hat{\Theta}_K^{\text{ANN}}))$ representa un conjunto de tamaño K de redes neuronales artificiales, $\hat{\Theta}_k^{\text{ANN}}$ representan los parámetros de cada una de las K redes neuronales artificiales (pesos de la red), W_n es un factor de normalización definido como:

$$W_n = \sum_{k=1}^K w(\boldsymbol{\mu}_k, \mathbf{y}_n), \quad (3.4)$$

y $w(\boldsymbol{\mu}_k, \mathbf{y}_n)$ es una función con valores en el intervalo $[0, 1]$ que representa la similaridad entre el parche de baja resolución \mathbf{y}_n y la media $\boldsymbol{\mu}_k$ de los parches usados para entrenar la k -ésima red neuronal artificial, \mathbf{x}_n es el parche de alta resolución estimado para \mathbf{y}_n .

Note que la función de pesos se puede reemplazar por otra función donde, a manera de caso especial, se obtendrían los valores de $w(\boldsymbol{\mu}_k, \mathbf{y}_n) = 1$ si la media $\boldsymbol{\mu}_k$ es la más cercana a \mathbf{y}_n , casos contrarios $w(\boldsymbol{\mu}_k, \mathbf{y}_n) = 0$. Esto significa que sólo se usa una red neuronal artificial para estimar el parche de alta resolución correspondiente a \mathbf{y}_n . Esto supone una ventaja computacional ya que sólo se evalúa una red neuronal artificial por cada parche \mathbf{y}_n . Sin embargo, al utilizar más de una red neuronal es mas probable que algunos artefactos introducidos por una mala estimación de una red neuronal artificial se reduzcan, ya que se toman en cuenta (con diferentes pesos) muchas estimaciones, lo que hace a este modelo más robusto.

La función $w(\boldsymbol{\mu}_k, \mathbf{y}_n)$ se puede definir de muchas maneras. En este trabajo se sugiere utilizar la siguiente definición:

$$w(\boldsymbol{\mu}_k, \mathbf{y}_n) = \exp^{\frac{-\text{MSE}(\boldsymbol{\mu}_k, \mathbf{y}_n)^2}{h^2}}, \quad (3.5)$$

donde h es un parámetro que controla la cantidad de redes neuronales artificiales que se usarán en la estimación de un parche de alta resolución, es decir, pesos muy pequeños se consideran como cero y por lo tanto la red neuronal artificial asociada a ese peso no participará en la estimación del parche de alta resolución. h además, ayuda a regular los pesos de cada red neuronal que participa en la estimación. En el Capítulo de resultados (Cap. 4) se detalla un poco más el efecto de este parámetro.

La función MSE se define como:

$$\text{MSE}(\boldsymbol{\mu}_k, \mathbf{y}_n) = \frac{1}{n} \sum_{p=1}^P (\boldsymbol{\mu}_k(p) - \mathbf{y}_n(p))^2, \quad (3.6)$$

donde, p representa es un índice dentro de los vectores $\boldsymbol{\mu}$ y \mathbf{y}_n y P es la cantidad de elementos en cada vector o tamaño del parche.

Este modelo es simple y una vez que las redes neuronales artificiales han sido entrenadas, estimar una nueva imagen de alta resolución es bastante rápido en comparación con la mayoría de los algoritmos de Single Image Super Resolution previamente propuestos por otros autores.

3.2.2. Modelo Basado en Diccionarios

Como se mencionó en el Capítulo 2, el uso de diccionarios y representación sparse se ha vuelto muy popular en años recientes como solución al problema de Super Resolution. Sin embargo, muchos de los modelos propuestos no toman ventaja de la redundancia de información que existe en las imágenes naturales. Además, varios de estos algoritmos son iterativos lo que resulta en soluciones que requieren resolver varios problemas de optimización repetidamente. Esto los hace lentos y deja camino para generar nuevas propuestas.

Esta tesis propone un modelo basado en diccionarios, que toma ventaja de la información redundante en las imágenes naturales y que no es un proceso iterativo. El modelo se define como:

$$\mathbf{x}_n = f(\mathbf{y}_n | G^D) = \frac{1}{W_n} \sum_{k=1}^K w(\boldsymbol{\mu}_k, \mathbf{y}_n) \hat{\mathbf{D}}_k^x \boldsymbol{\alpha}^*(\hat{\mathbf{D}}_k^y, \mathbf{y}_n), \quad \forall n \in N, \quad (3.7)$$

donde, $G^D = \{(\hat{\mathbf{D}}_k^y, \hat{\mathbf{D}}_k^x) | \hat{\mathbf{D}}_k^y \in G_y^D, \hat{\mathbf{D}}_k^x \in G_x^D, k = 0, 1, \dots, K\}$ un conjunto de tamaño K de pares de diccionarios indizados de baja resolución $\hat{\mathbf{D}}_k^y \in \mathbb{R}^{P \times M}$ y alta resolución $\hat{\mathbf{D}}_k^x \in \mathbb{R}^{P \times M}$, donde G_y^D y G_x^D son los conjuntos que contienen a los diccionarios de baja y alta resolución respectivamente. W_n es un factor de normalización definido como:

$$W_n = \sum_{k=1}^K w(\boldsymbol{\mu}_k, \mathbf{y}_n), \quad (3.8)$$

donde $w(\boldsymbol{\mu}_k, \mathbf{y}_n)$ es una función que representa la similaridad entre \mathbf{y}_n y la media $\boldsymbol{\mu}_k$ de los parches usados para entrenar el k -ésimo diccionario de baja resolución, x_n es el parche de alta resolución estimado para \mathbf{y}_n y $\boldsymbol{\alpha}^*(\hat{\mathbf{D}}_k^y, \mathbf{y}_n)$ es el resultado de un proceso de optimización desacoplado que depende de $\hat{\mathbf{D}}_k^y$ y \mathbf{y}_n , y que se explica más delante. Por facilidad de notación definimos $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}^*(\hat{\mathbf{D}}_k^y, \mathbf{y}_n)$. La definición de $w(\boldsymbol{\mu}_k, \mathbf{y}_n)$ es la misma que en eq. (3.5).

Note que la función de pesos se puede reemplazar por otra función donde, a manera de caso especial, se obtendrían los valores de $w(\boldsymbol{\mu}_k, \mathbf{y}_n) = 1$ si la media $\boldsymbol{\mu}_k$ es la más cercana a \mathbf{y}_n , casos contrarios $w(\boldsymbol{\mu}_k, \mathbf{y}_n) = 0$. Esto significa que sólo se usan un par de diccionarios para estimar el parche de alta resolución correspondiente a \mathbf{y}_n . Esto supone una ventaja computacional ya que sólo se resuelve un pequeño problema de optimización por cada parche \mathbf{y}_n , las ventajas son similares a las explicadas en la Sección anterior.

Basados en la idea que se presenta en el trabajo de Couzinie-Devy et al. (2011) (ver Sec. 2.3.1.2) de restringir el vector de coeficientes $\boldsymbol{\alpha}$ sobre un par de diccionarios entrenados con parches con blur y sin blur (en nuestro caso de baja y alta resolución) a ser el mismo, se propone resolver el siguiente problema de optimización para cada parche \mathbf{y}_n :

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmín}} \|\boldsymbol{\alpha}\|_0 \quad \text{s.t.} \|\hat{\mathbf{D}}_k^y \boldsymbol{\alpha} - \mathbf{y}_n\|_2^2 \leq \epsilon_1. \quad (3.9)$$

Sin embargo, este problema de optimización no convexo es NP-Hard lo que lo vuelve extremadamente difícil de resolver. En la literatura es común encontrar trabajos que

sugieren usar la norma l_1 para aproximar la solución de la norma l_0 . En el trabajo de Carlos Ramirez and Argaez (2013) se explica el porque usar la norma l_1 genera una buena aproximación a la solución de la norma l_0 y también es posible ver en algunos de los métodos explicados en el Capítulo 2 el uso de la norma l_1 para resolver problemas similares.

El modelo usado para obtener el vector de coeficientes α^* es el siguiente:

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmín}} \|\boldsymbol{\alpha}\|_1 \quad \text{s.t.} \|\hat{\mathbf{D}}_k^y \boldsymbol{\alpha} - \mathbf{y}_n\|_2^2 \leq \epsilon_1, \quad (3.10)$$

y la formulación equivalente usando multiplicadores de Lagrange:

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmín}} \|\hat{\mathbf{D}}_k^y \boldsymbol{\alpha} - \mathbf{y}_n\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1. \quad (3.11)$$

Semi-acoplando el modelo de esta manera se restringe el vector de coeficientes $\boldsymbol{\alpha}$ a ser el mismo para los diccionarios de alta y baja resolución.

3.3. Estimación de los Parámetros de los Modelos

Al inicio de este Capítulo se propone el uso de una función $f(\cdot | \Theta)$ que es capaz de transformar un parche de baja resolución a su versión en alta resolución. Esta función requiere de ciertos parámetros Θ que deben ser estimados. En este trabajo de tesis se proponen dos modelos matemáticos para definir a la función $f(\cdot | \Theta)$. El primer de estos modelos trabaja con un conjunto de redes neuronales artificiales que anteriormente se definió como G^{ANN} . El segundo modelo está compuesto por un conjunto de pares de diccionarios definido anteriormente como G^D . En esta sección se detalla como se lleva a cabo la estimación de los parámetros para cada modelo propuesto.

Retomando las definiciones hechas en anteriormente en este documento; El conjunto de pares de parches de entrenamiento (baja y alta resolución) fue definido como $G^J = \{(\mathbf{y}_j, \mathbf{x}_j) | \mathbf{y}_j \in G_y^J, \mathbf{x}_j \in G_x^J, j = 1, 2, \dots, J\}$, además, como los modelos propuestos están basados en grupos de parches similares es necesario dividir el conjunto G^J en K subconjuntos con pares de parches con estructura similar. Se define como $G_k^Q = \{(\mathbf{y}_q, \mathbf{x}_q) | \mathbf{y}_q \in$

$G_{y,k}^Q, \mathbf{x}_q \in G_{x,k}^Q, q = 1, 2, \dots, Q_k\}$ el conjunto de pares de parches indizados que fueron asignados al grupo k , $Q_k = |G_k^Q|$ como la cantidad de parches en el conjunto G_k^Q , $G_{y,k}^Q$ y $G_{x,k}^Q$ representan los conjuntos que contienen los parches de baja y alta resolución del k -ésimo grupo respectivamente. Como resultado del proceso de agrupamiento (explicado en la Sec. 3.3.3) se obtiene un conjunto de centroides $G^\mu = \{\boldsymbol{\mu}_k | k = 1, 2, \dots, K\}$ que son usados por ambos modelos propuestos.

3.3.1. Entrenamiento de las Redes Neuronales Artificiales

Dado el conjunto G^J se busca entrenar un conjunto de K redes neuronales que aprendan la transformación de parches de baja resolución a su versión en alta resolución. Como se menciona anteriormente, el conjunto G^J es dividido en K sub-conjuntos denominados G_k^Q . Es importante mencionar que antes de aplicar el agrupamiento es necesario transformar todos los parches en la base de datos de entrenamiento usando la siguiente definición:

$$\begin{aligned}\hat{\mathbf{y}}_j &= \mathbf{y}_j - \text{mean}(\mathbf{y}_j), j = 1, 2, \dots, J, \\ \hat{\mathbf{x}}_j &= \mathbf{x}_j - \text{mean}(\mathbf{y}_j), j = 1, 2, \dots, J,\end{aligned}\tag{3.12}$$

donde, $\text{mean}(\cdot)$ es una función que calcula el promedio de los elementos de un vector.

Centrar los valores de cada parche de baja resolución alrededor de 0 ayuda a que las redes neuronales artificiales aprendan de mejor manera la estructura de cada parche. En el caso de los parches \mathbf{x}_j se usa la media de su versión en baja resolución (en lugar de su propia media), debido a que si se usaría la transformación $\mathbf{x}_j - \text{mean}(\mathbf{x}_j)$ no sería posible recuperar la media del parche de alta resolución al evaluar una red neuronal artificial con algún parche de baja resolución en el proceso de reconstrucción. Por otro lado, no se sugiere normalizar usando la desviación estándar, ya que tampoco es posible estimar este valor y los resultados se ven afectados negativamente de forma significativa, ya que los detalles de alta frecuencia están relacionados fuertemente con la varianza de cada parche.

Debido a la función de activación usada en las redes neuronales artificiales, es necesario cambiar el rango dinámico de cada parche de $(-255, 255)$ a $(-1, 1)$:

$$\begin{aligned}\hat{\mathbf{y}}_q^* &= \frac{\hat{\mathbf{y}}_q}{z_k^y}, \quad \forall \hat{\mathbf{y}}_q \in G_{y,k}^Q, \\ \hat{\mathbf{x}}_q^* &= \frac{\hat{\mathbf{x}}_q}{z_k^x}, \quad \forall \hat{\mathbf{y}}_q \in G_{x,k}^Q,\end{aligned}\tag{3.13}$$

donde, z_k^y y z_k^x son los parámetros de normalización de cada grupo k . Definimos el conjunto de pares de factores de normalización como $G^z = \{(z_k^y, z_k^x) | k = 1, 2, \dots, K\}$. Estos términos representan los valores máximos de cada grupo, calculados como el máximo de los valores absolutos de todas las intensidades de cada parche en cada grupo.

Reescribiendo la Eq. (3.3) usando las transformaciones anteriores se obtiene:

$$\begin{aligned}\mathbf{x}_n &= f(\mathbf{y}_n | G^{\text{ANN}}) = \frac{1}{W_n} \sum_{k=1}^K w(\boldsymbol{\mu}_k, \mathbf{y}_n - \text{mean}(\mathbf{y}_n)) \\ &\quad (\text{ANN}_k(\frac{\mathbf{y}_n - \text{mean}(\mathbf{y}_n)}{z_k^y} | \Theta_k^{\text{ANN}}) z_k^x + \text{mean}(\mathbf{y}_n)), \quad \forall n \in N.\end{aligned}\tag{3.14}$$

Como punto importante a resaltar, el preproceso de agrupamiento sobre la base de datos de entrenamiento para el modelo basado en redes neuronales artificiales se realizó con los parches transformados usando las Eq. (3.12). Es por esta razón que es necesario restar la media al parche \mathbf{y}_n antes de evaluar la función $w(., .)$.

3.3.1.1. Topología de la Red Neuronal Artificial

Se puede observar en la Fig. 3.5 la estructura general que se está usando en las redes neuronales artificiales. Es una red neuronal artificial de tres capas. La capa de entrada recibe las intensidades de un parche de baja resolución $\mathbf{y} = (y_1, y_2, \dots, y_P)$, una capa oculta que de tamaño mayor a $\frac{2}{3}$ de las neuronas en la capa de entrada y menor o igual al doble que la entrada (ver Karsoliya (2012)), y la capa de salida genera un parche de alta resolución $\mathbf{x} = (x_1, x_2, \dots, x_P)$.

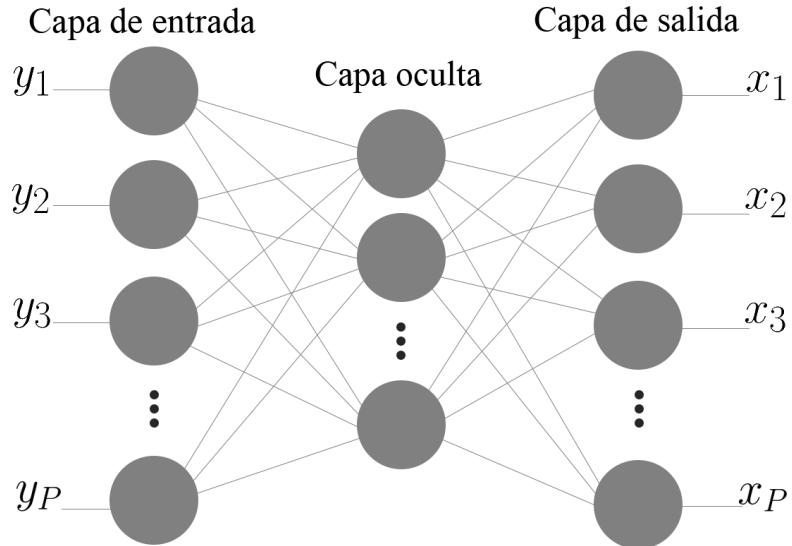


Figura 3.5: Estructura de una Red Neuronal Artificial.

Para la función de activación en las capas oculta y de salida se usa una función simétrica sigmoidea, el dominio de esta función es $[-1, 1]$, la definición es la siguiente:

$$w = \tanh(s * z) = \frac{2}{1 + \exp(-2 * s * z)} - 1, \quad (3.15)$$

donde, w es la salida, z es la entrada a la función de activación, s es el parámetro de steepness (positivo). En la práctica se encontró que un valor $s = 0.1$ funciona bien.

Recordar que el modelo basado en redes neuronales artificiales depende del conjunto G^{ANN} . Para cada elemento de este conjunto se aplicó la misma topología (aunque se podría probar diferentes configuraciones).

3.3.2. Entrenamiento de los Diccionarios

En una Sección anterior (Sec 3.2.2) se explicó como el modelo basado en diccionarios permite obtener un parche de alta resolución x a partir de un parche de baja resolución y . Para llevar a cabo la estimación se usa un conjunto de pares de diccionarios definido previamente como G^D . Estos diccionarios son entrenados de tal forma que cada par de diccionarios comparta la misma representación o código α para un parche de baja resolución y y su versión correspondiente en alta resolución x . En esta Sección se explica como llevar a cabo el entrenamiento de estos diccionarios para que cumplan esta característica.

3.3.2.1. Entrenamiento del Diccionario de Baja Resolución

Definimos el conjunto de coeficientes $G_k^\alpha = \{\boldsymbol{\alpha}_q | \boldsymbol{\alpha}_q \in \mathbb{R}^M, q = 1, 2, \dots, Q_k\}$ que representa al conjunto de parches de baja resolución respecto al diccionario \mathbf{D}_k^y para $\mathbf{y}_q \in G_{y,k}^Q$, donde $\mathbf{D}_k^y = (\mathbf{d}_{k,1}^y, \mathbf{d}_{k,2}^y, \dots, \mathbf{d}_{k,M}^y)$, $m = 1, 2, \dots, M$ para $\mathbf{d}_{k,m}^y$ los átomos del diccionario. Si \mathbf{D}_k^y es bien aprendido entonces existe un vector de coeficientes $\boldsymbol{\alpha}_q$ que cumple $\mathbf{x}_q = \mathbf{D}_k^y \boldsymbol{\alpha}_q$, para \mathbf{x}_q la versión en alta resolución del parche \mathbf{y}_q . Se define la matriz de coeficientes $\mathbf{A}_k = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_{Q_k})$ que representan a un conjunto de observaciones $\bar{\mathbf{Y}}_k = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{Q_k})$ sobre el diccionario \mathbf{D}_k^y , para Q_k la cantidad de observaciones o parches de baja resolución en el conjunto $G_{y,k}^Q$. El diccionario \mathbf{D}_k^y es estimado a partir de las observaciones $\bar{\mathbf{Y}}$ usando un algoritmo de optimización (descrito en esta sección), pero también es posible aproximar los átomos del diccionario mediante el uso de PCA, Wavelets, Transformada de Fourier, entre otras técnicas.

El problema de estimación de diccionarios es bien conocido en la literatura y varios métodos de Super Resolution, como los que se mencionan en el Capítulo 2 en la Sec. 2.3.1 hacen uso de modelos similares. Se usa el siguiente modelo (común en la literatura) en este trabajo de tesis:

$$\begin{aligned} \mathbf{D}_k^y, \mathbf{A}_k = \underset{\mathbf{D}_k^y, \mathbf{A}_k}{\operatorname{argmín}} \frac{1}{2} \sum_{q=1}^{Q_k} \|\mathbf{y}_q - \mathbf{D}_k^y \boldsymbol{\alpha}_q\|_2^2 + \lambda \|\boldsymbol{\alpha}_q\|_1, \quad k = 1, 2, \dots, K, \\ \text{s.t. } \|\mathbf{d}_{k,m}^y\|_2^2 \leq 1, \quad m = 1, 2, \dots, M. \end{aligned} \quad (3.16)$$

La norma l_1 sirve para forzar a que la solución obtenida sea sparse, la norma l_2 previene la obtención de valores muy grandes en el diccionario \mathbf{D}_k^y . Esta formulación Eq. (3.16) es no convexa en \mathbf{D}_k^y y \mathbf{A}_k , sin embargo, si \mathbf{D}_k^y o \mathbf{A}_k son dados, entonces es convexa en cualquiera de los casos. A continuación se describe el proceso de optimización para resolver Eq.(3.16):

1. Inicializar \mathbf{D}_k^y usando valores y vectores propios obtenidos del método SVD con $\bar{\mathbf{Y}}_k$. También se pueden seleccionar parches de la base de datos aleatoriamente. Usando SVD para estimar el diccionario inicial:

$$\begin{aligned} \mathbf{U}, \mathbf{S}, \mathbf{V}^T &= \text{SVD}(\bar{\mathbf{Y}}_k), \\ \mathbf{D}_k^y &= \mathbf{S}\mathbf{V}^T, \\ \mathbf{A}_k &= \mathbf{U}, \end{aligned} \tag{3.17}$$

como el procedimiento anterior no asegura que el diccionario \mathbf{D}_k^y tenga los M átomos o elementos deseados, se aplica la siguiente regla; Si la cantidad de elementos actual del diccionario es menor que M entonces se agregan los elementos faltantes con valor de cero, si por el contrario la cantidad de elementos actual es mayor que M entonces se eliminan los elementos sobrantes que estén más a la derecha del diccionario.

2. Una vez que \mathbf{D}_k^y es definido, el siguiente paso es actualizar la matriz de coeficientes o códigos \mathbf{A}_k . Se propone usar LARS (Efron et al., 2004) para computar la solución Lasso del siguiente problema de optimización:

$$\mathbf{A}_k = \underset{\mathbf{A}_k}{\operatorname{argmín}} \frac{1}{2} \sum_{q=1}^{Q_k} \|\mathbf{y}_q - \mathbf{D}_k^y \boldsymbol{\alpha}_q\|_2^2 + \lambda \|\boldsymbol{\alpha}_q\|_1, \quad k = 1, 2, \dots, K. \tag{3.18}$$

3. El siguiente paso es actualizar el diccionario \mathbf{D}_k^y , con \mathbf{A} fijo. Se usa una actualización de estrategia uno-por-uno para resolver el siguiente problema de optimización:

$$\begin{aligned} \mathbf{D}_k^y &= \underset{\mathbf{D}_k^y}{\operatorname{argmín}} \frac{1}{2} \sum_{q=1}^{Q_k} \|\mathbf{y}_q - \mathbf{D}_k^y \boldsymbol{\alpha}_q\|_2^2 + \lambda \|\boldsymbol{\alpha}_q\|_1, \quad k = 1, 2, \dots, K, \\ \text{s.t. } &\|\mathbf{d}_{k,m}^y\|_2^2 \leq 1, \quad m = 1, 2, \dots, M. \end{aligned} \tag{3.19}$$

4. Iterar alternativamente entre los pasos 2 y 3 hasta convergencia.

Este proceso se usa para obtener el diccionario con los parches de baja resolución. En la siguiente Sección se presenta la manera en que se estima el diccionario de alta resolución, de tal forma que los coeficientes $\boldsymbol{\alpha}$ entre ambos diccionarios sean los mismos.

3.3.2.2. Restricciones Sobre la Representación Entre Diccionarios

Dado el conjunto G_k^Q (con k un índice) formado por pares de parches de baja y alta resolución y el conjunto de diccionarios de baja resolución $G_y^D = \{\mathbf{D}_k^y | \mathbf{D}_k^y \in \mathbb{R}^{P \times M}, k = 1, 2, \dots, K\}$ entrenados previamente con los parches de baja resolución correspondientes $\mathbf{y}_q \in G_{y,k}^Q$, se busca aprender K diccionarios de alta resolución \mathbf{D}_k^x tal que se cumpla que $\bar{\mathbf{Y}}_k = \mathbf{D}_k^y \mathbf{A}_k$ y $\bar{\mathbf{X}}_k = \mathbf{D}_k^x \mathbf{A}_k$, donde, $\bar{\mathbf{Y}}_k = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{Q_k})$ y $\bar{\mathbf{X}}_k = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{Q_k})$ son matrices formadas por los parches de baja y alta resolución del conjunto G_k^Q . Definimos $G_x^D = (\mathbf{D}_k^x | \mathbf{D}_k^x \in \mathbb{R}^{P \times M}, k = 1, 2, \dots, K)$ como el conjunto de diccionarios de alta resolución.

En otros trabajos de Single Image Super Resolution (ver Capítulo 2) se proponen diferentes maneras de estimar los parches de alta resolución. Algunos modelos estiman los diccionarios de baja y alta resolución de forma conjunta en un sólo proceso de optimización. Otros, como en este trabajo, proponen modelos desacoplados donde se realiza un proceso de optimización para estimar primero el diccionario de baja resolución y luego estimar en otro proceso el diccionario de alta resolución. La norma l_1 es comúnmente usada para estimar ambos diccionarios. Sin embargo, en este trabajo de tesis se propone usar la norma de Frobenius $\|\cdot\|_F$ que aporta ciertas ventajas. Las ventajas de usar la norma $\|\cdot\|_F$ es que permite obtener una formula cerrada, además de que al estar entrenando diccionarios con parches con estructuras muy similares entre sí, no es necesario entrenar diccionarios sobre-completos y por lo tanto no se necesita una representación sparse. Esto ayuda a que puedan usarse diccionarios con pocos átomos, lo que reduce el tiempo computacional tanto en el proceso de entrenamiento como en el de reconstrucción. El modelo que se propone para estimar los diccionarios de alta resolución es el siguiente:

$$G_x^D = \underset{G_x^D}{\operatorname{argmín}} \sum_{k=1}^K \|\bar{\mathbf{X}}_k - \mathbf{A}_k^y \mathbf{D}_k^x\|_F^2 + \lambda \|\mathbf{D}_k^x\|_F^2, \quad (3.20)$$

donde, $\|\cdot\|_F$ la norma de Frobenius de una matriz, definida como $\sqrt{\operatorname{Tr}(\mathbf{B}\mathbf{B}^H)}$ donde Tr es la traza de una matriz y \mathbf{B}^H es la transpuesta conjugada.

En el trabajo de Cedeño et al. (2015) se hace referencia a la formula cerrada para el problema Eq. (3.20):

$$\mathbf{D}_k^x = (\mathbf{A}_k^{yT} \mathbf{A}_k^y + \lambda \mathbf{I})^{-1} \mathbf{A}_k^{yT} \bar{\mathbf{X}}_k, \quad (3.21)$$

donde, \mathbf{I} es la matriz identidad, λ representa un parámetro positivo de regularización que asegura que $\mathbf{A}_k^{yT} \boldsymbol{\alpha}_k^{yT} + \lambda \mathbf{I}$ es positiva definida y tiene inversa.

Con fines de visualización se presentan en la Fig. 3.6 ocho pares de diccionarios entrenados con lo explicado en esta Sección. Estos pares de diccionarios corresponden a los primeros ocho grupos cuyos centroides se muestran en Fig. 3.8. Se puede notar que algunos átomos comparten una estructura similar entre los diccionarios de baja resolución y sus correspondientes en alta. La diferencia es que los átomos del diccionario de alta resolución parecen tener más información o frecuencias altas que los átomos de los diccionarios de baja resolución.

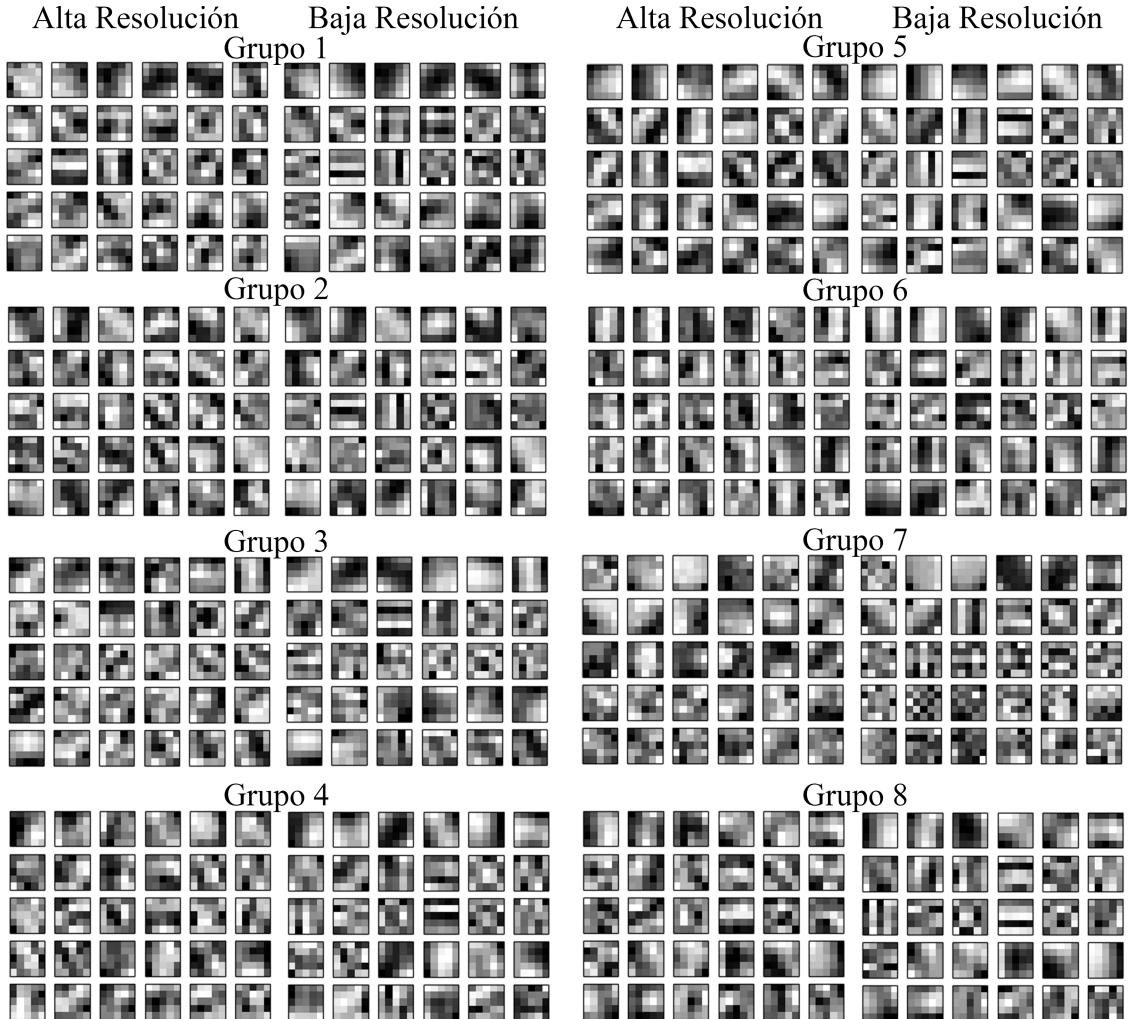


Figura 3.6: Diccionarios de Alta Resolución y sus Correspondientes en Baja Resolución. Cada par de diccionarios fue entrenado con un conjunto de parches de tamaño 5×5 . Cada diccionario tiene 30 átomos.

3.3.3. Usando Agrupamiento en los Datos de Entrenamiento

En las Secciones anteriores se explicó que para parches suficientemente pequeños existen formas como líneas curvas, líneas rectas, esquinas, etc., que aparecen a lo largo de las imágenes naturales (ver Fig. 3.1). Los modelos propuestos hacen uso de esta información de tal forma que se pueda mejorar la estimación de los parámetros. La mejora al estimar los diccionarios se debe a que en lugar de estimar sólo un par de diccionarios con un conjunto de datos complicado (compuesto de varias estructuras de parches) se estiman varios pares de diccionarios donde cada estimación es independiente entre sí y los datos usados para estimar cada par de diccionarios tienen estructuras similares, lo que ayudaría a que los diccionarios estén mejor estimados. En esta Sección se explica como se lleva a cabo el agrupamiento y se muestran algunos gráficos que soportan la idea de que usar agrupamien-

to en los modelos ayuda a mejorar los resultados obtenidos.

Dado el conjunto G^J de pares de parches de entrenamiento, la idea principal es que las redes neuronales o los diccionarios pueden representar mejor parches con estructuras similares. De esta manera se está aprendiendo la transformación entre baja y alta resolución de un elemento con una estructura definida. Algunos trabajos que apoyan esta idea son Wang et al. (2012) y Dong et al. (2011), sin embargo, en estos trabajos la idea del agrupamiento no se incorpora de manera directa en los modelos matemáticos propuestos.

En esta tesis se usa K -Means como técnica de agrupamiento. Es una técnica sencilla de implementar y en general los resultados que genera son adecuados para nuestros propósitos. Otra razón para usar esta técnica es que puede manejar un número de grupos y parches bastante grande. En las pruebas se usaron cerca de 1 millón de parches y al rededor de 2000 grupos. Una de las desventajas de esta técnica es que al aumentar el número de grupos y parches el costo computacional se incrementa considerablemente. Esto debe considerarse a la hora de llevar a cabo el agrupamiento en el proceso de entrenamiento.

El modelo a minimizar en K -Means es el siguiente:

$$\operatorname{argmín}_C \sum_{k=1}^K \sum_{\mathbf{y}_j \in C_k} \|\mathbf{y}_j - \boldsymbol{\mu}_k\|^2, \quad (3.22)$$

donde, $\{\mathbf{y}_j | \mathbf{y}_j \in \mathbb{R}^P, j = 1, 2, \dots, J\}$ es un conjunto de observaciones donde cada observación es un parche de baja resolución en una forma de vector. $\{C_k | k = 1, 2, \dots, K\}$ representa los diferentes grupos, $\boldsymbol{\mu}_k$ es la media de los de los parches en C_k .

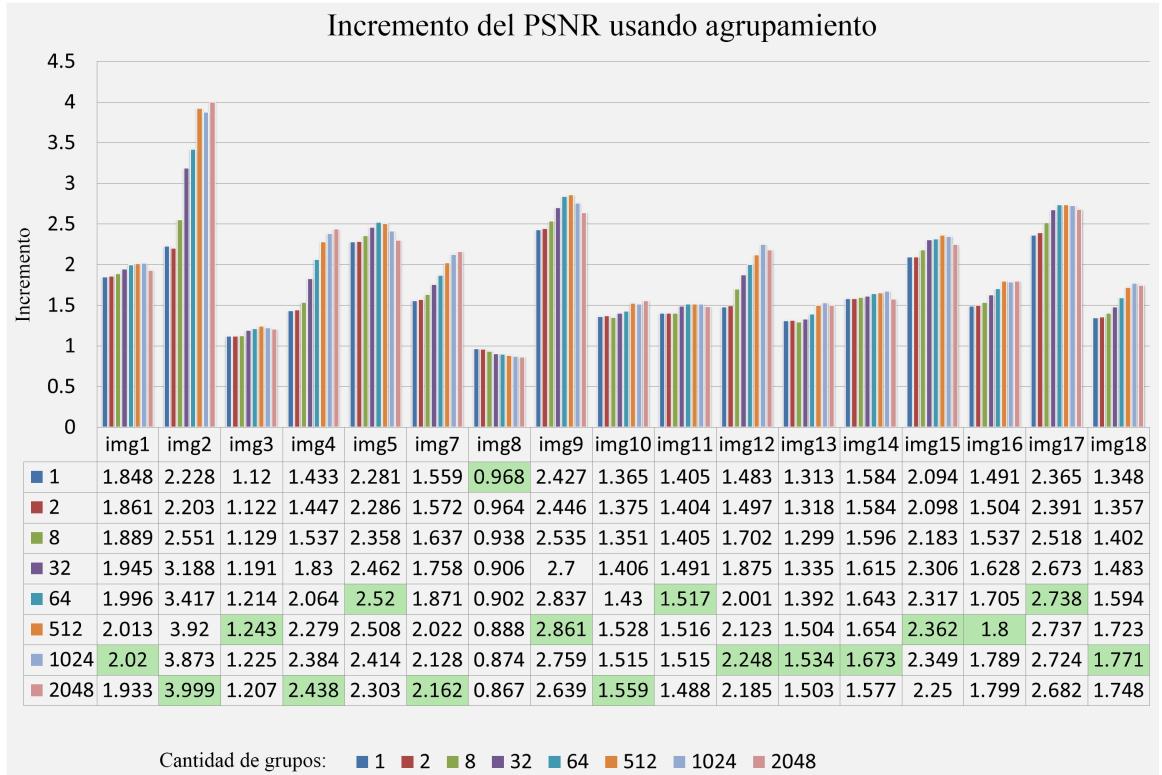


Figura 3.7: Incremento de PSNR vs Cantidad de grupos.

Gráfica y tabla de datos que muestran la diferencia entre el PSNR obtenido por interpolación bicubic y el PSNR obtenido con el modelo de diccionarios. La cantidad de pares de parches usada para esta prueba fue de 577200.

Para apoyar la idea de que el agrupamiento puede mejorar los resultados de los modelos propuestos, se entrenaron varios conjuntos de diccionarios con diferentes números de grupos 1, 2, 8, 32, 64, 256, 512, 1024, 2048, se aplicó el modelo propuesto basado en diccionarios y se estimó la imagen de alta resolución para diferentes imágenes. En la Fig. 3.7 se muestra una gráfica con diferentes valores, estos valores representan el PSNR de la estimación obtenida con nuestro modelo menos el PSNR obtenido por interpolación bicubica.

Note que en la Fig. 3.7 se ven las ventajas de usar agrupamiento. En casi toda las imágenes que se probaron se incrementa el PSNR a medida que las pruebas se acercan al conjunto de grupos de tamaño $K = 512$ y después comienza a decrecer. En el Capítulo 4 se muestra con más detalle la relación entre el número de parches en la base de datos de entrenamiento y el valor óptimo para K , en general, a mayor cantidad de parches en la base de datos un mayor número de grupos. También se muestra en el mismo Capítulo el comportamiento para el modelo de redes neuronales artificiales, el cual resulta similar.

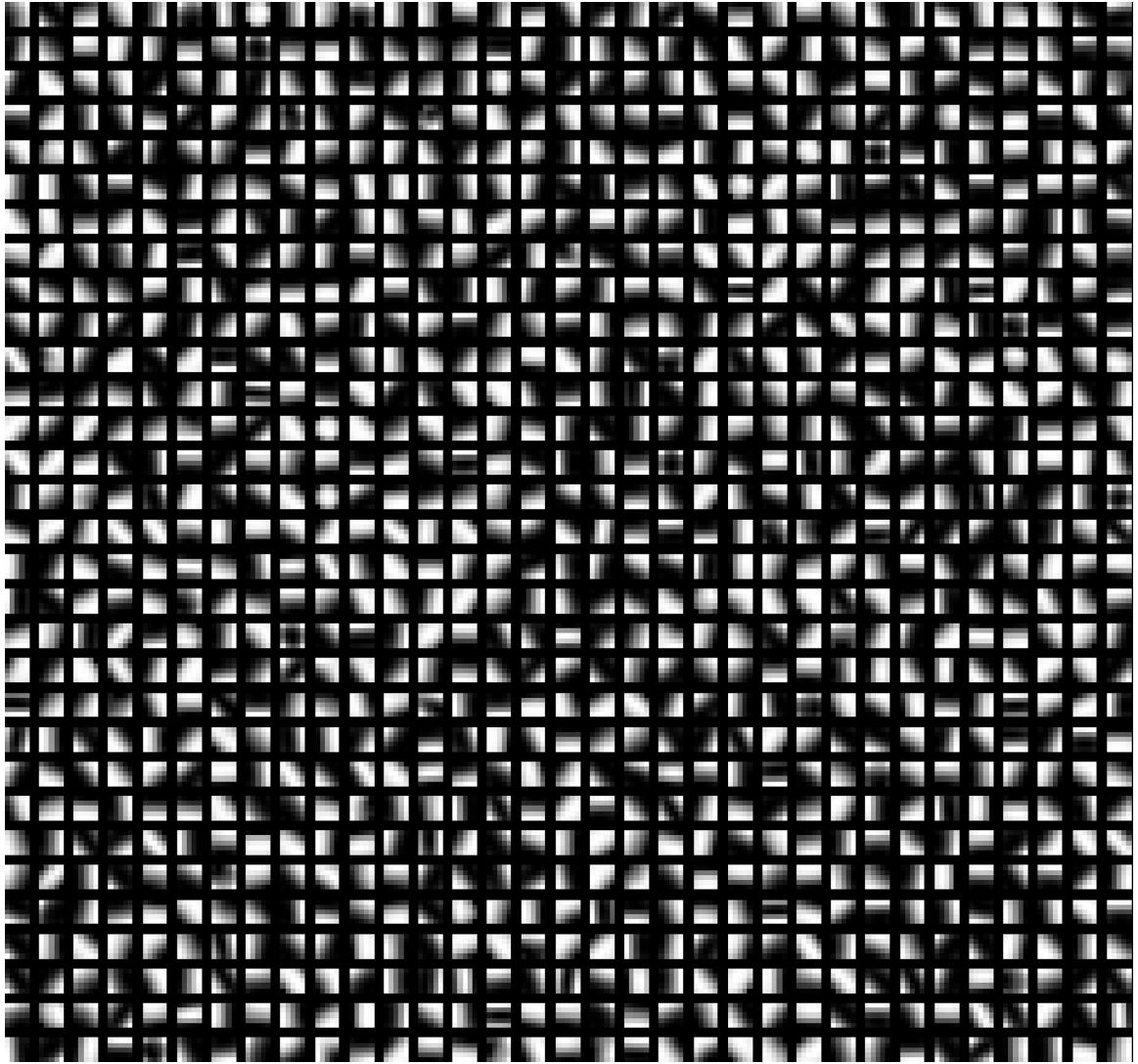


Figura 3.8: 1024 grupos obtenidos de un conjunto de 577700 parches.
Estos centroides son una buena aproximación de las diferentes estructuras que se pueden observar en parches pequeños como líneas rectas, líneas curvas, esquinas, círculos, etc.

Una vez que se tienen calculados los K grupos y sus respectivos centroides μ_k , el siguiente paso es optimizar los modelos. Es importante mencionar que los centroides μ_k son utilizados en el proceso de reconstrucción por lo que al implementar este trabajo es necesario guardar la información de estos además de los respectivos parámetros de los modelos.

En este Capítulo se ha hablado de que para parches suficientemente pequeños existen ciertas estructuras. Un experimento interesante para ver como son estas estructuras es mostrar los centroides μ_k en forma de imagen. Estos centroides representan una idea de las estructuras que pueden observarse comúnmente en los parches pequeños extraídos de imágenes naturales. En la Fig. 3.8 se muestra un conjunto de centroides.

En la Fig. 3.8 se pueden observar 1024 centroides calculados usando *K*-Means con una base de datos de 577700 pares de parches. Note que aparecen centroides similares, sin embargo, es importante mencionar que por motivos de visualización se cambió el rango dinámico de cada centroide para que fuera más fácil ver su estructura, por lo que centroides con estructura similar en realidad representan grupos de parches con diferentes rangos dinámicos en cuanto a intensidad de píxeles.

3.4. Reconstrucción de una Imagen de Alta Resolución

En esta Sección se explica el proceso de reconstrucción para obtener la versión de alta resolución dada una imagen de baja resolución. Se detalla como llevar a cabo la estimación de parches de alta resolución y el proceso de reconstrucción usando solapamiento de parches. También, se introduce una restricción más al problema de Single Image Super Resolution y se explica el proceso para mantener esta restricción en la solución generada por las propuestas presentadas en esta tesis.

3.4.1. Estimando los Parches de Alta Resolución

A continuación, se presentan los pasos a seguir para procesar una imagen de baja resolución \mathbf{I}_l con el objetivo de generar su versión en alta resolución. Primero, se convierte \mathbf{I}_l en un conjunto de parches de baja resolución $G_y^N = \{\mathbf{y}_n | \mathbf{y}_n \in \mathbb{R}^P, i = 1, 2, \dots, N\}$ con N el número total de píxeles en \mathbf{I}_l . Para poder obtener un parche por cada píxel es necesario definir las condiciones de frontera y se recomienda usar condiciones de frontera simétricas. Una vez que se tienen los parches de baja resolución junto con los centroides μ_k (obtenidos en el proceso de entrenamiento) se evalúa alguno de los modelos para obtener la versión en alta resolución. Los algoritmos Alg. 1 y Alg. 2 muestran el proceso de reconstrucción para el modelo basado en redes neuronales artificiales y para el modelo basado en diccionarios.

El problema con el algoritmo 2 es el costo computacional que se genera al evaluar al menos N veces (una por cada parche) un problema de optimización para obtener la solución al problema de LASSO. Para reducir el tiempo computacional se sugiere usar el caso especial donde en la Eq.(3.7) sólo se evalúa un par de diccionarios, es decir, el par de diccionarios que corresponde a la similaridad más alta entre el parche de baja resolución y un cierto centroide μ_k . De esta forma sólo se resuelven exactamente N problemas de optimización

Algorithm 1: Estimación de parches de alta resolución usando el modelo basado en redes neuronales artificiales.

Result: Una matriz con los parches de alta resolución.

Input : Una imagen de baja resolución $\mathbf{I}_l \in \mathbb{R}^{R \times C}$.

Un conjunto de redes neuronales artificiales G^{ANN} .

Un conjunto de centroides G^μ .

Un conjunto de factores de normalización G^z (ver Sec.3.3.1).

Output: $\hat{\mathbf{X}} = (\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots, \hat{\mathbf{x}}_N)$.

1 Define $\mathbf{Y} = (\mathbf{I}_l \uparrow_s)$.

2 Define $G^y = \{\mathbf{y}_n | n = 1, 2, \dots, N\}$ como el conjunto de los parches de baja resolución en \mathbf{Y} .

3 **foreach** y_n in G^y **do**

4 | Evaluar la Eq.(3.14) con y_n para obtener $\hat{\mathbf{x}}_n$.

5 | Agrega $\hat{\mathbf{x}}_n$ a una nueva fila en la matriz $\hat{\mathbf{X}}$.

6 **end**

7 regresa $\hat{\mathbf{X}}$.

Algorithm 2: Estimación de parches de alta resolución usando el modelo basado en diccionarios.

Result: Una matriz con los parches de alta resolución.

Input : Una imagen de baja resolución $\mathbf{I}_l \in \mathbb{R}^{R \times C}$.

Un conjunto de diccionarios G^D .

Un conjunto de centroides G^μ .

Output: $\hat{\mathbf{X}} = (\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots, \hat{\mathbf{x}}_N)$.

1 Define $\mathbf{Y} = (\mathbf{I}_l) \uparrow_s$.

2 Define $G^y = \{\mathbf{y}_n | n = 1, 2, \dots, N\}$ como el conjunto de los parches de baja resolución en \mathbf{Y} .

3 **foreach** y_n in G^y **do**

4 | Evaluar Eq.(3.7) para obtener $\hat{\mathbf{x}}_n$.

5 | Agregar $\hat{\mathbf{x}}_n$ como una nueva fila a la matriz $\hat{\mathbf{X}}$.

6 **end**

7 regresar $\hat{\mathbf{X}}$.

y los resultados no se ven afectados negativamente de manera significativa.

3.4.2. Reconstrucción de la Imagen de Alta Resolución

En la Sección 3.4 se presentaron 2 algoritmos (Alg. 1 y Alg. 2) que reciben una imagen de baja resolución $\mathbf{I}_l \in \mathbb{R}^{R \times C}$, extraen los $N = RC$ parches de baja resolución $\mathbf{y}_n, n = 1, 2, \dots, N$ de la imagen de entrada y generan una matriz $\hat{\mathbf{X}} = (\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots, \hat{\mathbf{x}}_N)$ con la versión de alta resolución correspondiente a cada parche de baja resolución. En esta Sección se explica el proceso que se lleva a cabo para reconstruir una imagen de alta resolución \mathbf{I}_h usando $\hat{\mathbf{X}}$. Notar que el tamaño de parche debe ser seleccionado en el proceso de entrenamiento y esto no es un parámetro variable en el proceso de reconstrucción. Definimos la cantidad de elementos en un parche como $P = (2a + 1)^2$. El Algoritmo 3 describe el proceso de reconstrucción. Es importante resaltar que hay que definir las condiciones de frontera para el algoritmo de reconstrucción.

3.4.3. Restricción Global de Reconstrucción

Hasta este punto se ha explicado ya el proceso de estimación de los parches de alta resolución y como reconstruir una imagen de alta resolución \mathbf{I}_h . Sin embargo, esta aproximación a la solución real no cumple con el modelo de adquisición o degradación de la imagen, de tal manera que si se genera una imagen de baja resolución usando $(\mathbf{I}_h * H_\sigma) \downarrow_s$ suponiendo que se conocen los buenos parámetros y asumiendo ausencia de ruido, no se será posible obtener la misma imagen que tuvo como entrada el algoritmo, es decir, $(\mathbf{I}_h * H_\sigma) \downarrow_s \neq \mathbf{I}_l$. Es por esta razón, con base en la Eq.(2.3) que se propone hacer uso de la **Restricción Global de Reconstrucción**, para mantener una coherencia entre la solución generada y la imagen de baja resolución.

Una manera de mantener esta restricción es proyectando la imagen estimada de alta resolución \mathbf{I}_h sobre el espacio de la imagen de baja resolución $(\mathbf{I}_h * H_\sigma) \downarrow_s$:

$$\mathbf{I}_h^* = \underset{\mathbf{X}}{\operatorname{argmín}} \|\mathbf{X} - \mathbf{I}_h\| \quad \text{s.t. } (\mathbf{X} * H_\sigma) \downarrow_s = \mathbf{I}_l, \quad (3.23)$$

donde, H_σ representa un kernel gaussiano con el valor σ usado en modelo de degradación que generó la imagen \mathbf{I}_l .

Algorithm 3: Reconstrucción de la imagen de alta resolución.

Result: Una imagen de alta resolución.

Input : Mismos parámetros de entrada que en los algoritmos Alg. 1 y Alg. 2 según sea el caso.

```
1 Define la matriz  $\mathbf{I}_h^0 = \mathbf{0}$ 
2 Define la matriz  $\mathbf{Z}^0 = \mathbf{0}$ 
3 Define  $t = 0$ 
4 Define  $g = 2a + 1$ 
5 Evaluar Alg. 1 ó Alg. 2 para obtener  $\hat{\mathbf{X}} = (\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots, \hat{\mathbf{x}}_N)$ 
6 foreach  $i = 0, 1, \dots, R$  do
7   foreach  $j = 0, 1, \dots, C$  do
8     /* Índice de la fila en  $\hat{\mathbf{X}}$  que corresponde al parche en la coordenada  $(i, j)$ . */
9      $n^* = i * R + j$ 
10    /* Extraer el parche en el índice  $n^*$ . */
11     $\hat{\mathbf{x}}_n^* = \hat{\mathbf{X}}(n^*)$ 
12    foreach  $r = -a, \dots, 0, \dots, a$  do
13      foreach  $c = -a, \dots, 0, \dots, a$  do
14         $\mathbf{I}_h^{t+1}(i+r, j+c) = \mathbf{I}_h^t(i+r, j+c) + \hat{\mathbf{x}}_n^*((a+r)*g + (a+c))$ 
15         $\mathbf{Z}^{t+1}(i+r, j+c) = \mathbf{Z}^t(i+r, j+c) + 1$ 
16      end
17    end
18  end
19  /* Normalizar la matriz  $\mathbf{I}_h^t$  */
20  foreach  $i = 0, 1, \dots, R$  do
21    foreach  $j = 0, 1, \dots, C$  do
22       $\mathbf{I}_h^{t+1}(i+r, j+c) = \mathbf{I}_h^t(i+r, j+c) / \mathbf{Z}^t(i+r, j+c)$ 
23       $t \leftarrow t + 1$ 
24  end
25 regresar  $\mathbf{I}_h^t$ .
```

La solución a este problema de optimización Eq.(3.23) puede ser obtenida mediante la aplicación del algoritmo de Back-Projection propuesto por Irani and Peleg (1991). La formulación iterativa de este método es la siguiente:

$$\mathbf{I}_h^{t+1} = \mathbf{I}_h^t + (\mathbf{I}_l - (\mathbf{I}_h^t * H_\sigma) \downarrow_s) \uparrow_s * B, \quad (3.24)$$

donde, \mathbf{I}_h^t es la imagen de alta resolución después de t iteraciones, B es un filtro de Back-Projection, H_σ es un kernel gaussiano, $*$ es el operador de convolución, \downarrow_s y \uparrow_s son los operadores de downsampling y upsampling respectivamente con un factor de escala s . El algoritmo 4 muestra este proceso.

Algorithm 4: Algoritmo de Back-Projection

Input : Una imagen de baja resolución $I_l \in \mathbb{R}^{R \times C}$.
 Una imagen estimada de alta resolución $\mathbf{I}_h \in \mathbb{R}^{sR \times sC}$.
 Un factor de escala s .
 Un valor σ para el kernel gaussiano.
 Un kernel de Back-Projection B .
 Número de iteraciones $iter$.
 Una tolerancia tol .

Output: Una imagen de alta resolución \mathbf{I}_h^* .

```

1 Define  $t \leftarrow 0$ 
2 Define  $H_\sigma$  como un kernel gaussiano con valor  $\sigma$ 
3 Define  $\mathbf{X}^t = \mathbf{I}_h$ 
4 foreach  $i$  in  $iter$  do
5   Resolver  $\hat{\mathbf{I}}_l = (\mathbf{X}^t * H_\sigma) \downarrow_s$ 
6   Resolver  $\mathbf{e} = \mathbf{I}_l - \hat{\mathbf{I}}_l$ 
7   Resolver error =  $RMSE(\mathbf{I}_l, \hat{\mathbf{I}}_l)$ 
8   if  $error < tol$  then
9     | break
10   end
11   Resolver  $\mathbf{X}^{t+1} = \mathbf{X}^t + (\mathbf{e} \uparrow_s) * B$ 
12   Define  $t \leftarrow t + 1$ 
13 end
14  $\mathbf{I}_h^* = \mathbf{X}^t$ 
15 regresa  $\mathbf{I}_h^*$ 
```

En el algoritmo 4: la función RMSE se define como $RMSE(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2}$.

En cuanto al kernel de Back-Projection B se puede usar un kernel gaussiano que permita distribuir el error en el espacio de mayor resolución. En nuestro trabajo se está usando este kernel. Sin embargo, otros kernel pueden ser definidos y en general se ve como trabajo a futuro la búsqueda de un mejor kernel que permita distribuir el error de mejor manera.

3.4.4. Resumen del Proceso de Reconstrucción

En esta Sección se presentan en detalle los 2 algoritmos enfocados al proceso de reconstrucción. Una vez que se han estimado los parámetros de la función $f(.|\Theta)$ basta con seguir los pasos descritos en la Fig.3.3 para obtener una estimación de alta resolución de una imagen de baja resolución \mathbf{I}_l . Los algoritmos Alg. 5 y Alg. 6 muestran a detalle estos pasos.

Algorithm 5: Proceso de Reconstrucción Basado en Redes Neuronales Artificiales

Result: Una imagen de alta resolución.

Input : Una imagen de baja resolución \mathbf{I}_l .

Un conjunto de redes neuronales artificiales G^{ANN} .

Un conjunto de centroides G^μ .

Un conjunto de factores de normalización G^z .

Un factor de escala s , debe ser el mismo factor de escala usado en el proceso de entrenamiento.

Un kernel gaussiano H_σ .

Un kernel de Back-Projection B .

Un valor de normalización h para la función de pesos $w(., .)$.

Output: Una imagen de alta resolución \mathbf{I}_h^*

```

/* Usar interpolación bicubic */
```

- 1 Evaluar $\mathbf{Y} = \mathbf{I}_l \uparrow_s$

```

/* Estimar los parches de alta resolución
correspondientes a los parches de baja resolución en
 $\mathbf{Y}$  */
```
- 2 Evaluar algoritmo 1 para obtener la matriz $\hat{\mathbf{X}}$ con los parches de alta resolución estimados
- 3 Evaluar algoritmo 3 con $\hat{\mathbf{X}}$, para obtener la imagen \mathbf{I}_h de alta resolución
- 4 Usar el algoritmo 4 con \mathbf{I}_h , para mantener la restricción global de reconstrucción
- 5 return \mathbf{I}_h^*

Los algoritmos 5 y 6 son similares, en general varios métodos propuestos para resolver el problema de Single Image Super Resolution están basados en este esquema. Como

Algorithm 6: Proceso de Reconstrucción Basado en Pares de Diccionarios

Result: Una imagen de alta resolución.

Input : Una imagen de baja resolución I_l .

Un conjunto de pares de diccionarios G^D .

Un conjunto de centroides G^μ .

Un conjunto de factores de normalización G^z .

Un factor de escala s , debe ser el mismo factor de escala usado en el proceso de entrenamiento.

Un kernel gaussiano H_σ .

Un kernel de Back-Projection B .

Un valor de normalización h para la función de pesos $w(., .)$.

Output: Una imagen de alta resolución I_h^*

```
/* Usar interpolación bicubic */  
1 Evaluar  $\mathbf{Y} = I_l \uparrow_s$   
    /* Estimar los parches de alta resolución  
       correspondientes a los parches de baja resolución en  
        $\mathbf{Y}$  */  
2 Evaluar algoritmo 2 para obtener la matriz  $\hat{\mathbf{X}}$  con los parches de alta resolución  
   estimados  
3 Evaluar algoritmo 3 con  $\hat{\mathbf{X}}$ , para obtener la imagen  $I_h$  de alta resolución  
4 Usar el algoritmo 4 con  $I_h$ , para mantener la restricción global de reconstrucción  
5 return  $I_h^*$ 
```

se mencionó previamente el problema más difícil es proponer un buen modelo y estimar sus parámetros. Si esta propuesta es buena la calidad del algoritmo se verá afectada positivamente. En el siguiente Capítulo se muestran algunos experimentos y resultados con los cuales se comparan los modelos propuestos en este trabajo de tesis con algunas otras propuestas que se han presentando en años recientes.

3.5. Factores de Escala

Existen 2 posibles opciones para generar imágenes con diferentes factores de escala. Por ejemplo, si se define $s = 2$ en el paso de entrenamiento entonces se podrá repetir el proceso de reconstrucción dos o más veces para obtener factores de escala más grandes. Sin embargo, factores de escala mayores a 4 comenzarán a alejarse demasiado de la solución real debido a los errores de estimación. Esto en general pasa con todos los algoritmos del estado del arte. La segunda opción para obtener factores de escala diferentes es, entrenar diferentes conjuntos de diccionarios con los factores de escala deseados.

En general las 2 opciones anteriores generan resultados similares. La ventaja de usar

la primer opción es que el algoritmo de Back-Projection parece tener mejores resultados en factores de escala pequeños ($s = 2$ por ejemplo). Sin embargo, entrenar diccionarios o redes neuronales directamente con el factor de escala deseado puede resultar mejor si no se aplica el algoritmo de Back-Projection o se aplica con pocas iteraciones. Esto dependerá de varios factores como las bases que aprendieron los diccionarios o los pesos de las redes neuronales artificiales. Parte de lo que se explica aquí fue resultado de la experiencia al desarrollar este trabajo de tesis y falta realizar una investigación mas a fondo sobre esto para poder dar conclusiones más certeras.

Capítulo 4

Experimentos y Resultados

En este Capítulo se evalúa el desempeño de los métodos propuestos en este trabajo contra algunos algoritmos del estado del arte. Haciendo uso de las métricas PSNR y SSIM se realiza un análisis comparativo evaluando la calidad de los resultados generados por cada algoritmo. Otro objetivo de este Capítulo es mostrar el efecto que tienen los distintos parámetros involucrados con los métodos propuestos en esta tesis y dar una idea del porque se seleccionaron ciertos valores en los experimentos presentados en este Capítulo.

La estructura de este Capítulo está definida de la siguiente forma; En la Sección **Construcción de la Base de Datos de Entrenamiento** (Sec. 4.2) se explican los pasos para construir una base de datos similar a las usadas en los experimentos de este trabajo. En la Sección **Número Óptimo de Grupos** (Sec. 4.4) se realizó un estudio que ayuda a explicar la relación entre la cantidad de pares de parches en la base de datos de entrenamiento y el número de grupos K usado en el agrupamiento. En la Sección **Estimación de los Parámetros para el Modelo Basado en Redes Neuronales Artificiales** (Sec. 4.5) se realizaron algunos experimentos para estudiar el efecto en términos de PSNR que tiene incluir una mayor cantidad de capas ocultas en la topología de la red neuronal artificial. En la Sección **Estimación de los Parámetros para el Modelo Basado en Diccionarios** (Sec. 4.6) se experimentó con diversos tamaños de diccionarios con el objetivo de determinar si algún tamaño ayuda a generar mejores resultados, también se realizaron experimentos con el objetivo de estudiar el efecto del factor λ (ver Eq. (3.11)) en los resultados obtenidos. En la Sección **Estimación del Parámetro de la Función de Pesos** (Sec. 4.7) se estudió como afecta el parámetro h de la Eq. 3.5 a los resultados obtenidos en términos de PSNR. Finalmente, en la Sección **Comparación con Otros Algoritmos** (Sec. 4.8) se muestran resultados y comparaciones con distintos algoritmos del estado del arte, haciendo uso de las métricas PSNR y SSIM se compara mediante imágenes y tablas la calidad de cada uno de los algoritmos propuestos contra los del estado del arte.

4.1. Métricas Usadas en los Experimentos

Las siguiente lista muestra las métricas usadas para evaluar la similitud entre un par de imágenes y son usadas a lo largo de esta Sección.

- **MSE**.- Mean Square Error, se define como:

$$\text{MSE}(\mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^L (\mathbf{x}_i - \mathbf{y}_i)^2 \quad (4.1)$$

- **RMSE**.- Root Mean Square Error, se define como:

$$\text{RMSE}(\mathbf{x}, \mathbf{y}) = \sqrt{\text{MSE}(\mathbf{x}, \mathbf{y})} \quad (4.2)$$

- **PSNR**.- Peak Signal to Noise Ratio, se define como:

$$\text{PSNR}(\mathbf{x}, \mathbf{y}) = 10 \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}(\mathbf{x}, \mathbf{y})} \right), \quad (4.3)$$

donde, MAX representa el valor máximo de intensidad que puede tomar la imagen.

- **SSIM**.- Structural Similarity, se define como:

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (4.4)$$

donde, μ_x es el promedio de \mathbf{x} , μ_y es el promedio de \mathbf{y} , σ_x^2 es la varianza de \mathbf{x} , σ_y^2 es la varianza de \mathbf{y} , $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ son dos variables que estabilizan la división, L es el rango dinámico de los valores de los píxeles, $k_1 = 0.01$ y $k_2 = 0.03$ por defecto.

4.2. Construcción de la Base de Datos de Entrenamiento

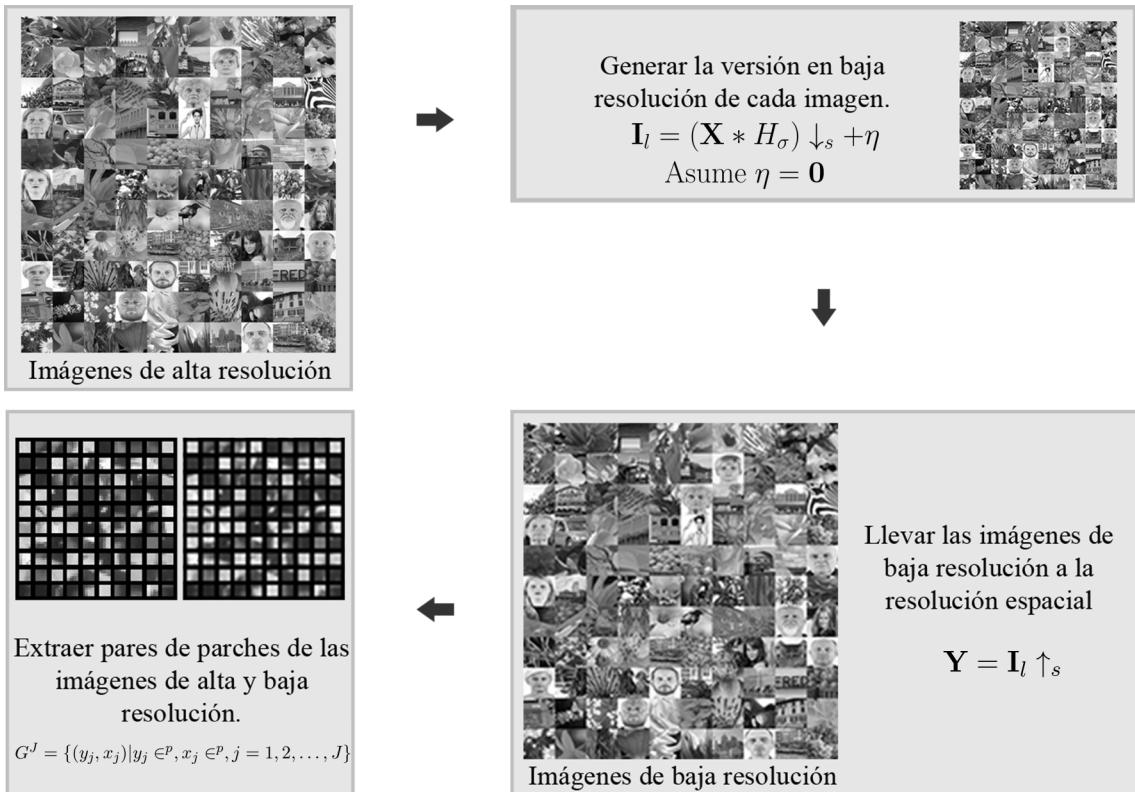


Figura 4.1: Diagrama de creación de la base de entrenamiento.

En esta sección se explica como construir una base de datos de entrenamiento, similar a la usada en los experimentos presentados en este trabajo de tesis, para el aprendizaje de los parámetros de los modelos propuestos en la Sec 3.2.

En la Fig. 4.1 se muestran los 3 pasos definidos para crear una base de datos de entrenamiento dado un conjunto de imágenes de alta resolución. El primer paso es crear una imagen de baja resolución para cada imagen de alta resolución usando el modelo de degradación Eq. (2.3), los valores para σ del kernel gaussiano y el factor de escala s dependen del problema que se quiera resolver y una vez definidos para la creación de la base de datos ya no es posible cambiarlos en la etapa de reconstrucción. Por otro lado, se asume un modelo sin ruido, es decir, $\eta = 0$. El segundo paso es re-escalar cada imagen de baja resolución usando algún método de interpolación (en nuestro caso bicubic), de forma que, cada imagen vuelva a tener la resolución espacial original. El tercer paso consiste en extraer un conjunto de pares de parches correspondientes (de cierto tamaño) de baja y alta resolución usando los conjuntos de imágenes mencionados en el paso anterior (las imágenes de alta resolución y las imágenes re-escaladas). Para cada par de imágenes correspondientes de

alta y baja resolución se extraen de manera aleatoria c pares de parches y definimos el total de pares de parches extraídos como J . Finalmente, este conjunto de pares de parches es a lo que se denomina base de datos de entrenamiento a lo largo de este documento de tesis.

El tamaño de parche puede considerarse un parámetro más que debe ser definido al momento de crear la base de datos de entrenamiento. Para definir este parámetro se realizaron una serie de experimentos para estudiar el efecto que tiene el tamaño de parche en los resultados obtenidos (ver Sec. 4.3). Por notación se define el tamaño de parche en forma de matriz como $(2a + 1) \times (2a + 1)$ donde a representa el “radio” del parche y el tamaño en forma vectorizada queda representado por la letra P de manera que $P = (2a+1)^2$.

4.3. Tamaño de Parche

En este Sección se muestra el resultado de algunos experimentos realizados con el propósito de estudiar el efecto que tiene el tamaño de parche en los resultados obtenidos en términos de PSNR y SSIM.

Para los experimentos mostrados en esta Sección se construyeron 5 bases de datos con las siguientes características; Los parámetros para el modelo de degradación (o adquisición) se definieron como $s = 2$ y $\sigma = 0.8$, la cantidad de pares de parches en cada base de datos fue de 65 mil pares de parches aproximadamente y la cantidad de grupos en que se dividió cada base de datos fue de 64 grupos. Cada una de las bases de datos fue construida con un tamaño de parche específico y los tamaños de parche usados en la creación de cada base de datos fueron $a = 1, a = 2, a = 3, a = 4$ y $a = 5$ de tal forma que una base de datos se construyó con únicamente con parches usando $a = 1$, otra base de datos usando únicamente $a = 2$ y así sucesivamente hasta tener las 5 bases de datos.

Los parámetros para el algoritmo de Back-Projection fueron; un kernel gaussiano con $\sigma = 0.8$ usado como el kernel de back-projection B , la cantidad de iteraciones de 750 y una tolerancia de error de $\text{RMSE} < 0.0001$.

Los experimentos realizados consistieron en usar cada una de las bases de datos creadas de manera individual para aprender los parámetros del modelo basado en diccionarios y los parámetros del modelo basado en redes neuronales artificiales y con cada uno de estos parámetros se evaluó el modelo correspondiente para obtener el PSNR y SSIM de un conjunto de imágenes de naturales.

En la Tabla 4.3 se puede observar el incremento en el PSNR obtenido con el modelo basado en diccionarios respecto al PSNR obtenido por interpolación bicubic, esto para cada tamaño de parche. Aunque no parece existir un tamaño de parche que sea totalmente mejor que otro, se puede observar que para los valores de $a \in [2, 3]$ se obtienen los mejores resultados. Sin embargo, se puede notar que algunos tamaños de parche funcionaron mejor en ciertas imágenes por lo que como trabajo futuro sería interesante experimentar si existen cierto tipo de estructuras para las cuales un tamaño de parche sea mejor. En la Tabla 4.2 se puede observar el resultado en términos de SSIM y aunque no hay gran diferencia entre los resultados se puede notar que con el tamaño de parche $a = 2$ en general se obtienen los mejores resultados.

Como conclusión sobre el tamaño de parche cuando se hace uso del modelo basado en diccionarios, es recomendable usar $a = 2$, esto porque al ser los parches más pequeños (25 píxeles por parche) se necesita menos memoria durante el entrenamiento y reconstrucción, además que, el tamaño de parche repercute directamente en el número de operaciones que se realizan durante el proceso de reconstrucción por lo que un tamaño de parche mas pequeño representa una menor cantidad de operaciones maquina y esto a gran escala se refleja en tiempo computacional.

En la Tabla 4.3 se puede observar el incremento en el PSNR obtenido con el modelo basado en redes neuronales artificiales respecto al PSNR obtenido por interpolación bicubic, esto para cada tamaño de parche. Se puede observar que pasa algo similar que con el modelo basado en diccionarios. El tamaño de parche en el rango $a \in [2, 3]$ parece dar mejores resultados, incluso parece que $a = 2$ es mejor. Sin embargo, en el promedio se puede observar que ambos tamaños de parche generan resultados muy cercanos entre sí. En términos de SSIM (Tabla 4.4) los resultados, de igual manera, parecen favorecer a parches cuyo tamaño esta entre $a \in [2, 3]$.

De manera similar al modelo basado en diccionarios es recomendable usar un tamaño de parche tal que $a = 2$, ya que como se explica anteriormente esto puede ayudar a que la estimación de una imagen de alta resolución sea más eficiente. Al incrementar el valor de a se espera (de manera intuitiva) que los resultados no sean mejores, esto debido a que la redundancia a nivel de parches va disminuyendo con tamaños de parche más grande. Al tener tamaños de parche más grandes la variabilidad de los parches en cada uno de los grupos en que se divide la base de entrenamiento es más grande y el proceso de optimización para estimar los valores de las redes neuronales artificiales o diccionarios se complica.

| IMG/ Tamaño Parche | $a = 5$ | $a = 4$ | $a = 3$ | $a = 2$ | $a = 1$ |
|--------------------|-------------|-------------|-------------|-------------|-------------|
| microchip | 3.28 | 3.78 | 3.88 | 4.05 | 3.68 |
| ocean | 0.81 | 0.86 | 0.95 | 0.95 | 0.96 |
| plane | 2.72 | 2.8 | 2.87 | 2.96 | 2.99 |
| woman_2 | 1.83 | 1.86 | 1.78 | 1.7 | 1.46 |
| sailing3 | 1.67 | 1.78 | 1.74 | 1.75 | 1.79 |
| sailing2 | 1.79 | 1.83 | 1.84 | 1.93 | 1.83 |
| carnivaldolls | 2.52 | 2.5 | 2.19 | 2.35 | 0.96 |
| cemetery | 2.11 | 2.19 | 2.18 | 2.09 | 1.78 |
| plant | 1.08 | 1.07 | 1.04 | 1.01 | 0.97 |
| squirrel | 1.22 | 1.24 | 1.27 | 1.32 | 1.27 |
| churchandcapitol | 1.66 | 1.7 | 1.75 | 1.8 | 1.42 |
| statue | 1.55 | 1.51 | 1.57 | 1.56 | 1.55 |
| stream | 1.18 | 1.18 | 1.2 | 1.22 | 1.16 |
| bikes | 2.08 | 2.2 | 2.3 | 2.38 | 2.34 |
| coinsinfoountain | 2.0 | 1.99 | 2.01 | 1.89 | 1.63 |
| girl_1 | 2.42 | 2.51 | 2.45 | 2.4 | 2.0 |
| text | 6.26 | 6.31 | 6.31 | 6.23 | 6.03 |
| woman_1 | 3.08 | 3.17 | 3.18 | 3.02 | 2.59 |
| man_3 | 2.67 | 2.76 | 2.72 | 2.57 | 2.22 |
| sailing4 | 1.24 | 1.29 | 1.38 | 1.43 | 1.46 |
| lena | 1.86 | 1.96 | 2.11 | 2.17 | 2.3 |
| oldMan_1 | 1.91 | 1.93 | 1.97 | 1.97 | 1.88 |
| oldWoman_1 | 1.8 | 1.85 | 1.99 | 2.09 | 2.04 |
| parrots | 2.28 | 2.44 | 2.48 | 2.44 | 2.28 |
| woman | 1.31 | 1.33 | 1.36 | 1.36 | 1.31 |
| sailing1 | 1.11 | 1.18 | 1.22 | 1.32 | 1.23 |
| lighthouse2 | 1.39 | 1.41 | 1.46 | 1.52 | 1.47 |
| face | 1.0 | 0.99 | 1.0 | 0.95 | 0.89 |
| flowersonih35 | 1.5 | 1.49 | 1.5 | 1.53 | 1.11 |
| caps | 1.44 | 1.48 | 1.54 | 1.58 | 1.58 |
| man_1 | 2.27 | 2.33 | 2.32 | 2.26 | 1.86 |
| womanhat | 1.57 | 1.68 | 1.72 | 1.7 | 1.66 |
| manfishing | 1.79 | 1.86 | 1.87 | 1.81 | 1.18 |
| man_2 | 2.01 | 1.99 | 1.97 | 1.87 | 1.67 |
| lighthouse | 1.38 | 1.42 | 1.41 | 1.52 | 1.56 |
| monarch | 2.93 | 3.29 | 3.48 | 3.69 | 3.7 |
| dancers | 1.58 | 1.64 | 1.64 | 1.69 | 1.43 |
| flowers | 1.6 | 1.64 | 1.69 | 1.64 | 1.54 |
| house | 1.4 | 1.39 | 1.41 | 1.39 | 1.29 |
| redHair | 1.53 | 1.52 | 1.52 | 1.52 | 1.43 |
| man_4 | 1.75 | 1.76 | 1.79 | 1.82 | 1.61 |
| building2 | 1.77 | 1.82 | 1.86 | 1.82 | 1.65 |
| promedio | 1.91 | 2.00 | 2.01 | 1.99 | 1.83 |

Máximo Mínimo

Tabla 4.1: Incremento en el PSNR obtenido con el modelo basado en diccionarios respecto al PSNR obtenido por interpolación bicubica. Pruebas realizadas con distintos tamaños de parche $(2a + 1) \times (2a + 1)$.

Cada fila (con excepción de la última) representa los resultados para una imagen distinta.

| IMG/ Tamaño Parche | $a = 5$ | $a = 4$ | $a = 3$ | $a = 2$ | $a = 1$ |
|--------------------|---------------|---------------|---------------|---------------|---------------|
| microchip | 0.9743 | 0.9763 | 0.9782 | 0.9786 | 0.9786 |
| ocean | 0.876 | 0.8771 | 0.8782 | 0.8788 | 0.8788 |
| plane | 0.9284 | 0.9287 | 0.9296 | 0.9303 | 0.9306 |
| woman_2 | 0.9553 | 0.9555 | 0.9547 | 0.9542 | 0.9528 |
| sailing3 | 0.9191 | 0.9201 | 0.9209 | 0.9209 | 0.9213 |
| sailing2 | 0.9205 | 0.9207 | 0.9214 | 0.9224 | 0.9218 |
| carnivaldolls | 0.9185 | 0.9205 | 0.9236 | 0.9247 | 0.9232 |
| cemetry | 0.8759 | 0.8798 | 0.8805 | 0.8814 | 0.8793 |
| plant | 0.9534 | 0.9532 | 0.9531 | 0.953 | 0.9524 |
| squirrel | 0.8645 | 0.8648 | 0.866 | 0.8674 | 0.8673 |
| churchandcapitol | 0.8883 | 0.8905 | 0.8932 | 0.8958 | 0.8946 |
| statue | 0.9322 | 0.9319 | 0.9327 | 0.9324 | 0.9324 |
| stream | 0.8036 | 0.8036 | 0.8041 | 0.8047 | 0.8037 |
| bikes | 0.8975 | 0.8991 | 0.9014 | 0.903 | 0.9031 |
| coinsinfountain | 0.9146 | 0.9148 | 0.9159 | 0.9154 | 0.9153 |
| girl_1 | 0.9792 | 0.9795 | 0.9792 | 0.979 | 0.9783 |
| text | 0.9513 | 0.9522 | 0.9513 | 0.9505 | 0.9487 |
| woman_1 | 0.9648 | 0.9657 | 0.9659 | 0.9655 | 0.9643 |
| man_3 | 0.9684 | 0.9684 | 0.9683 | 0.9681 | 0.9671 |
| sailing4 | 0.8691 | 0.8698 | 0.8726 | 0.8737 | 0.8752 |
| lena | 0.9299 | 0.9308 | 0.9328 | 0.9328 | 0.9348 |
| oldMan_1 | 0.9364 | 0.9363 | 0.9367 | 0.937 | 0.9362 |
| oldWoman_1 | 0.9011 | 0.902 | 0.9032 | 0.9038 | 0.9038 |
| parrots | 0.9605 | 0.9609 | 0.9613 | 0.9615 | 0.961 |
| woman | 0.8796 | 0.8803 | 0.881 | 0.8812 | 0.8805 |
| sailing1 | 0.8506 | 0.851 | 0.8523 | 0.8536 | 0.852 |
| lighthouse2 | 0.9024 | 0.9022 | 0.9035 | 0.9043 | 0.9041 |
| face | 0.8828 | 0.8827 | 0.8828 | 0.8825 | 0.8818 |
| flowersonih35 | 0.8645 | 0.8657 | 0.8684 | 0.869 | 0.8674 |
| caps | 0.9353 | 0.9351 | 0.9361 | 0.9363 | 0.9361 |
| man_1 | 0.9722 | 0.9723 | 0.9723 | 0.972 | 0.971 |
| womanhat | 0.9149 | 0.916 | 0.9163 | 0.9162 | 0.916 |
| manfishing | 0.8587 | 0.8597 | 0.8645 | 0.8646 | 0.8626 |
| man_2 | 0.9721 | 0.9719 | 0.9719 | 0.9716 | 0.9709 |
| lighthouse | 0.8734 | 0.874 | 0.874 | 0.8755 | 0.876 |
| monarch | 0.9655 | 0.9666 | 0.9677 | 0.9683 | 0.9678 |
| dancers | 0.8558 | 0.8571 | 0.8607 | 0.8637 | 0.8629 |
| flowers | 0.9244 | 0.9267 | 0.9282 | 0.9278 | 0.9267 |
| house | 0.8945 | 0.8943 | 0.8951 | 0.8949 | 0.8944 |
| redHair | 0.9377 | 0.9375 | 0.9379 | 0.9379 | 0.9374 |
| man_4 | 0.9324 | 0.9322 | 0.9327 | 0.933 | 0.932 |
| building2 | 0.8574 | 0.8592 | 0.861 | 0.8615 | 0.8609 |
| promedio | 0.9122 | 0.9129 | 0.914 | 0.9144 | 0.9139 |

Tabla 4.2: SSIM obtenido de evaluar el modelo basado en diccionarios. Pruebas realizadas con distintos tamaños de parche $(2a + 1) \times (2a + 1)$.

Cada fila (con excepción de la última) representa los resultados para una imagen distinta.

| IMG/ Tamano Parche | <i>a</i> = 5 | <i>a</i> = 4 | <i>a</i> = 3 | <i>a</i> = 2 | <i>a</i> = 1 |
|--------------------|--------------|--------------|--------------|--------------|--------------|
| microchip | 1.15 | 1.4 | 1.62 | 2.19 | 2.15 |
| ocean | 0.83 | 0.89 | 0.98 | 1.0 | 0.8 |
| plane | 1.39 | 1.49 | 1.65 | 1.78 | 1.91 |
| woman_2 | 1.63 | 1.82 | 1.87 | 1.7 | 0.92 |
| sailing3 | 0.87 | 1.17 | 1.34 | 1.38 | 1.02 |
| sailing2 | 1.41 | 1.64 | 1.73 | 1.95 | 1.58 |
| carnivaldolls | 1.38 | 1.5 | 1.31 | 0.97 | 0.91 |
| cemetry | 1.49 | 1.58 | 1.57 | 1.68 | 1.61 |
| plant | 1.16 | 1.29 | 1.36 | 1.21 | -0.55 |
| squirrel | 0.95 | 1.0 | 1.09 | 1.14 | 1.09 |
| churchandcapitol | 1.22 | 1.34 | 1.4 | 1.52 | 1.41 |
| statue | 1.25 | 1.39 | 1.53 | 1.53 | 1.34 |
| stream | 0.83 | 0.93 | 1.1 | 1.18 | 1.14 |
| bikes | 1.54 | 1.75 | 2.02 | 2.17 | 2.09 |
| coinsinfountain | 1.43 | 1.62 | 1.78 | 1.75 | 1.47 |
| girl_1 | 1.73 | 2.05 | 2.29 | 2.19 | 1.23 |
| text | 1.7 | 1.74 | 1.8 | 1.91 | 2.76 |
| woman_1 | 2.09 | 2.54 | 2.84 | 2.77 | 1.94 |
| man_3 | 1.92 | 2.22 | 2.32 | 2.35 | 1.71 |
| sailing4 | 1.1 | 1.21 | 1.38 | 1.43 | 1.35 |
| lena | 1.34 | 1.79 | 2.14 | 2.17 | 1.69 |
| oldMan_1 | 1.33 | 1.51 | 1.74 | 1.94 | 1.71 |
| oldWoman_1 | 0.97 | 1.1 | 1.29 | 1.44 | 1.54 |
| parrots | 1.44 | 1.61 | 1.87 | 2.11 | 1.96 |
| woman | 0.91 | 1.04 | 1.19 | 1.24 | 1.2 |
| sailing1 | 0.86 | 0.96 | 1.24 | 1.3 | 1.2 |
| lighthouse2 | 0.95 | 1.1 | 1.3 | 1.43 | 1.34 |
| face | 0.83 | 0.94 | 1.06 | 1.05 | 0.64 |
| flowersonih35 | 1.05 | 1.13 | 1.18 | 1.21 | 1.18 |
| caps | 1.92 | 1.99 | 2.04 | 2.0 | 1.61 |
| man_1 | 1.81 | 1.95 | 2.17 | 2.12 | 1.15 |
| womanhat | 1.31 | 1.52 | 1.67 | 1.67 | 1.28 |
| manfishing | 1.28 | 1.34 | 1.26 | 1.14 | 1.13 |
| man_2 | 1.76 | 1.97 | 2.09 | 1.91 | 1.07 |
| lighthouse | 1.15 | 1.24 | 1.36 | 1.38 | 1.52 |
| monarch | 2.02 | 2.28 | 2.79 | 3.45 | 3.41 |
| dancers | 0.97 | 1.08 | 1.21 | 1.34 | 1.31 |
| flowers | 1.06 | 1.26 | 1.5 | 1.61 | 1.35 |
| house | 1.07 | 1.23 | 1.36 | 1.42 | 1.27 |
| redHair | 1.13 | 1.32 | 1.49 | 1.54 | 1.19 |
| man_4 | 1.28 | 1.48 | 1.69 | 1.83 | 1.52 |
| building2 | 1.17 | 1.25 | 1.31 | 1.34 | 1.43 |
| promedio | 1.3 | 1.47 | 1.62 | 1.68 | 1.42 |

Tabla 4.3: Incremento en el PSNR obtenido con el modelo basado en redes neuronales artificiales entrenadas con diferentes tamaños de parche ($P = (2a + 1)^2$) respecto al PSNR obtenido por interpolación Bicubic.

Cada fila (con excepción de la última) representa los resultados para una imagen distinta.

| IMG/ Tamano Parche | $a = 5$ | $a = 4$ | $a = 3$ | $a = 2$ | $a = 1$ |
|--------------------|---------------|---------------|---------------|---------------|---------------|
| microchip | 0.9669 | 0.9686 | 0.97 | 0.9712 | 0.9634 |
| ocean | 0.8744 | 0.8753 | 0.8767 | 0.8762 | 0.8709 |
| plane | 0.9246 | 0.9263 | 0.9283 | 0.9294 | 0.9286 |
| woman_2 | 0.9516 | 0.9534 | 0.9538 | 0.9521 | 0.9463 |
| sailing3 | 0.9128 | 0.9156 | 0.9178 | 0.9179 | 0.9109 |
| sailing2 | 0.9167 | 0.9188 | 0.9199 | 0.92 | 0.911 |
| carnivaldolls | 0.9046 | 0.9133 | 0.9149 | 0.9206 | 0.9185 |
| cemetry | 0.8674 | 0.8715 | 0.875 | 0.8769 | 0.8737 |
| plant | 0.9501 | 0.9509 | 0.9514 | 0.95 | 0.9388 |
| squirrel | 0.8564 | 0.8584 | 0.8617 | 0.8632 | 0.8602 |
| churchandcapitol | 0.8816 | 0.8853 | 0.8876 | 0.8909 | 0.885 |
| statue | 0.927 | 0.9289 | 0.9309 | 0.9303 | 0.9256 |
| stream | 0.7896 | 0.7939 | 0.7999 | 0.802 | 0.7997 |
| bikes | 0.8886 | 0.8928 | 0.8978 | 0.8994 | 0.8964 |
| coinsinfountain | 0.908 | 0.9112 | 0.9142 | 0.9143 | 0.9104 |
| girl_1 | 0.9748 | 0.9762 | 0.9769 | 0.9763 | 0.9722 |
| text | 0.8541 | 0.8567 | 0.8606 | 0.8638 | 0.8864 |
| buildings | 0.8189 | 0.8231 | 0.8282 | 0.8316 | 0.8292 |
| woman_1 | 0.9586 | 0.9619 | 0.9635 | 0.963 | 0.9584 |
| man_3 | 0.9642 | 0.9658 | 0.9664 | 0.9663 | 0.9621 |
| sailing4 | 0.8663 | 0.8682 | 0.8714 | 0.8717 | 0.8675 |
| lena | 0.9215 | 0.9258 | 0.9299 | 0.9297 | 0.922 |
| oldMan_1 | 0.9287 | 0.9312 | 0.9333 | 0.9365 | 0.933 |
| oldWoman_1 | 0.8888 | 0.8916 | 0.8964 | 0.8987 | 0.8968 |
| parrots | 0.9582 | 0.9589 | 0.9596 | 0.9585 | 0.9516 |
| woman | 0.8709 | 0.8741 | 0.8774 | 0.8776 | 0.8747 |
| sailing1 | 0.8422 | 0.8459 | 0.8504 | 0.8511 | 0.8473 |
| lighthouse2 | 0.8941 | 0.8965 | 0.8995 | 0.9 | 0.8937 |
| face | 0.8793 | 0.8811 | 0.8834 | 0.8832 | 0.8785 |
| flowersonih35 | 0.8544 | 0.8596 | 0.8602 | 0.8642 | 0.8648 |
| caps | 0.9387 | 0.9388 | 0.9384 | 0.9367 | 0.9307 |
| man_1 | 0.9686 | 0.9696 | 0.971 | 0.9707 | 0.9649 |
| womanhat | 0.9099 | 0.9119 | 0.9139 | 0.9132 | 0.9073 |
| manfishing | 0.8514 | 0.8564 | 0.8578 | 0.8622 | 0.8598 |
| man_2 | 0.9694 | 0.9705 | 0.9713 | 0.9702 | 0.9658 |
| lighthouse | 0.8688 | 0.8708 | 0.8731 | 0.8727 | 0.8688 |
| monarch | 0.9619 | 0.9631 | 0.9649 | 0.9653 | 0.9581 |
| dancers | 0.8453 | 0.8504 | 0.856 | 0.8623 | 0.856 |
| flowers | 0.9193 | 0.9223 | 0.927 | 0.9276 | 0.9212 |
| house | 0.8873 | 0.8899 | 0.8922 | 0.8923 | 0.8868 |
| redHair | 0.9315 | 0.9338 | 0.9358 | 0.9352 | 0.9285 |
| man_4 | 0.9252 | 0.9285 | 0.931 | 0.9325 | 0.928 |
| building2 | 0.8452 | 0.8493 | 0.8527 | 0.8546 | 0.8556 |
| promedio | 0.9027 | 0.9055 | 0.908 | 0.9089 | 0.9049 |

Máximo  Mínimo

Tabla 4.4: SSIM obtenido de evaluar el modelo basado en redes neuronales artificiales con diferentes tamaños de parche $P = (2a + 1)^2$.

Cada fila (con excepción de la última) representa los resultados para una imagen distinta.

4.4. Número Óptimo de Grupos, *K*-Means

Como se menciona en Capítulos previos, los algoritmos propuestos hacen uso de agrupamiento con la finalidad de tomar ventaja de la redundancia a través de imágenes naturales de algunas estructuras de parches pequeños (ver Sec. 3.3.3). En esta Sección se muestra el efecto que tiene en los resultados el usar agrupamiento y la relación que tiene la cantidad de pares de parches en la base de datos con el número de grupos.

Para los experimentos mostrados en esta Sección se construyeron 4 bases de datos con las siguientes características; Los parámetros para el modelo de degradación (o adquisición) se definieron como $s = 2$ y $\sigma = 0.8$, la cantidad de pares de parches en cada base de datos fue de 31200, 124800, 577200 y 1121992 pares de parches para cada base de datos respectivamente, el tamaño de parche se definió como $a = 2$. Los parámetros para el algoritmo de Back-Projection fueron: un kernel gaussiano con $\sigma = 0.8$ usado como el kernel de back-projection B , la cantidad de iteraciones de 750 y una tolerancia de error de $\text{RMSE} < 0.0001$. Para la función de pesos definida en la Eq. (3.6) se definió $h = 12$. Se usó $\lambda = 0.1$ para el modelo basado en diccionarios, tanto para el entrenamiento como para la reconstrucción.

Cada uno de los experimentos consistió en probar diferentes cantidades de grupos K , es decir, se tomó una de las bases de datos y se dividió en K grupos. Con estos grupos se aprendieron los parámetros para el modelo basado en diccionarios y se evalúo un conjunto de imágenes para obtener el PSNR del resultado de cada imagen. Este experimento se repitió para los diferentes valores de K , específicamente para los valores 1, 2, 8, 32, 64, 256, 512, 1024 y 2048, y para cada base de datos, de manera que el conjunto de imágenes se evalúo 36 veces, 4 bases de datos por 9 distintos tamaños de agrupamiento.

La diferencia entre el PSNR obtenido por interpolación bicubic y el obtenido por el modelo basado en diccionarios es graficada en las figuras figuras Fig. 4.2 y Fig. 4.3. Como se puede observar en estas mismas figuras, a mayor número de parches en la base de datos los resultados parecen estabilizarse de manera que a mayor cantidad de grupos se obtiene un mejor PSNR. Para la imagen representada por la línea roja (cara) se puede observar que el PSNR no mejora conforme se aumenta el número de grupos y la cantidad de parches. Sin embargo, se observa una estabilización de manera que para bases de datos de tamaño 577200 y 1121992 decrece de manera más lenta conforme se aumenta el número de grupos. Para las imágenes representadas por las líneas verde y azul se observa que el PSNR aumenta de manera considerable con mayor información en la base de datos de entrenamiento y

con un agrupamiento al rededor de 1024 grupos.

Dado que el experimento de la Fig. 4.2 sugiere que para bases de datos de tamaño mayor a medio millón de pares de parches los resultados parecen estabilizarse, se decidió realizar un segundo experimento usando el modelo de redes neuronales artificiales. Se construyó una base de datos de aprox. 577000 pares de parches usando los mismos parámetros definidos para las bases de datos en esta Sección y se estimaron los parámetros para el modelo de redes neuronales artificiales usando diferentes tamaños de agrupamiento, específicamente 60, 254, 528, 1024 y 2000. Con cada modelo estimado con los diferentes tamaños de agrupamiento se evaluó un conjunto de 47 imágenes naturales de escenarios muy diferentes entre sí y de manera similar al experimento de la Fig. 4.2, se grafica la diferencia entre el PSNR obtenido por el método de interpolación bicubic y el PSNR obtenido con el modelo de redes neuronales artificiales. Los resultados se muestran en las gráficas de la Fig. 4.4 y Fig. 4.5.

Podemos observar en la Fig. 4.4 que para la mayoría de las imágenes el PSNR incrementa más usando un tamaño de agrupamiento entre 528 y 2000 grupos, lo cual tiene cierta similitud al resultado obtenido con el modelo basado en diccionarios.

En general se puede observar que el PSNR aumenta para la mayoría de las imágenes de prueba si los modelos fueron entrenados usando una base de datos con más de medio millón de pares de parches y con un tamaño de agrupamiento mayor a 500 grupos. Los dos modelos propuestos parecen tener un comportamiento similar respecto al tamaño de la base de datos y cantidad de grupos.

4.5. Estimación de los Parámetros para el Modelo Basado en Redes Neuronales Artificiales

Uno de los problemas con las redes neuronales artificiales es determinar el número de capas ocultas y las neuronas dentro de cada capa. En el trabajo de Karsoliya (2012) sobre estimación de número de neuronas en las capas ocultas se mencionan dos reglas para definir estos parámetros. La primera regla sugiere no tener más de dos veces la cantidad de neuronas en la capa de entrada como el total de neuronas en las capas ocultas. La segunda regla sugiere tener de 1 a 3 capas ocultas con aproximadamente dos tercios de la cantidad de neuronas en la capa de entrada en cada capa oculta. Puede observarse que estas reglas no se contradicen entre si, ya que si se tienen 3 capas ocultas con dos tercios de las neuronas en la capa de entrada en cada capa, el total de neuronas sera de el justamente el doble de la

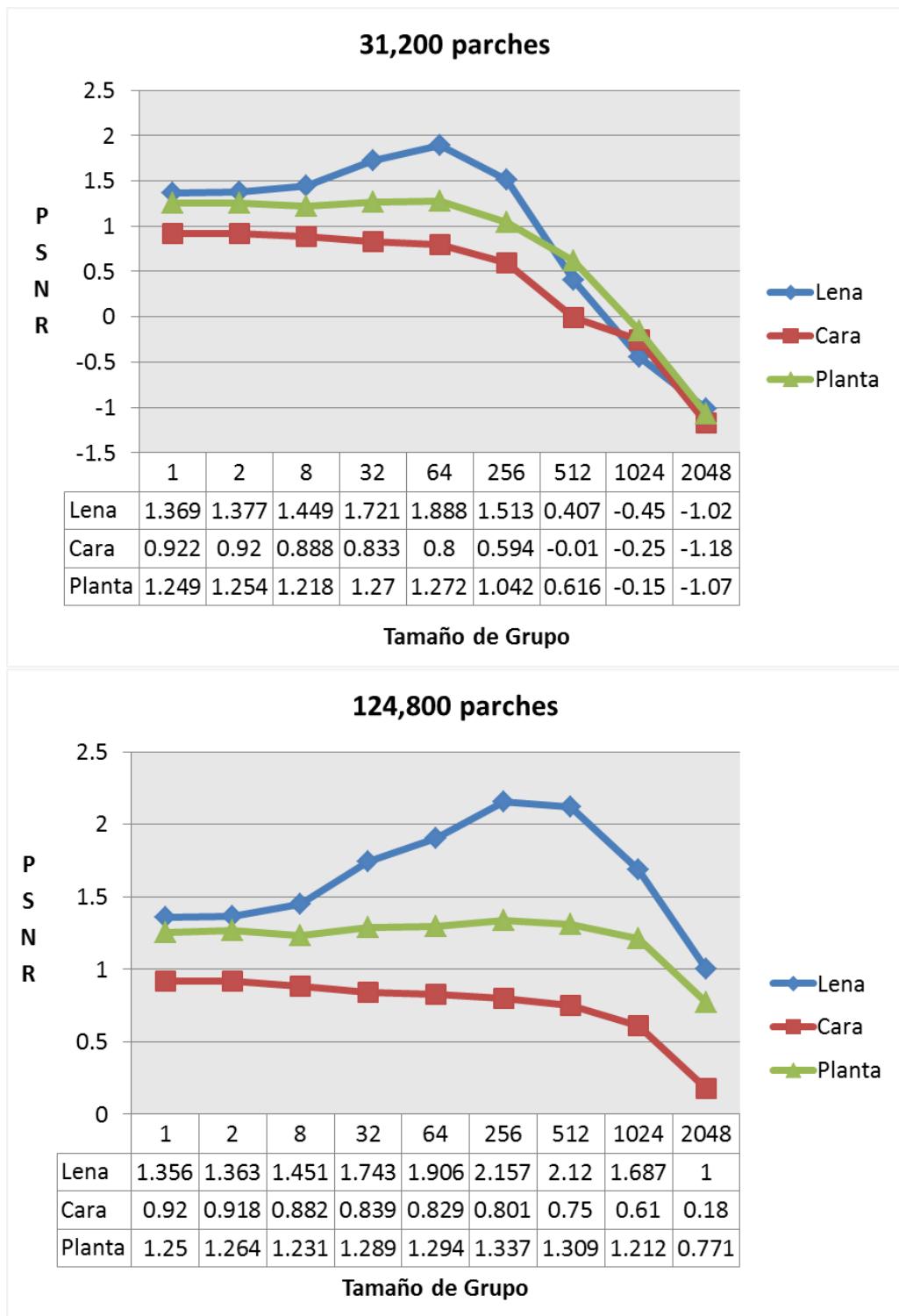


Figura 4.2: Incremento del PSNR obtenido con diccionarios respecto al PSNR obtenido con interpolación bicubíca.

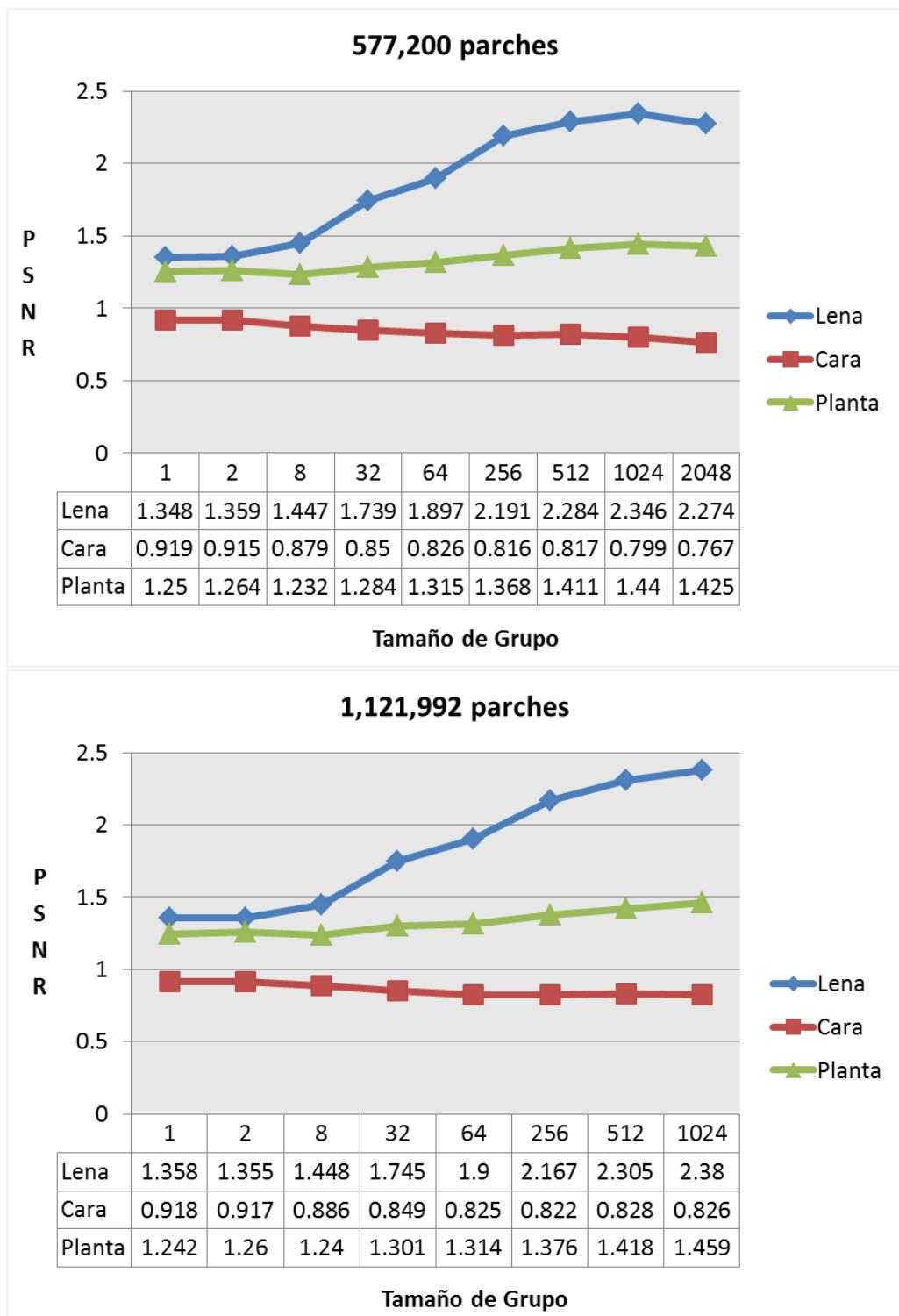
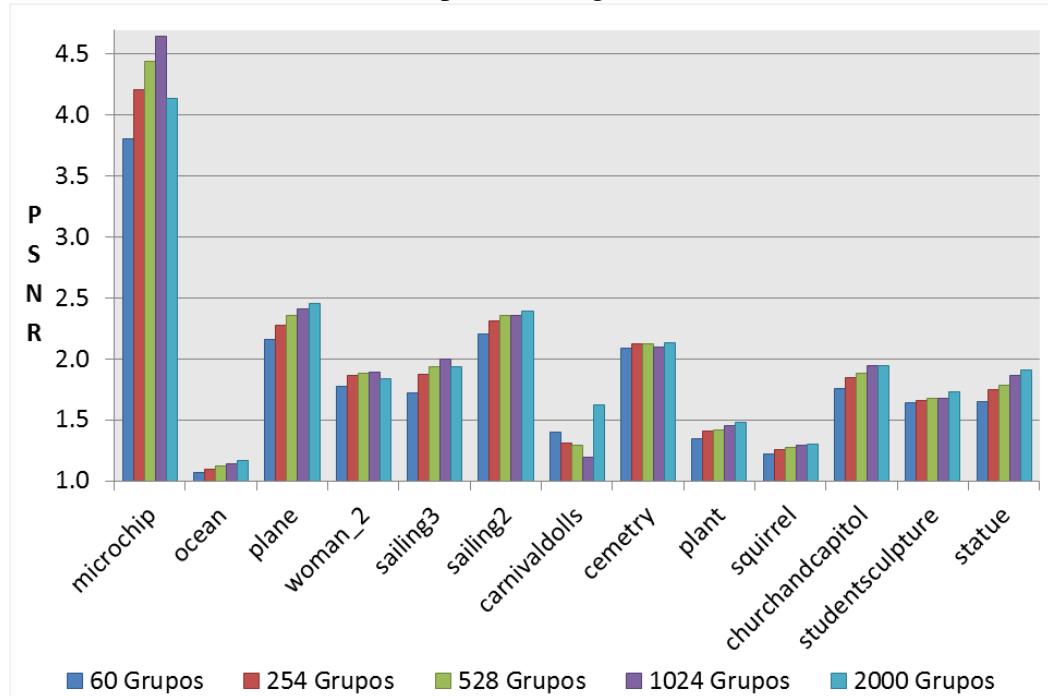


Figura 4.3: Incremento del PSNR obtenido con diccionarios respecto al PSNR obtenido con interpolación bicubic. Continuación de la Fig. 4.2

Grupo 1 de imágenes.



Grupo 2 de imágenes.

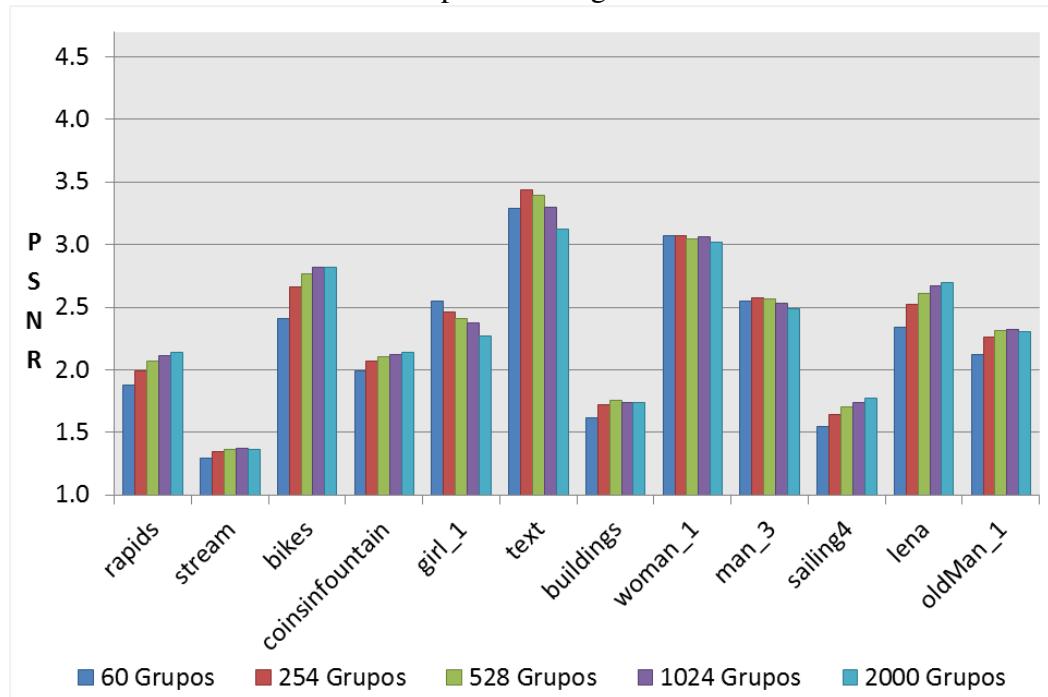
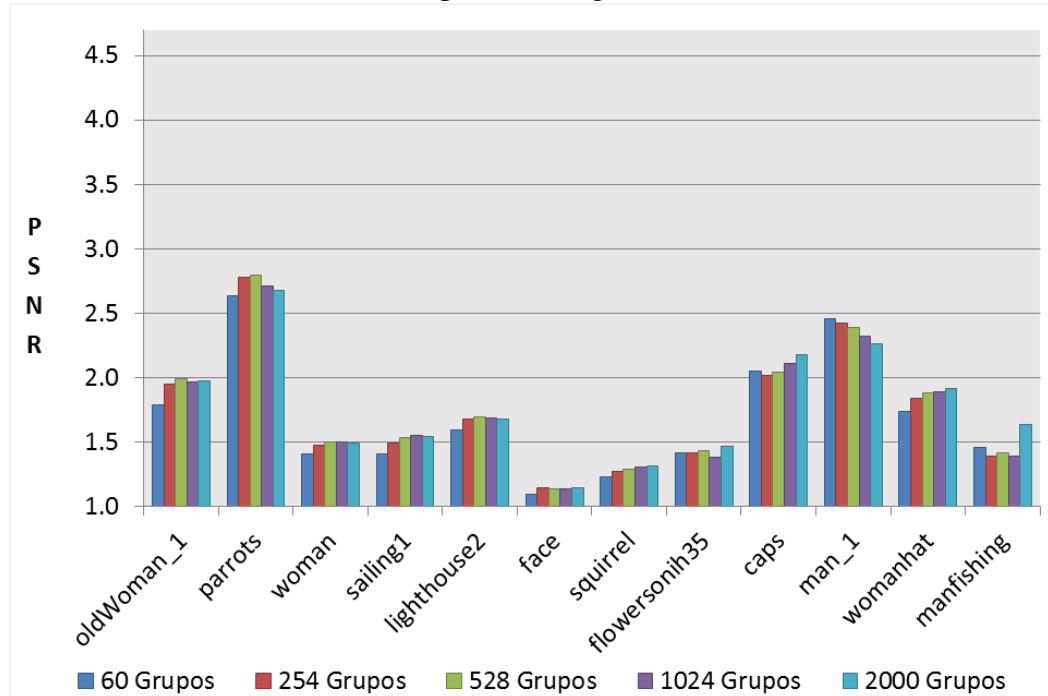


Figura 4.4: Incremento del PSNR obtenido con redes neuronales artificiales respecto al PSNR obtenido con interpolación bicubic.

Grupo 3 de imágenes.



Grupo 4 de imágenes.

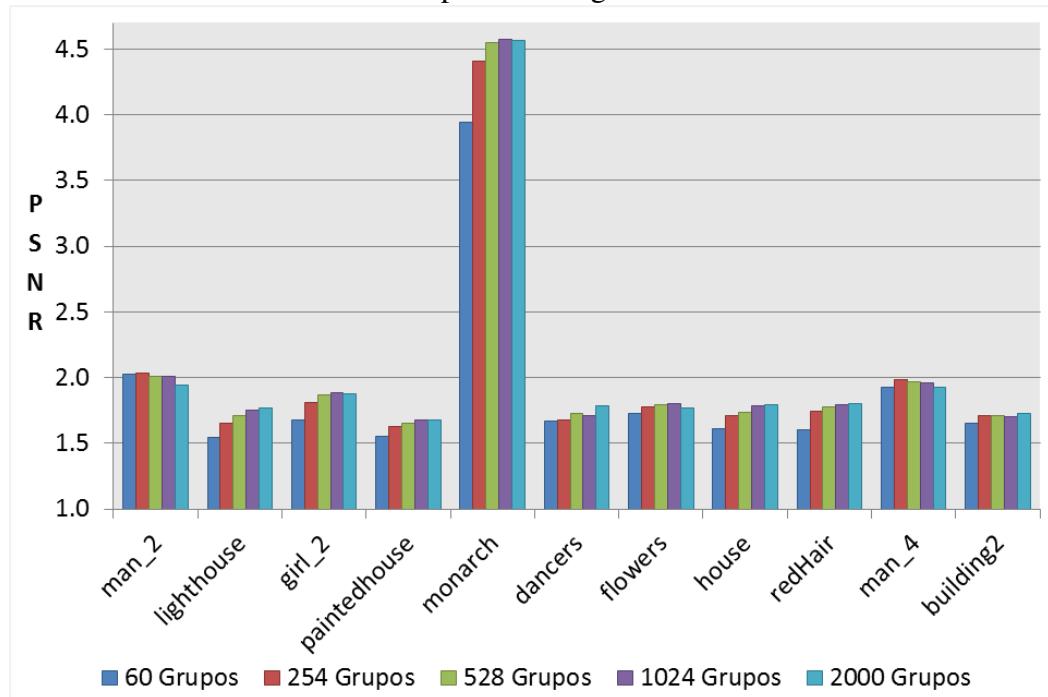


Figura 4.5: Incremento del PSNR obtenido con redes neuronales artificiales respecto al PSNR obtenido con interpolación bicubic.

capa de entrada.

Para los experimentos mostrados en esta Sección se construyó 1 base de datos con las siguientes características; Los parámetros para el modelo de degradación (o adquisición) se definieron como $s = 2$ y $\sigma = 0.8$, la cantidad de pares de parches en la base de datos fue de 577200 pares de parches, el tamaño de parche se definió como $a = 2$. Los parámetros para el algoritmo de Back-Projection fueron; un kernel gaussiano con $\sigma = 0.8$ usado como el kernel de back-projection B , la cantidad de iteraciones de 750 y una tolerancia de error de $\text{RMSE} < 0.0001$. Para la función de pesos definida en la Eq. (3.6) se definió $h = 12$.

En base a lo anterior se desarrolló el siguiente experimento; Se entrenaron 3 conjuntos de redes neuronales artificiales con distintas configuraciones. El primer conjunto está formado por redes neuronales artificiales con la configuración $25 \times 16 \times 25$, que quiere decir, 25 neuronas en la capa de entrada, 16 neuronas en la única capa oculta y 25 neuronas en la capa de salida. El segundo conjunto está formado por redes neuronales artificiales con la configuración $25 \times 16 \times 16 \times 25$, 25 neuronas en la capa de entrada, 16 neuronas en cada una de las dos capas ocultas y 25 neuronas en la capa de salida. Finalmente, el tercer conjunto está formado por redes neuronales artificiales con la configuración $25 \times 16 \times 16 \times 16 \times 25$. Se evaluó un conjunto de imágenes con cada conjunto de redes neuronales y se calculó el PSNR para cada imagen en cada conjunto, los resultados se muestran en la Tabla 4.5.

En la Tabla 4.5 podemos observar que los resultados no varían demasiado conforme se agregan más capas ocultas al problema. Esto es un buen resultado ya que menor cantidad de capas representa menor cantidad de neuronas en la red y esto se traduce por un menor número de operaciones al momento de evaluar la red neuronal. Este experimento se probó con más imágenes y el resultado es similar. En base a estos resultados se sugiere que usar una capa oculta con la cantidad de neuronas igual a dos tercios de neuronas de la capa de entrada es suficiente para obtener resultados competitivos.

4.6. Estimación de los Parámetros para el Modelo Basado en Diccionarios

En esta Sección se muestran algunos experimentos que ayudan a definir algunos de los parámetros relacionados con el modelo basado en diccionarios. Entre estos parámetros se encuentran el número de átomos en los diccionarios (tamaño del diccionario), el factor λ que define que tan sparse es la solución con respecto al diccionario de baja resolución y

| Imagen / Capas | 1 | 2 | 3 |
|------------------|-------|-------|-------|
| microchip | 35.11 | 35.12 | 34.91 |
| plane | 31.92 | 31.94 | 31.9 |
| sailing3 | 32.37 | 32.3 | 32.28 |
| carnivaldolls | 26.68 | 26.76 | 26.9 |
| plant | 39.6 | 39.61 | 39.65 |
| churchandcapitol | 26.81 | 26.83 | 26.82 |
| statue | 32.28 | 32.29 | 32.27 |
| stream | 24.43 | 24.41 | 24.39 |
| coinsinfountain | 30.03 | 30.0 | 29.98 |
| text | 20.95 | 20.8 | 20.7 |
| woman.1 | 35.34 | 35.31 | 35.27 |
| sailing4 | 29.99 | 29.98 | 29.97 |
| Promedio | 30.59 | 30.58 | 30.56 |

Máximo Mínimo

Tabla 4.5: PSNR vs Número de Capas Ocultas.

En esta tabla se muestra el PSNR obtenido mediante diferentes conjuntos de redes neuronales artificiales con diferente configuración. La variación de los resultados no es significativa por lo que usar una capa oculta es suficiente. Cada fila (con excepción de la última) representa los resultados para una imagen distinta.

a la cantidad de diccionarios que se usarán para la estimación de un nuevo parche de alta resolución.

4.6.1. Cantidad de Átomos en el Diccionario

Para los experimentos mostrados en esta Sección se construyó 1 base de datos con las siguientes características; Los parámetros para el modelo de degradación (o adquisición) se definieron como $s = 2$ y $\sigma = 0.8$, la cantidad de pares de parches en la base de datos fue de 577700 pares de parches y el tamaño de parche se definió como $a = 2$. Los parámetros para el algoritmo de Back-Projection fueron: un kernel gaussiano con $\sigma = 0.8$ usado como el kernel de back-projection B , la cantidad de iteraciones de 750 y una tolerancia de error de $\text{RMSE} < 0.0001$. Para la función de pesos definida en la Eq. (3.6) se usó el caso especial donde sólo se toma en cuenta el par de diccionarios tal que el centroide del grupo usado para entrenar ese par de diccionarios es el mas parecido. Se usó $\lambda = 0.1$ para el modelo basado en diccionarios, tanto para el entrenamiento como para la reconstrucción.

Aplicando 3 tamaños de agrupamiento a la base de datos descrita en el párrafo anterior (64, 300 y 1024 grupos respectivamente), se aprendieron 4 conjuntos de parámetros (para cada tamaño distinto de agrupamiento) para el modelo basado en diccionarios con distintas cantidades de átomos en cada conjunto, un conjunto de parámetros con pares de diccionarios con 30 átomos cada diccionario y para los 3 conjuntos restantes con 60 átomos, 120

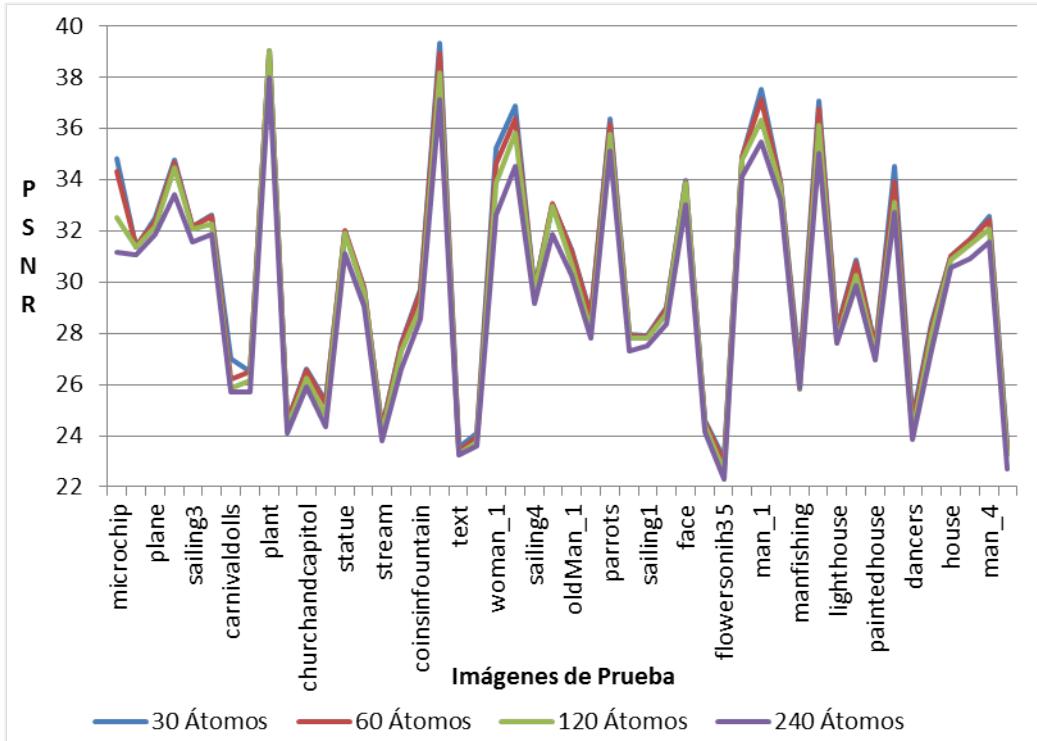
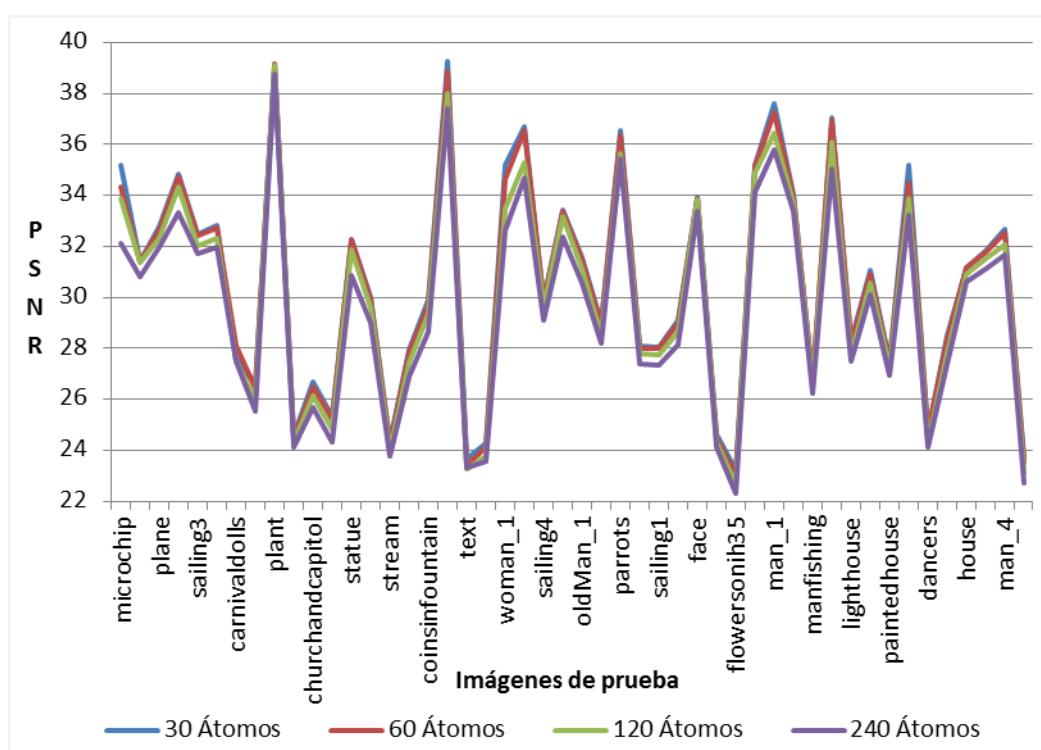
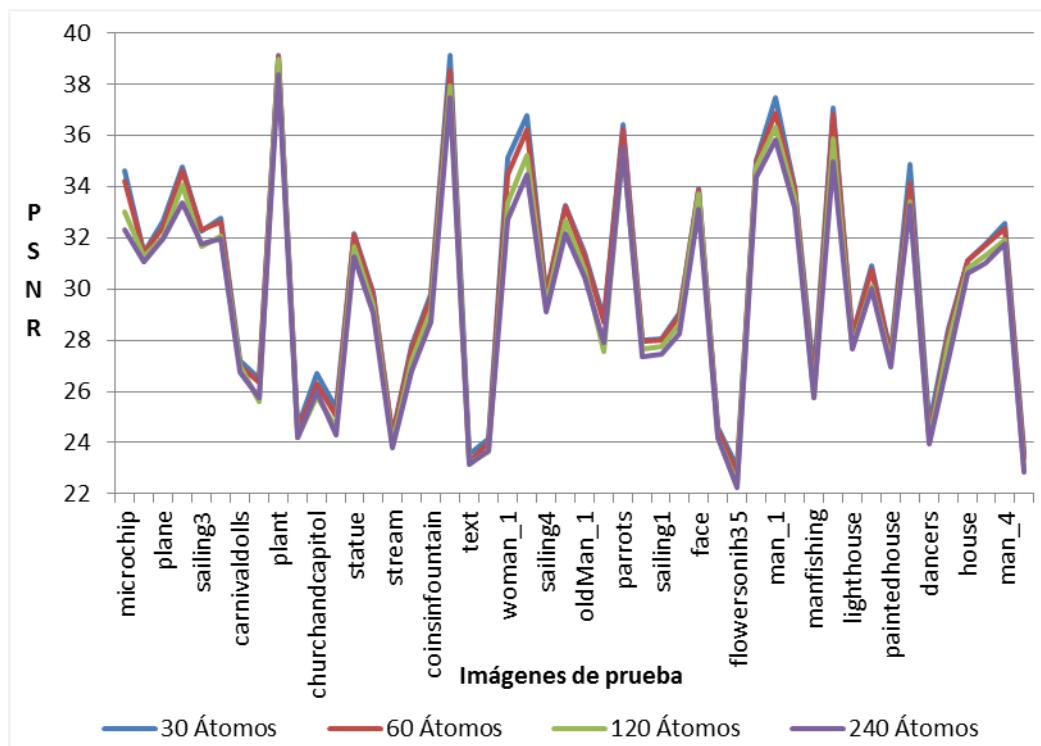


Figura 4.6: PSNR vs Número de Átomos - 64 Grupos.

átomos y 240 átomos cada uno. Aplicando el algoritmo 6 con cada uno de los conjuntos de parámetros estimados anteriormente (12 conjuntos en total), se calculó el PSNR de un conjunto de 47 imágenes naturales de prueba. En las Figuras Fig. 4.6, Fig. 4.7 y Fig. 4.6 se muestra el PSNR del resultado de las pruebas.

En la gráfica de la Fig. 4.6 se puede notar un patrón, ya que el PSNR va decreciendo conforme se incrementa el número de átomos. Esto sugiere que los diccionarios aprenden bases que representan mejor los parches de entrenamiento con un número pequeño de átomos. Con el objetivo de confirmar esto y verificar que el número de átomos no está ligado a la cantidad de parches usados para el entrenamiento, se realizaron experimentos similares con diferentes cantidades de parches y grupos.

En todas las figuras Fig. 4.7 y Fig. 4.8 se puede observar el mismo comportamiento. La línea correspondiente a diccionario con 30 átomos se mantiene por arriba de las demás. Esto nos permite suponer que diccionarios con un tamaño pequeño (30 átomos) son suficientes para aprender las bases de cada grupo de parches. Esto probablemente se deba a que cada grupo contiene en su mayoría parches con estructuras similares lo que permite representar estos parches con un número menor de bases.



4.6.2. Estimación del Factor Sparse en el Modelo Basado en Diccionarios - ¿Solución Sparse?

En la Eq. (3.16)(Sec, 3.3.2.1) se definió un parámetro λ que es el que controla que tan sparse es el vector α . En esta Sección se trata de explicar mediante experimentos cual es el efecto que tiene este factor en los resultados. Recordamos que α es el vector que representa los coeficientes de un parche de baja resolución sobre un diccionario D^y , jugando un papel importante en el modelo propuesto. Es por ello que se considera importante mostrar el comportamiento de este factor.

Para los experimentos mostrados en esta Sección se construyó 1 base de datos con las siguientes características; Los parámetros para el modelo de degradación (o adquisición) se definieron como $s = 2$ y $\sigma = 0.8$, la cantidad de pares de parches en la base de datos fue de 65400 pares de parches, el tamaño de parche se definió como $a = 2$. Los parámetros para el algoritmo de Back-Projection fueron; un kernel gaussiano con $\sigma = 0.8$ usado como el kernel de back-projection B , la cantidad de iteraciones de 750 y una tolerancia de error de $\text{RMSE} < 0.0001$. Para la función de pesos definida en la Eq. (3.6) se usó el caso especial donde sólo se toma en cuenta el par de diccionarios tal que el centroide del grupo usado para entrenar ese par de diccionarios es el mas parecido.

Usando la base de datos descrita anteriormente, se entrenaron diferentes conjuntos de parámetros para el modelo basado en diccionarios usando para cada conjunto de parámetros un valor distinto de λ , los valores usados son 5, 20, 35 y 50, la cantidad de átomos en cada diccionario fue de 300 átomos. Las pruebas realizadas en la Sección anterior (Sec. 4.6.1) mostraron que los diccionarios de tamaño cercano a 30 átomos generan buenos resultados. Sin embargo, el valor de λ usado en esas pruebas fue de 0.1 lo que significa que el peso para que la solución sea sparse es pequeño (en general esto dio mejores resultados) y como en esta Sección lo que se quiere mostrar es el comportamiento de λ entonces se decidió incrementar la cantidad de átomos (hasta 300 átomos) para tener diccionarios sobre-completos y así poder ver los efectos de λ sobre las imágenes resultantes.

En la Fig. 4.9 se puede observar que en la imagen A al parecer las bases o átomos aprendidos en los diccionarios no fueron suficiente buenas y generan una especie de ruido al rededor de los bordes. Sin embargo, al incrementar λ este problema parece disminuir, pero algunos detalles importantes de la imagen comienzan a desaparecer. Si se observan las huellas dactilares del dedo que está sosteniendo el microchip (mayor parte del fondo de la imagen) entonces puede verse que conforme λ incrementa estas huellas van desapareciendo y esto no es deseado. En general, este efecto aparece sin importar la cantidad de átomos en el diccionario, y es normal, ya que al aumentar λ se exige mayor *sparsidad* en la

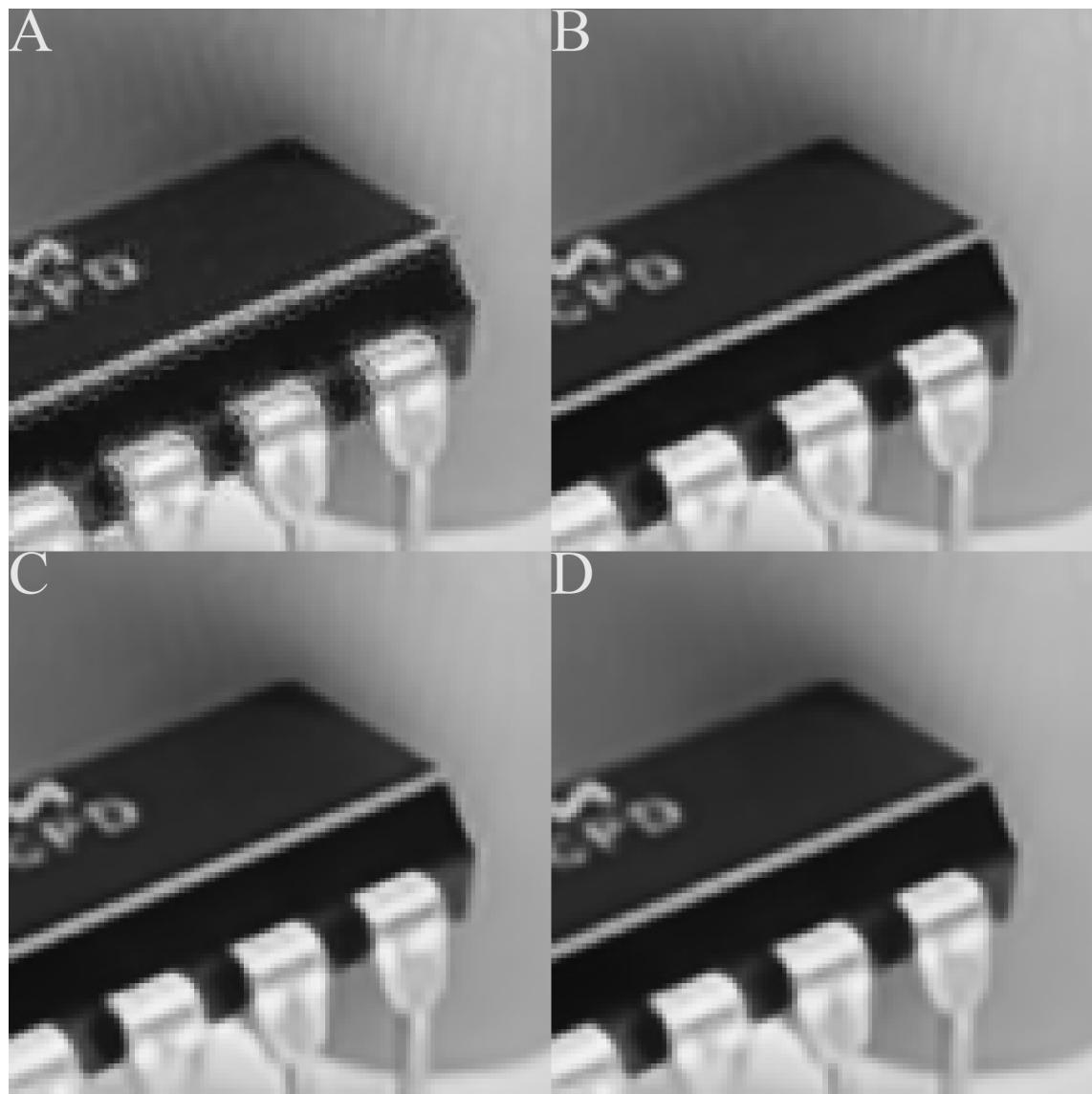


Figura 4.9: Efecto al incrementar λ .

Imágenes obtenidas mediante conjuntos de diccionarios entrenados con distintos valores de λ . A) $\lambda = 5.0$, B) $\lambda = 20.0$, C) $\lambda = 35.0$ y D) $\lambda = 50.0$.

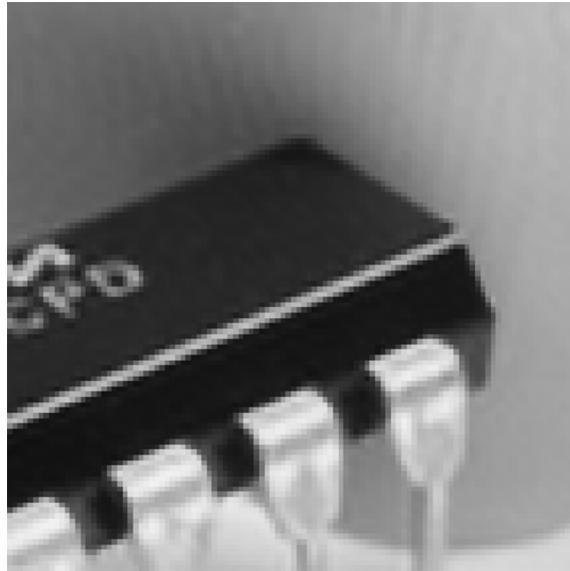


Figura 4.10: Resultado para $\lambda = 0.1$.

solución, buscando la base o átomo que mejor represente el parche de baja resolución sobre el diccionario de baja resolución, difícilmente una base podrá representar bien un parche. Los resultados de este trabajo sugieren que las soluciones no deben ser tan sparse como se ha popularizado en los métodos del estado del arte. Por el contrario es mejor tener un conjunto de átomos cuya combinación lineal pueda adaptarse de una mejor manera a los parches de baja resolución.

En la Fig. 4.10 se puede observar el resultado de una prueba sobre un conjunto de diccionarios aprendidos con $\lambda = 0.1$ y solamente 30 átomos, ya que se busca que los parches de alta resolución estimados sean combinación lineal de varios átomos en el diccionario. Se puede notar que los bordes están bien definidos y las huellas dactilares no se pierden como pasa en el caso de valores para λ demasiado grandes.

En la Fig. 4.11 se muestra el incremento en el PSNR obtenido por dos conjuntos de pares de diccionarios entrenados con diferentes valores para λ respecto al PSNR obtenido por interpolación bicubic. El primer conjunto se entrenó con $\lambda = 0.1$ y el segundo con $\lambda = 1.0$, la cantidad de átomos en cada diccionario fue de 30 átomos. Se puede observar una ligera ventaja del valor de $\lambda = 0.1$. Este experimento apoya la hipótesis de que se pueden obtener mejores resultados tomando en cuenta una mayor cantidad de bases en el diccionario, es decir, buscando una solución que no sea tan sparse.

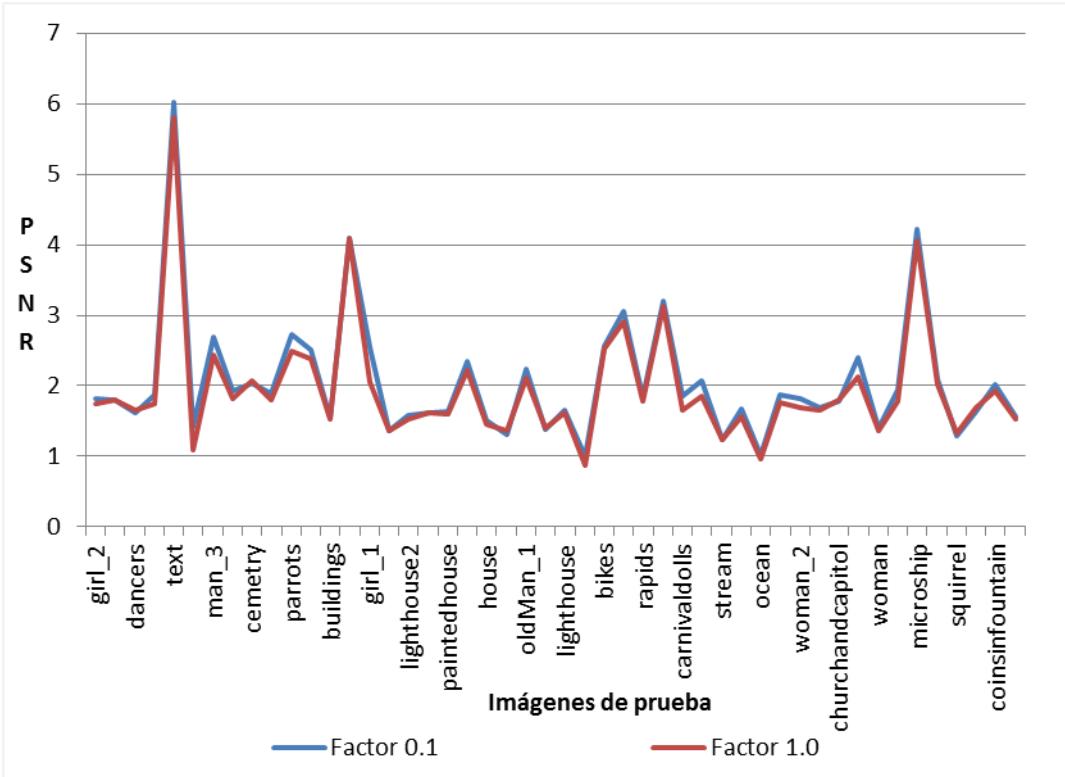


Figura 4.11: Comportamiento del factor λ .

4.7. Estimación del Parámetro de la Función de Pesos

La función de pesos $w(., .)$ (Eq. 3.5) depende de un parámetro h que se puede interpretar como el factor que determina cuantas redes neuronales artificiales o cuantos pares de diccionarios interfieren en la estimación de un parche de alta resolución. La función de pesos $w(., .)$ determina en cierta forma qué tan parecido es un parche y_n a un centroide μ_k y en base a esto le da un peso a la estimación de x_n usando la k -ésima red neuronal artificial o los pares de diccionarios, según corresponda.

La función $w(., .)$ puede ser usada para optimizar el proceso de reconstrucción de distintas maneras. Por ejemplo, si el peso generado por $w(., \mu_k) < \epsilon$ (umbral o restricción de similaridad) para $\epsilon \in [0, 1]$ entonces se decide evaluar el modelo usando los parámetros del conjunto k . De esta forma el usuario tendrá control de que tan parecidos deben ser los centroides para decidir evaluar un parche con los parámetros estimados a partir del conjunto de datos del centroide de referencia. El problema con la sugerencia anterior es que existen parches que pueden no cumplir la restricción de similaridad (ver Tab. 4.6) y se quedarían sin estimación. Otra forma es usar los l modelos más cercanos al parche pesados por la función $w(., .)$. De esta manera, se asegura que todos los parches serán evaluados y

la estimación será pesada por la similitud que tienen los parches con los centroides.

En la práctica se encontró que un valor de $h \geq 12$ tiene una gran probabilidad de generar valores tal que $w(.,.) \geq 0.1$. Esto es deseado ya que para pesos menores a 0.1 se podría considerar que el impacto que tendrán en la solución no será significativo.

En la Fig. 4.12 se muestra la distribución de las evaluaciones sobre cada grupo (de un conjunto de 1024 grupos). Esto sugiere dos cosas importantes; La primera es que la mayoría de los parches son mapeados a un conjunto mínimo de grupos, es decir, existe una estructura de parches que es más común a las demás y esto es lógico ya que gran parte del área de las imágenes son zonas planas, sin gradientes, es por ello que todo este conjunto de parches es ligado a un conjunto pequeño de grupos. La segunda cosa que se puede observar en esta gráfica es que al incrementar el valor de h se comienzan a tomar en cuenta un mayor número de grupos.

Tomar en cuenta un mayor número de grupos para aproximar el valor de un parche de alta resolución tiene sentido ya que, como se menciona anteriormente, el problema de Single Image Super Resolution es mal condicionado debido a que existen muchas soluciones \hat{x}_n que pueden generar un parche y_n , entonces evaluar varios grupos cuyos centroides son parecidos a y_n podría mejorar la estimación. Por otro lado, si $h \rightarrow \infty$ entonces se tomarán en cuenta todos los grupos con el mismo peso al estimar un parche de alta resolución y esto obviamente generará un gran error en la estimación. En la práctica se encontró que considerar un conjunto de 5 a 15 grupos pesados por $w(.,.)$ con $h = 12$ genera buenos resultados ayudando a mejorar la estimación tanto en PSNR como en SSIM.

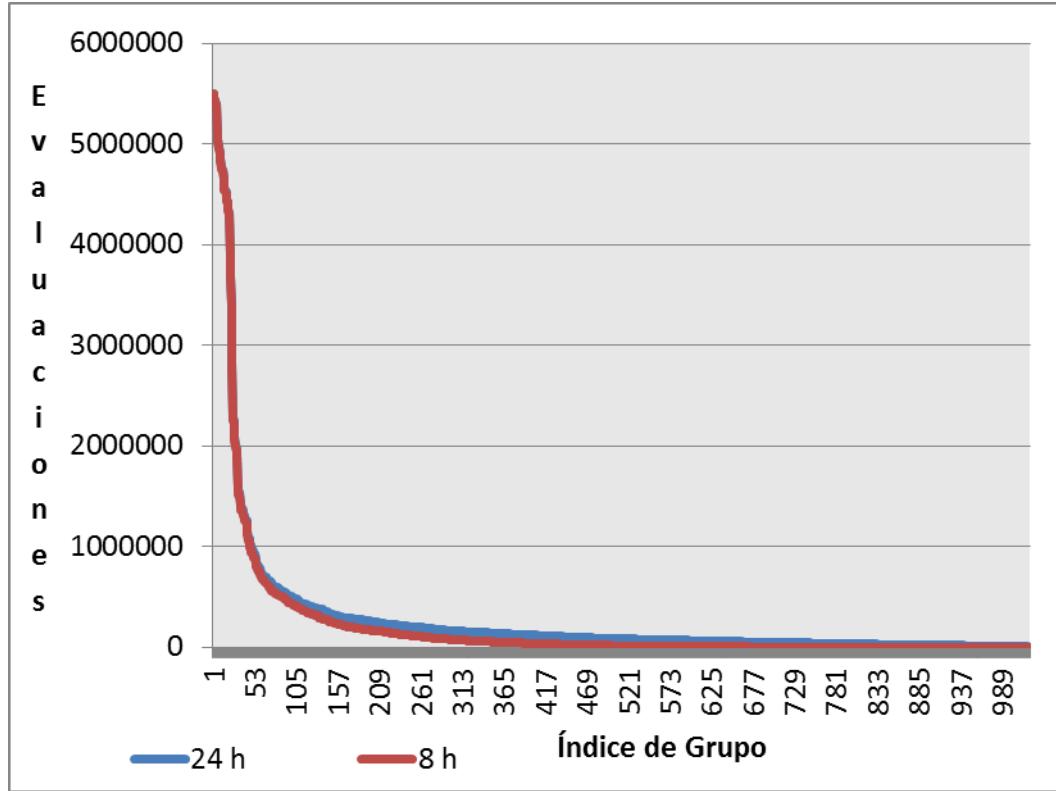


Figura 4.12: Cantidad de evaluaciones de redes neuronales artificiales.

Sobre un conjunto de 1024 grupos y 47 imágenes de prueba se calculó el número de evaluaciones de redes neuronales artificiales y la distribución de estas evaluaciones sobre los grupos, es decir, cuantas veces se evaluó la red neuronal artificial que pertenece al grupo k -ésimo.

| h | Total de Evals | Promedio | 1 Quartile | 2 Quartile | 3 Quartile | Min Eval p/g |
|-----|----------------|-----------|------------|------------|------------|--------------|
| 24 | 286607834 | 279890.46 | 38643.5 | 83623 | 203262.5 | 3509 |
| 20 | 284131625 | 277472.29 | 34875.75 | 80237 | 202466.25 | 2302 |
| 16 | 277969897 | 271454.98 | 26219.25 | 70713.5 | 195947.25 | 828 |
| 12 | 263057693 | 256892.28 | 12080.25 | 49164.5 | 170621.5 | 87 |
| 8 | 229727238 | 224343.01 | 1120.75 | 13751.5 | 108755.25 | 0 |

Tabla 4.6: Datos sobre la distribución de las evaluaciones de redes neuronales artificiales sobre un conjunto de parches.

La columna *Min Eval p/g* significa la cantidad mínima de evaluaciones que tuvo alguna o varias redes neuronales artificiales asociadas con algún grupo. En esta tabla se pueden observar otros datos sobre la distribución de las evaluaciones de las redes neuronales artificiales sobre cada grupo. También se puede ver que al incrementar el parámetro h se comienzan a realizar una mayor cantidad de evaluaciones, en general los valores que más cambian son el primer y segundo quartile. Finalmente, se puede observar que para un valor de $h = 8$ hay grupos que no cumplen con la restricción de similitud y por lo tanto nunca son considerados en cuenta para la estimación de algún parche de alta resolución.

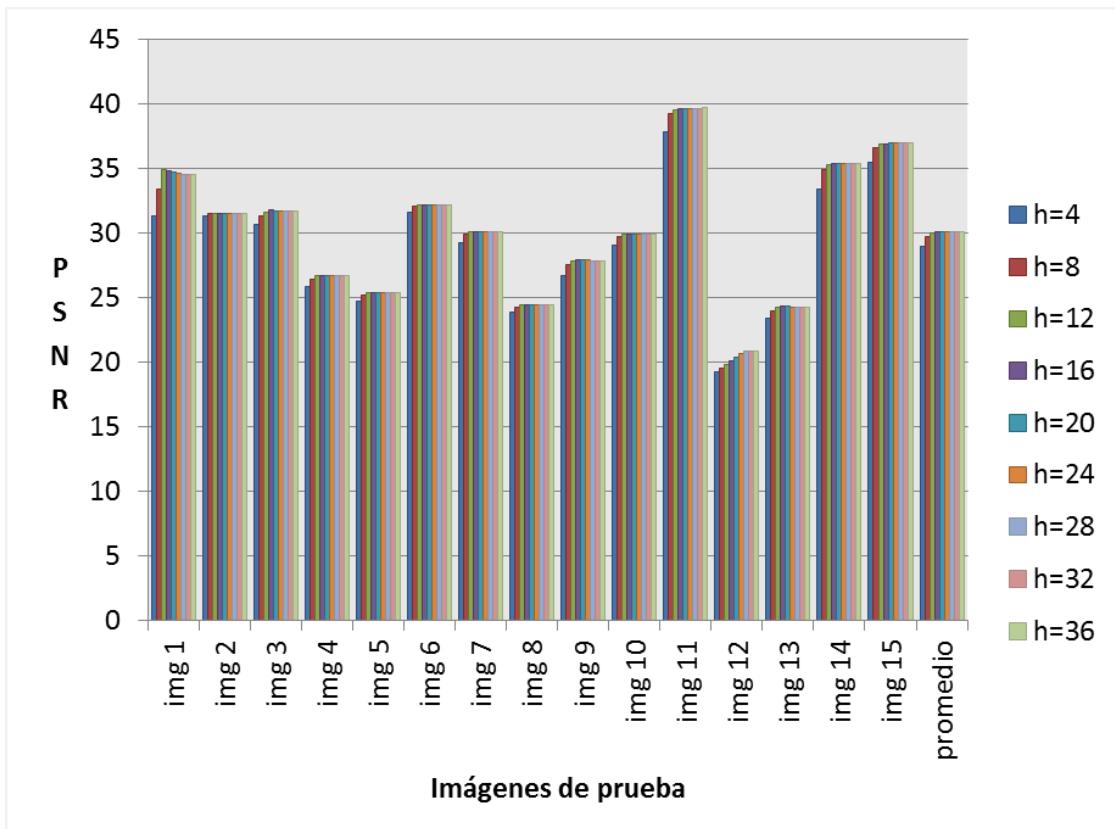


Figura 4.13: PSNR Obtenido variando los valores del parámetro h .
 Las pruebas se realizaron con el modelo de redes neuronales artificiales, 1024 redes neuronales artificiales fueron entrenadas con aprox. tres cuartos de millón de parches.

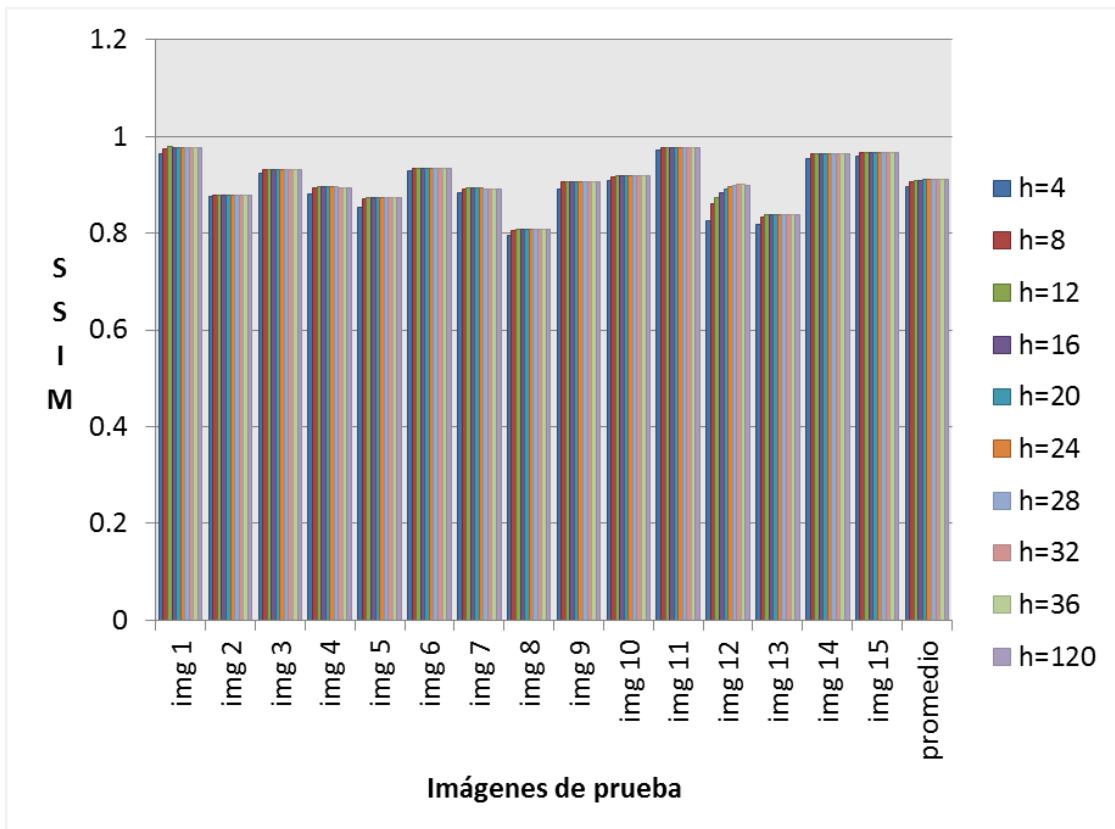


Figura 4.14: SSIM Obtenido variando los valores del parámetro h .
Las pruebas se realizaron con el modelo de redes neuronales artificiales, 1024 redes neuronales artificiales fueron entrenadas con aprox. tres cuartos de millón de parches.

En la Fig. 4.13 se puede observar el comportamiento del PSNR al incrementar el factor h de la función de pesos $w(.,.)$. Para esta prueba se usaron los primeros 15 grupos con la distancia más pequeña con respecto a cada parche de baja resolución. Se puede observar que al llegar al valor de $h = 12$ se alcanza una buena estimación y de ahí en adelante se estabiliza con los demás valores de h . Es importante resaltar que si se toman en cuenta una mayor cantidad de grupos entonces al incrementar h el PSNR comenzará a bajar.

Aunque las pruebas en esta sección se realizaron con el modelo de redes neuronales artificiales, los resultados son similares en el modelo basado en diccionarios.

4.8. Comparación con Otros Algoritmos

En esta Sección se comparan los resultados obtenidos por las propuestas presentadas en este trabajo de tesis y los trabajos desarrollados por Irani and Peleg (1991), Kim and Kwon (2010), Glasner et al. (2009), Freedman and Fattal (2010), Dong et al. (2011), Yang and Yang (2013), Timofte et al. (2014) y Dong et al. (2014). Todos los códigos fueron provistos por uno de los autores de Yang et al. (2014) con excepción del trabajo de Dong et al. (2014) que fue obtenido de una de las páginas web de sus autores.

La base de datos usada para realizar las pruebas es la denominada *Live1* en el trabajo de Yang et al. (2014). Esta base de datos consta de 29 imágenes de distintas escenas. Además de esta base de datos, se usaron 10 imágenes de rostros del autor Daniel Boschung's y otras 8 imágenes más tomadas de Google Imágenes, para así formar una base de prueba de 47 imágenes de distintas fuentes y con distintas resoluciones.

A lo largo de esta Sección se muestran tablas con los valores obtenidos con cada método usando las métricas PSNR y SSIM por ser de las más usadas en la literatura. Los trabajos de Freedman and Fattal (2010) y Dong et al. (2011) no se incluyen en las tablas debido a que los resultados que generan no están alineados con la imagen real y esto hace que no se puedan usar las métricas anteriores. También se muestran resultados visuales comparando diferentes imágenes obtenidas por todos los algoritmos.

La mayoría de las pruebas se realizaron con $\sigma = 0.8$ y $s = 2$. En general el comportamiento de los algoritmos es similar si se varían los valores de σ entre $[0, 1.0]$. En cambio para valores de $s = 4$ o $s = 8$ los resultados comienzan a variar y en general se complica la evaluación con estos factores de escala debido a que la mayoría de las implementaciones

de los métodos con quien comparamos el trabajo de esta tesis son extremadamente demandantes en tiempo computacional, con excepción del trabajo de Dong et al. (2014). Es por ello que la mayoría de las pruebas presentadas en esta Sección son para un factor de escala igual a dos.

4.8.1. Pruebas con factor de escala 2x

Para los experimentos mostrados en esta Sección se construyeron 2 bases de datos (una para cada modelo propuesto) con las siguientes características; Los parámetros para el modelo de degradación (o adquisición) se definieron como $s = 2$ y $\sigma = 0.8$ para ambas bases de datos, la cantidad de pares de parches en la base de datos fue de 577700 pares de parches en cada base de datos, el tamaño de parche se definió como $a = 2$ para el modelo basado en diccionarios y para $a = 3$ para las redes neuronales artificiales. Los parámetros para el algoritmo de Back-Projection fueron; un kernel gaussiano con $\sigma = 0.8$ usado como el kernel de back-projection B , la cantidad de iteraciones de 750 y una tolerancia de error de $\text{RMSE} < 0.0001$. Para la función de pesos definida en la Eq. (3.6) se usó el caso especial para el modelo basado en diccionarios, donde sólo se toma en cuenta el par de diccionarios tal que el centroide del grupo usado para entrenar ese par de diccionarios es el mas parecido, y para el modelo basado en redes neuronales artificiales se tomó $h = 12$. Se usó $\lambda = 0.1$ para el modelo basado en diccionarios, tanto para el entrenamiento como para la reconstrucción.

En esta Sección se muestran los resultados de diferentes algoritmos del estado del arte evaluados con un factor de escala $s = 2$ y $\sigma = 0.8$ para todos los algoritmos por igual. Para presentar los resultados se hace uso de tablas con código de color donde el color verde representa los valores más altas para PSNR o SSIM (mejor resultado) y el color rojo representa el valor más bajo. Es importante aclarar que la medida de PSNR no está delimitada y que siempre un PSNR mayor es mejor, de manera similar para SSIM con la diferencia de que los valores para esta medida están en el intervalo $[0, 1]$ donde, más cercano sea el resultado a 1 la calidad de la imagen estimada es mejor.

4.8.1.1. Base de Datos de Imágenes Naturales

A continuación en las Fig. 4.15 se muestran distintas imágenes donde cada imagen fue obtenida por un método de Single Image Super Resolution diferente. Después de cada figura se escribe un análisis de los resultados.

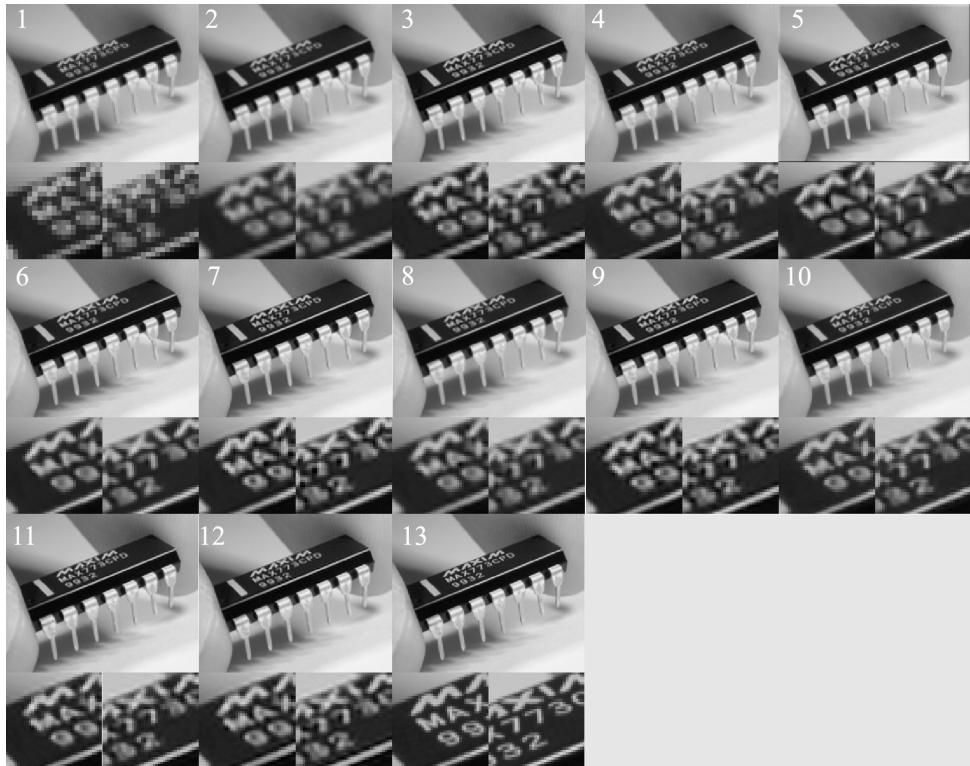


Figura 4.15: Comparación Entre Métodos del Estado del Arte - Microchip . De izquierda a derecha y de arriba a abajo; 1) Imagen de baja resolución (entrada de los algoritmos), 2) Interpolación Bicubic, 3) Irani and Peleg (1991), 4) Kim and Kwon (2010), 5) Freedman and Fattal (2010), 6) Glasner et al. (2009), 7) Dong et al. (2011), 8) Timofte et al. (2014), 9) Yang and Yang (2013), 10) Dong et al. (2014), 11) Modelo basado en diccionarios, 12) Modelo basado en redes neuronales artificiales, 13) Ground Truth o Imagen Real.

En la Fig 4.15 difícilmente se pueden apreciar grandes diferencias (sin hacer zoom) en los resultados de los diferentes algoritmos. Por este motivo las siguientes comparaciones se realizarán únicamente entre los dos métodos propuestos en este trabajo de tesis y 2 algoritmos más del estado del arte, esto con la finalidad de poder observar mejor los resultados de cada método y ver las ventajas y desventajas de cada uno de ellos con respecto a los métodos propuestos.



Figura 4.16: Comparación de Resultados Entre Algoritmos.
 Imagen “Dancers”.- A) Imagen real o ground truth. B) Imagen de baja resolución (entrada de los algoritmos). C) Imagen obtenida usando el modelo basado en redes neuronales artificiales. D) Imagen obtenida usando el modelo basado en diccionarios.

En la Fig. 4.16 se muestra el resultado de una de las imágenes listadas en la Tabla 4.4. El resultado se magnificó de tal forma que sea posible apreciar los píxeles de las imágenes para poder comparar los resultados y notar las fallas y aciertos de cada uno de los algoritmos. En este caso el resultado obtenido entre los modelos propuestos en esta tesis es muy similar tanto visualmente como en PSNR (ver Tabla 4.4). Ambos algoritmos logran recuperar detalles importantes como la cara de la bailarina (a la derecha de la imagen), mejoran la definición de los contornos en el automóvil (arriba a la derecha), de los vestidos de las bailarinas, etc.



Figura 4.17: Comparación de Resultados Entre Algoritmos.
Imagen “Dancers”- A) Interpolación bicubic. B) Irani and Peleg (1991). C) Modelo basado en redes neuronales artificiales. D) Modelo basado en diccionarios.

En la Fig. 4.17 se puede observar que el resultado de la interpolación bicubic (imagen A) es superado por todos los algoritmos en la imagen. Los bordes son menos borrosos y hay mas detalles en la imagen. El resultado de la imagen B aunque mejora la interpolación bicubic no es capaz de recuperar algunos detalles de la imagen y los bordes no son tan definidos como en las imágenes C y D. Es importante hacer notar que el algoritmo propuesto por Irani and Peleg (1991) (imagen B) es básicamente el paso de back-projection usado como post-proceso en los métodos propuestos en esta tesis. Sin embargo, debido al mal condicionamiento del problema de Single Image Super Resolution, el resultado del algoritmo de back-projection dependerá en gran medida de la imagen inicial. Por esta razón se

logra una mejor estimación con los métodos propuestos.



Figura 4.18: Comparación de Resultados Entre Algoritmos.
Imagen “Dancers”.- A) Freedman and Fattal (2010). B) Glasner et al. (2009). C) Modelo basado en redes neuronales artificiales. D) Modelo basado en diccionarios.

En la Fig. 4.18 se puede observar que el método propuesto por Freedman and Fattal (2010) genera bordes sobre definidos y además tiene problemas en áreas con textura como lo es el fondo detrás de las bailarinas. También tiene problemas recuperando información de la cara (con respecto a la imagen original) de la bailarina a la derecha de la imagen. El modelo propuesto por Glasner et al. (2009) genera mejores resultados y recupera algo de información de textura en el fondo, sin embargo, no recupera suficiente información en algunas zonas como lo es la cara de la bailarina a la derecha de la imagen. En general

este algoritmo funciona bien para el factor de escala 2x pero a mayores factores de escala comienza a generar un efecto de *acartonamiento* debido a los bordes sobre definidos que produce.



Figura 4.19: Comparación de Resultados Entre Algoritmos.
Imagen “Dancers”.- A) Kim and Kwon (2010). B) Timofte et al. (2014). C) Modelo basado en redes neuronales artificiales. D) Modelo basado en diccionarios.

En la Fig. 4.19 se puede observar que los resultados mostrados en la imagen A y en la imagen B parecen similares. Sin embargo, los resultados de la imagen B tienen ligeramente los bordes menos definidos y las intensidades de gris de la imagen parecen estar un tanto sesgado con respecto a la imagen original. Se puede notar que los resultados de las imágenes C y D en comparación con A y B recuperan mayor cantidad de detalles como lo

es la textura en el fondo, además los bordes están mas resaltados y las intensidades de gris tienen un sesgo menor con respecto a la imagen original.



Figura 4.20: Comparación de Resultados Entre Algoritmos.
Imagen “Dancers”. - A) Kim and Kwon (2010). B) Dong et al. (2011). C) Modelo basado en redes neuronales artificiales. D) Modelo basado en diccionarios.

En la Fig. 4.20 se puede observar que la imagen B obtenida con el algoritmo presentado por Dong et al. (2011) genera bordes bien definidos y recupera mejor las intensidades de gris en comparación con la imagen A. Sin embargo, tiene problemas en ciertas zonas y genera artefactos no deseados muy notables. Uno de los artefactos no deseados en la imagen B se puede observar en el vestido de la bailarina a la izquierda de la imagen, donde aparecen unas líneas horizontales blancas y una línea vertical negra que las atraviesa. Este

patrón aparece repetidamente en distintas imágenes probadas con este algoritmo.



Figura 4.21: Comparación de Resultados Entre Algoritmos.
Imagen “Dancers”.- A) Yang and Yang (2013). B) Dong et al. (2014). C) Modelo basado en redes neuronales artificiales. D) Modelo basado en diccionarios.

En la Fig. 4.21 se comparan dos de los algoritmos que tienen mejores resultados visuales y numéricos. El algoritmo propuesto por Yang and Yang (2013) es muy similar al generado por los modelos propuestos en este trabajo de tesis. Esto es debido a que hace uso del algoritmo de back-projection como post-proceso. La imagen B fue generada por un método basado en redes convolucionales. Este algoritmo es de los más rápidos en cuanto eficiencia computacional y además genera muy buenos resultados visuales, sin embargo, tiene problemas ya que genera imágenes con cierto sesgo en las intensidades de los píxeles.

En la Tabla 4.7 se evalúan diferentes algoritmos de Single Image Super Resolution. Los parámetros definidos para esta prueba son los siguientes, factor de escala $s = 2$ y valor del kernel gaussiano $\sigma = 0.8$. Se muestran los resultados de evaluar 47 imágenes diferentes.

En las Tablas 4.7 y 4.8 se puede observar que el modelo basado en diccionarios es bastante estable y aproximadamente el 40 % de los resultados son mejores utilizando este modelo, seguido del modelo G (Glasner et al., 2009) con mejores resultados en aprox. 30 % de las pruebas. Sin embargo, se puede observar fácilmente gracias a los colores en la tabla que el método B (nuestro modelo basado en diccionarios) es mas estable en sus resultados que el método G. En cuanto al método C (basado en redes neuronales artificiales) a pesar de sólo obtener el mejor resultado en el 10 % de las pruebas los resultados también parecen ser estables superando a varios métodos como lo son E, F y H.

En la Tabla 4.8 se muestra el resultado obtenido de las mismas pruebas usadas para la Tabla 4.7 pero en términos de SSIM. En promedio el modelo basado en diccionarios sigue generando los mejores resultados, sin embargo, los demás métodos no están muy lejos de los resultados obtenidos sobre todo el modelo basado en redes neuronales artificiales y el trabajo de Yang and Yang (2013) (columna I) están bastante cerca en promedio del modelo basado en diccionarios.

| IMG | A | B | C | D | E | F | G | H | I |
|------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| microchip | 30.54 | 35.59 | 34.73 | 35.27 | 34.64 | 32.58 | 36.57 | 34.1 | 35.1 |
| ocean | 30.43 | 31.48 | 31.56 | 31.64 | 31.48 | 31.28 | 31.5 | 31.44 | 31.6 |
| plane | 29.52 | 32.94 | 31.99 | 31.61 | 31.69 | 30.6 | 32.59 | 30.86 | 31.67 |
| woman_2 | 33.08 | 35.03 | 34.97 | 34.79 | 34.87 | 34.46 | 34.73 | 34.74 | 34.97 |
| sailing3 | 30.45 | 32.61 | 32.39 | 32.6 | 32.75 | 31.56 | 32.87 | 32.02 | 32.43 |
| sailing2 | 30.73 | 32.92 | 33.11 | 33.11 | 32.9 | 32.04 | 33.27 | 32.64 | 32.96 |
| carnivaldolls | 25.54 | 27.84 | 26.96 | 27.18 | 27.66 | 26.78 | 28.75 | 26.37 | 26.44 |
| cemetry | 24.45 | 26.67 | 26.54 | 26.65 | 26.37 | 25.79 | 27.06 | 25.86 | 26.09 |
| plant | 38.01 | 39.5 | 39.49 | 39.35 | 39.61 | 39.05 | 39.89 | 39.57 | 39.5 |
| squirrel | 23.25 | 24.63 | 24.52 | 24.67 | 24.36 | 24.2 | 24.22 | 24.2 | 24.49 |
| churchandcapitol | 24.91 | 26.8 | 26.87 | 26.78 | 26.66 | 26.07 | 27.27 | 26.18 | 26.8 |
| studentsculpture | 23.71 | 25.44 | 25.42 | 25.17 | 25.32 | 24.95 | 25.56 | 24.92 | 25.27 |
| statue | 30.4 | 32.34 | 32.26 | 32.31 | 32.22 | 31.52 | 32.27 | 31.9 | 32.09 |
| rapids | 28.08 | 30.05 | 30.16 | 30.26 | 29.98 | 29.33 | 29.82 | 29.8 | 30.09 |
| stream | 23.08 | 24.39 | 24.44 | 24.35 | 24.22 | 24.04 | 24.15 | 24.12 | 24.36 |
| bikes | 25.27 | 28.07 | 28.05 | 28.2 | 27.75 | 26.73 | 27.78 | 27.41 | 27.97 |
| coinsinfountain | 27.89 | 29.99 | 30.04 | 29.71 | 29.84 | 29.28 | 30.23 | 29.36 | 29.41 |
| girl_1 | 37.04 | 39.77 | 39.45 | 39.3 | 39.44 | 39.01 | 39.67 | 39.43 | 39.72 |
| text | 17.75 | 23.86 | 21.05 | 21.82 | 20.04 | 19.74 | 23.46 | 19.85 | 20.6 |
| buildings | 22.65 | 24.31 | 24.38 | 24.36 | 24.19 | 23.71 | 24.46 | 23.97 | 24.24 |
| woman_1 | 32.23 | 35.55 | 35.27 | 34.88 | 34.9 | 34.26 | 34.97 | 34.74 | 35.24 |
| man_3 | 34.34 | 37.17 | 36.93 | 36.67 | 36.67 | 36.27 | 36.72 | 36.5 | 36.97 |
| sailing4 | 28.27 | 29.94 | 29.99 | 30.12 | 29.88 | 29.29 | 29.91 | 29.66 | 29.98 |
| lena | 30.86 | 33.63 | 33.49 | 33.55 | 33.47 | 32.14 | 33.4 | 32.98 | 33.57 |
| oldMan_1 | 29.3 | 31.69 | 31.63 | 31.66 | 31.57 | 30.64 | 31.43 | 31.24 | 31.61 |
| oldWoman_1 | 26.66 | 29.14 | 28.64 | 28.86 | 28.5 | 27.94 | 28.73 | 28.16 | 28.29 |
| parrots | 33.91 | 36.91 | 36.65 | 36.77 | 36.49 | 35.64 | 36.65 | 36.29 | 36.84 |
| woman | 26.65 | 28.15 | 28.16 | 28.1 | 28.04 | 27.71 | 27.97 | 27.87 | 28.12 |
| sailing1 | 26.64 | 28.12 | 28.17 | 28.16 | 27.97 | 27.6 | 27.92 | 27.85 | 28.12 |
| lighthouse2 | 27.53 | 29.2 | 29.21 | 29.21 | 29.01 | 28.67 | 28.99 | 28.83 | 29.12 |
| face | 33.03 | 34.12 | 34.18 | 33.83 | 34.11 | 33.87 | 34.02 | 34.04 | 34.2 |
| flowersonih35 | 21.67 | 23.17 | 23.12 | 22.98 | 22.97 | 22.79 | 23.24 | 22.66 | 23.01 |
| caps | 33.24 | 35.24 | 35.35 | 35.35 | 35.36 | 34.37 | 35.35 | 35.18 | 35.5 |
| man_1 | 35.34 | 37.99 | 37.69 | 37.35 | 37.45 | 37.06 | 37.63 | 37.41 | 37.78 |
| womanhat | 32.2 | 34.16 | 34.09 | 33.96 | 33.93 | 33.42 | 34.13 | 33.77 | 33.99 |
| manfishing | 24.9 | 26.92 | 26.37 | 26.3 | 26.56 | 25.97 | 27.44 | 25.74 | 26.17 |
| man_2 | 35.21 | 37.45 | 37.2 | 37.06 | 37.08 | 36.75 | 37.0 | 36.92 | 37.22 |
| lighthouse | 26.57 | 28.31 | 28.33 | 28.51 | 28.0 | 27.61 | 28.51 | 27.89 | 28.21 |
| girl_2 | 29.16 | 31.16 | 31.03 | 31.17 | 31.0 | 30.41 | 30.85 | 30.77 | 30.96 |
| Paintedhouse | 25.86 | 27.56 | 27.52 | 27.39 | 27.27 | 26.92 | 27.34 | 27.08 | 27.38 |
| monarch | 30.78 | 35.38 | 35.35 | 35.1 | 34.4 | 32.58 | 35.65 | 33.81 | 34.6 |
| dancers | 23.03 | 24.93 | 24.75 | 24.74 | 24.63 | 24.09 | 25.04 | 24.21 | 24.65 |
| flowers | 26.78 | 28.58 | 28.58 | 28.51 | 28.4 | 28.05 | 28.33 | 28.23 | 28.53 |
| house | 29.64 | 31.25 | 31.4 | 31.28 | 31.19 | 30.77 | 31.28 | 31.01 | 31.45 |
| redHair | 30.22 | 31.99 | 32.01 | 31.93 | 31.98 | 31.39 | 31.68 | 31.74 | 32.05 |
| man_4 | 30.77 | 32.79 | 32.73 | 32.51 | 32.53 | 32.12 | 32.56 | 32.35 | 32.64 |
| building2 | 21.81 | 23.72 | 23.53 | 23.63 | 23.29 | 23.09 | 23.62 | 23.06 | 23.35 |
| promedio | 28.58 | 30.82 | 30.63 | 30.61 | 30.48 | 29.88 | 30.78 | 30.19 | 30.54 |

Máximo  Mínimo

Tabla 4.7: PSNR Obtenido con Diferentes Algoritmos de Single Image Super Resolution. De izquierda a derecha, los algoritmos evaluados son A) Bicubic, B) Modelo basado en diccionarios, C)Modelo basado en redes neuronales artificiales, D)Dong et al. (2014), E) Kim and Kwon (2010), F) Irani and Peleg (1991), G) Glasner et al. (2009), H) Timofte et al. (2014) y I) Yang and Yang (2013).

| IMG | A | B | C | D | E | F | G | H | I |
|------------------|-------|--------------|--------------|--------------|-------|-------|--------------|--------------|--------------|
| microchip | 0.962 | 0.982 | 0.977 | 0.983 | 0.98 | 0.973 | 0.986 | 0.981 | 0.981 |
| ocean | 0.833 | 0.879 | 0.878 | 0.874 | 0.871 | 0.871 | 0.874 | 0.87 | 0.879 |
| plane | 0.903 | 0.932 | 0.933 | 0.931 | 0.93 | 0.926 | 0.932 | 0.927 | 0.934 |
| woman_2 | 0.93 | 0.954 | 0.953 | 0.951 | 0.952 | 0.949 | 0.951 | 0.95 | 0.955 |
| sailing3 | 0.892 | 0.922 | 0.923 | 0.923 | 0.925 | 0.916 | 0.929 | 0.922 | 0.923 |
| sailing2 | 0.896 | 0.922 | 0.923 | 0.923 | 0.922 | 0.917 | 0.925 | 0.921 | 0.924 |
| carnivaldolls | 0.884 | 0.93 | 0.928 | 0.929 | 0.928 | 0.914 | 0.934 | 0.917 | 0.928 |
| cemetery | 0.81 | 0.885 | 0.886 | 0.879 | 0.873 | 0.869 | 0.877 | 0.867 | 0.882 |
| plant | 0.939 | 0.952 | 0.95 | 0.952 | 0.949 | 0.949 | 0.953 | 0.952 | 0.95 |
| squirrel | 0.815 | 0.87 | 0.869 | 0.866 | 0.862 | 0.861 | 0.857 | 0.856 | 0.87 |
| churchandcapitol | 0.85 | 0.898 | 0.898 | 0.897 | 0.892 | 0.885 | 0.898 | 0.886 | 0.898 |
| studentsculpture | 0.788 | 0.873 | 0.874 | 0.862 | 0.859 | 0.856 | 0.863 | 0.853 | 0.873 |
| statue | 0.904 | 0.935 | 0.934 | 0.934 | 0.933 | 0.927 | 0.934 | 0.931 | 0.935 |
| rapids | 0.832 | 0.893 | 0.894 | 0.889 | 0.886 | 0.881 | 0.885 | 0.882 | 0.893 |
| stream | 0.709 | 0.806 | 0.808 | 0.79 | 0.786 | 0.791 | 0.785 | 0.779 | 0.807 |
| bikes | 0.831 | 0.909 | 0.91 | 0.907 | 0.901 | 0.888 | 0.903 | 0.895 | 0.909 |
| coinsinfountain | 0.869 | 0.918 | 0.919 | 0.913 | 0.912 | 0.908 | 0.913 | 0.909 | 0.917 |
| girl_1 | 0.97 | 0.978 | 0.977 | 0.979 | 0.978 | 0.977 | 0.979 | 0.979 | 0.979 |
| text | 0.775 | 0.95 | 0.903 | 0.91 | 0.867 | 0.865 | 0.94 | 0.862 | 0.889 |
| buildings | 0.764 | 0.838 | 0.841 | 0.833 | 0.828 | 0.823 | 0.836 | 0.822 | 0.837 |
| woman_1 | 0.941 | 0.966 | 0.964 | 0.962 | 0.962 | 0.96 | 0.962 | 0.962 | 0.966 |
| man_3 | 0.951 | 0.968 | 0.966 | 0.966 | 0.966 | 0.965 | 0.965 | 0.965 | 0.968 |
| sailing4 | 0.823 | 0.877 | 0.878 | 0.874 | 0.871 | 0.865 | 0.873 | 0.867 | 0.877 |
| lena | 0.899 | 0.935 | 0.935 | 0.935 | 0.933 | 0.923 | 0.933 | 0.93 | 0.936 |
| oldMan_1 | 0.906 | 0.939 | 0.939 | 0.937 | 0.937 | 0.931 | 0.936 | 0.934 | 0.941 |
| oldWoman_1 | 0.855 | 0.908 | 0.904 | 0.901 | 0.902 | 0.895 | 0.898 | 0.895 | 0.904 |
| parrots | 0.946 | 0.96 | 0.959 | 0.961 | 0.96 | 0.958 | 0.961 | 0.96 | 0.961 |
| woman | 0.821 | 0.882 | 0.883 | 0.875 | 0.873 | 0.872 | 0.874 | 0.87 | 0.882 |
| sailing1 | 0.789 | 0.855 | 0.856 | 0.848 | 0.845 | 0.844 | 0.847 | 0.841 | 0.856 |
| lighthouse2 | 0.862 | 0.904 | 0.904 | 0.902 | 0.898 | 0.897 | 0.899 | 0.896 | 0.904 |
| face | 0.851 | 0.883 | 0.884 | 0.874 | 0.879 | 0.877 | 0.878 | 0.878 | 0.885 |
| flowersonih35 | 0.795 | 0.871 | 0.87 | 0.861 | 0.856 | 0.857 | 0.862 | 0.849 | 0.871 |
| caps | 0.911 | 0.938 | 0.936 | 0.936 | 0.936 | 0.93 | 0.938 | 0.936 | 0.938 |
| man_1 | 0.961 | 0.973 | 0.971 | 0.971 | 0.972 | 0.971 | 0.972 | 0.972 | 0.974 |
| womanhat | 0.879 | 0.916 | 0.915 | 0.912 | 0.91 | 0.909 | 0.912 | 0.909 | 0.913 |
| manfishing | 0.811 | 0.868 | 0.87 | 0.864 | 0.861 | 0.856 | 0.868 | 0.854 | 0.869 |
| man_2 | 0.959 | 0.972 | 0.97 | 0.971 | 0.97 | 0.97 | 0.97 | 0.97 | 0.972 |
| lighthouse | 0.824 | 0.877 | 0.877 | 0.874 | 0.867 | 0.866 | 0.873 | 0.865 | 0.877 |
| girl_2 | 0.925 | 0.948 | 0.947 | 0.948 | 0.948 | 0.944 | 0.946 | 0.946 | 0.95 |
| paintedhouse | 0.833 | 0.894 | 0.895 | 0.888 | 0.886 | 0.882 | 0.891 | 0.883 | 0.894 |
| monarch | 0.951 | 0.969 | 0.967 | 0.971 | 0.968 | 0.962 | 0.971 | 0.967 | 0.969 |
| dancers | 0.804 | 0.869 | 0.873 | 0.869 | 0.865 | 0.853 | 0.872 | 0.856 | 0.87 |
| flowers | 0.884 | 0.93 | 0.931 | 0.925 | 0.925 | 0.921 | 0.926 | 0.922 | 0.931 |
| house | 0.845 | 0.895 | 0.896 | 0.89 | 0.888 | 0.887 | 0.888 | 0.885 | 0.896 |
| redHair | 0.91 | 0.938 | 0.938 | 0.936 | 0.937 | 0.933 | 0.934 | 0.934 | 0.939 |
| man_4 | 0.905 | 0.933 | 0.933 | 0.93 | 0.93 | 0.929 | 0.929 | 0.928 | 0.933 |
| building2 | 0.773 | 0.865 | 0.863 | 0.855 | 0.845 | 0.846 | 0.851 | 0.839 | 0.861 |
| promedio | 0.866 | 0.913 | 0.912 | 0.909 | 0.906 | 0.902 | 0.909 | 0.903 | 0.912 |

Máximo  Mínimo

Tabla 4.8: SSIM Obtenido con Diferentes Algoritmos de Single Image Super Resolution. De izquierda a derecha, los algoritmos evaluados son A) Bicubic, B) Modelo basado en diccionarios, C)Modelo basado en redes neuronales artificiales, D)Dong et al. (2014), E) Kim and Kwon (2010), F) Irani and Peleg (1991), G) Glasner et al. (2009), H) Timofte et al. (2014) y I) Yang and Yang (2013).

4.8.1.2. Base de Datos de Imágenes de Texto

En este sección se realizaron pruebas con la base de datos *ICDAR2015* (Competition on Text Image Super-Resolution) formada por dos conjuntos de imágenes de texto, uno para entrenamiento y otro para pruebas. Para cada conjunto de la base de datos se conocen las imágenes de alta resolución y las de baja resolución. De manera similar a experimentos anteriores se usó un factor de escala $s = 2$, sin embargo, en esta ocasión el valor de σ es desconocido. Para esta clase de imágenes se probó con el modelo de redes neuronales artificiales y diccionarios entrenados con imágenes naturales y con imágenes de texto. Se puede observar en los resultados que si los parámetros de cada modelo son estimados usando un conjunto específico de imágenes los resultados pueden mejorar considerablemente (ver Tabla 4.9).

Los parámetros usadas en esta Sección son los mismos que para la Sección de experimentos de imágenes naturales por un factor de escala de 2x.

La base de datos de prueba está formada por un conjunto de 141 imágenes de texto, sin embargo, sólo se muestran los resultados para 43 imágenes del conjunto (escogidas de manera aleatoria). En la Tabla 4.9 se muestran los resultados obtenidos por distintos métodos, sin embargo, se puede observar que los mejores resultados son generados por los métodos propuestos en esta tesis entrenados con imágenes de texto (misma clase). Esto apoya la idea de que usar una clase específica de imágenes para la estimación de los parámetros ayuda a obtener mejores resultados.

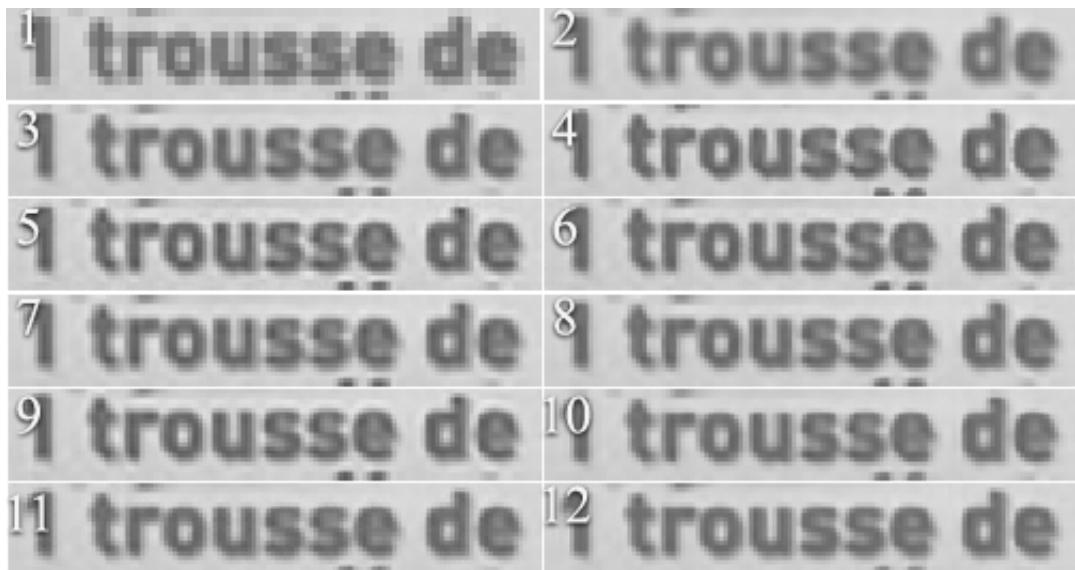


Figura 4.22: Comparación de imágenes de texto obtenidas por diferentes algoritmos.
 De arriba para abajo y de izquierda a derecha. 1.- Imagen de baja resolución, 2.- Inpoterlación bicubic, 3.- Timofte et al. (2014), 4.- Glasner et al. (2009), 5.- Yang and Yang (2013), 6.- Modelo basado en redes neuronales artificiales, 7.- Irani and Peleg (1991), 8.- Kim and Kwon (2010), 9.- Dong et al. (2011), 10.- Dong et al. (2014), 11.- Modelo basado en diccionarios, 12.- Imagen real (ground truth).

| | | | |
|----|--------|--------------|---------|
| 1 | HEURES | J.-L. BORLOO | ► 25 AI |
| 2 | HEURES | J.-L. BORLOO | ► 25 AI |
| 3 | HEURES | J.-L. BORLOO | ► 25 AI |
| 4 | HEURES | J.-L. BORLOO | ► 25 AI |
| 5 | HEURES | J.-L. BORLOO | ► 25 AI |
| 6 | HEURES | J.-L. BORLOO | ► 25 AI |
| 7 | HEURES | J.-L. BORLOO | ► 25 AI |
| 8 | HEURES | J.-L. BORLOO | ► 25 AI |
| 9 | HEURES | J.-L. BORLOO | ► 25 AI |
| 10 | HEURES | J.-L. BORLOO | ► 25 AI |
| 11 | HEURES | J.-L. BORLOO | ► 25 AI |
| 12 | HEURES | J.-L. BORLOO | ► 25 AI |

Figura 4.23: Comparación de imágenes de texto obtenidas por diferentes algoritmos.
 De arriba para abajo. 1.- Imagen de baja resolución, 2.- Inpoterlación bicubic, 3.- Irani and Peleg (1991), 4.- Timofte et al. (2014) , 5.- Dong et al. (2011) , 6.- Yang and Yang (2013) , 7.- Glasner et al. (2009), 8.- Kim and Kwon (2010), 9.- Dong et al. (2014) , 10.- Modelo basado en redes neuronales artificiales, 11.- Modelo basado en diccionarios, 12.- Imagen real (ground truth).

En las Figuras Fig. 4.22 y Fig. 4.23 se pueden observar resultados para 2 imágenes de texto diferentes. Estos resultados fueron obtenidos por diferentes algoritmos de Single Image Super Resolution. En la Fig. 4.22 queda claro que la interpolación bicubic no genera el mejor resultado, el resultado de la imagen 4 (Timofte et al., 2014) es demasiado borroso, los resultados para las imágenes 5 (Yang and Yang, 2013), 7 (Irani and Peleg, 1991) y 9 (Dong et al., 2011) parecen generar bordes demasiado definidos y los métodos restantes generan resultados muy similares entre si. En la Fig. 4.23 se pueden observar mayores diferencias entre los diferentes resultados. Una de las diferencias más significativas está en la palabra “HEURES” que está mejor definida en las últimas imágenes (9,10 y 11), otra diferencia se encuentra en la separación entre la letras O al final de la palabra “BORLOO” en la mayoría de las imágenes estas letras parecen estar pegadas. En general, estos resultados visuales y el PSNR obtenido muestra que los modelos propuestos en este trabajo de tesis (previamente entrenados con imágenes de texto) hacen una buena estimación de imágenes de texto.

| IMG | A | B | C | D | E | F | G | H | I | J | K |
|----------|-------|-------|-------|--------------|--------------|-------|--------------|-------|-------|--------------|--------------|
| Img 1 | 20.08 | 23.3 | 21.37 | 23.14 | 23.15 | 22.69 | 27.03 | 23.79 | 23.15 | 26.7 | 28.27 |
| Img 2 | 19.37 | 22.05 | 20.57 | 22.08 | 21.4 | 21.08 | 22.09 | 22.58 | 21.97 | 25.19 | 25.48 |
| Img 3 | 25.61 | 28.94 | 26.74 | 25.8 | 29.42 | 27.14 | 29.1 | 30.05 | 27.99 | 30.35 | 30.75 |
| Img 4 | 23.98 | 28.5 | 24.2 | 27.07 | 27.59 | 25.43 | 31.48 | 29.26 | 27.51 | 30.98 | 33.23 |
| Img 5 | 17.32 | 18.8 | 17.82 | 19.77 | 18.93 | 18.62 | 20.97 | 19.19 | 19.38 | 21.96 | 21.9 |
| Img 6 | 17.25 | 19.92 | 18.43 | 19.35 | 19.64 | 19.24 | 21.66 | 19.62 | 19.64 | 25.02 | 25.04 |
| Img 7 | 32.45 | 35.45 | 33.18 | 33.04 | 34.82 | 31.69 | 35.01 | 34.65 | 33.2 | 35.26 | 36.61 |
| Img 8 | 22.61 | 28.04 | 23.77 | 27.9 | 27.3 | 26.31 | 29.14 | 28.42 | 26.72 | 29.58 | 30.74 |
| Img 9 | 24.28 | 26.54 | 24.72 | 25.61 | 26.7 | 25.22 | 27.0 | 26.68 | 25.94 | 26.94 | 26.08 |
| Img 10 | 19.41 | 21.54 | 20.23 | 22.48 | 21.79 | 21.12 | 24.32 | 21.42 | 22.04 | 25.45 | 25.68 |
| Img 11 | 22.25 | 26.68 | 23.21 | 27.54 | 25.49 | 24.75 | 27.61 | 27.72 | 26.32 | 29.34 | 30.56 |
| Img 12 | 18.98 | 20.71 | 20.05 | 20.08 | 21.04 | 20.82 | 22.13 | 20.75 | 20.82 | 22.05 | 22.24 |
| Img 13 | 23.92 | 26.43 | 25.06 | 25.63 | 26.55 | 25.7 | 27.22 | 27.05 | 26.28 | 29.09 | 27.53 |
| Img 14 | 23.29 | 27.3 | 24.66 | 24.81 | 27.45 | 26.34 | 27.66 | 27.32 | 25.68 | 28.28 | 28.59 |
| Img 15 | 21.68 | 26.27 | 23.13 | 25.63 | 24.75 | 22.76 | 27.2 | 27.47 | 25.17 | 28.88 | 30.29 |
| Img 16 | 22.43 | 27.2 | 23.57 | 27.45 | 26.91 | 25.93 | 28.51 | 28.08 | 26.17 | 29.44 | 30.79 |
| Img 17 | 20.06 | 21.96 | 20.72 | 23.18 | 21.99 | 21.62 | 23.63 | 22.61 | 21.97 | 27.71 | 25.71 |
| Img 18 | 18.26 | 21.41 | 19.43 | 21.7 | 21.53 | 21.5 | 23.62 | 20.57 | 20.77 | 25.91 | 26.63 |
| Img 19 | 21.66 | 25.94 | 23.29 | 21.41 | 23.7 | 21.69 | 21.28 | 26.77 | 23.47 | 28.01 | 27.13 |
| Img 20 | 17.29 | 19.18 | 18.21 | 21.04 | 18.89 | 18.85 | 22.07 | 19.28 | 19.86 | 22.59 | 23.12 |
| Img 21 | 32.75 | 37.48 | 33.53 | 35.4 | 36.56 | 33.97 | 37.95 | 35.44 | 35.37 | 36.36 | 37.28 |
| Img 22 | 19.1 | 22.23 | 19.68 | 25.41 | 21.65 | 21.04 | 24.28 | 21.56 | 21.64 | 23.61 | 24.39 |
| Img 23 | 18.79 | 21.97 | 19.86 | 20.38 | 21.56 | 20.73 | 24.12 | 21.52 | 21.27 | 26.66 | 27.79 |
| Img 24 | 19.58 | 21.91 | 20.49 | 22.61 | 21.89 | 20.99 | 23.63 | 22.45 | 22.43 | 27.25 | 26.81 |
| Img 25 | 17.91 | 20.63 | 19.02 | 20.3 | 20.93 | 20.23 | 22.52 | 20.36 | 20.27 | 25.0 | 23.66 |
| Img 26 | 31.38 | 35.6 | 31.81 | 32.51 | 35.15 | 32.0 | 36.03 | 34.19 | 33.71 | 33.75 | 34.72 |
| Img 27 | 17.88 | 20.25 | 18.85 | 21.72 | 19.57 | 18.37 | 21.09 | 19.84 | 20.58 | 22.62 | 23.02 |
| Img 28 | 23.94 | 28.5 | 24.14 | 26.06 | 27.59 | 25.69 | 31.88 | 29.83 | 28.0 | 31.54 | 33.63 |
| Img 29 | 23.87 | 28.94 | 25.11 | 26.81 | 27.19 | 25.13 | 26.06 | 29.73 | 26.38 | 30.15 | 29.85 |
| Img 30 | 25.56 | 28.16 | 26.48 | 25.66 | 30.21 | 27.14 | 26.91 | 27.85 | 25.97 | 26.14 | 27.65 |
| Img 31 | 26.29 | 28.43 | 27.86 | 29.23 | 28.36 | 28.01 | 30.21 | 29.58 | 28.41 | 29.87 | 29.37 |
| Img 32 | 18.68 | 21.04 | 19.7 | 22.36 | 21.18 | 20.96 | 23.08 | 21.73 | 21.24 | 24.7 | 24.67 |
| Img 33 | 23.41 | 27.4 | 23.64 | 25.14 | 27.04 | 24.91 | 30.87 | 28.82 | 26.45 | 31.67 | 33.3 |
| Img 34 | 22.34 | 26.88 | 23.45 | 27.93 | 26.6 | 25.43 | 28.18 | 27.75 | 25.95 | 29.35 | 30.47 |
| Img 35 | 24.11 | 27.36 | 25.37 | 26.58 | 29.34 | 26.63 | 25.41 | 26.62 | 25.11 | 25.36 | 27.36 |
| Img 36 | 18.53 | 21.2 | 19.82 | 21.72 | 21.01 | 20.43 | 22.52 | 20.47 | 20.86 | 25.35 | 24.82 |
| Img 37 | 22.1 | 26.5 | 23.42 | 25.88 | 25.45 | 24.59 | 25.74 | 27.89 | 25.38 | 30.25 | 28.77 |
| Img 38 | 23.51 | 29.12 | 24.39 | 26.9 | 28.06 | 25.93 | 28.18 | 28.63 | 26.34 | 29.14 | 29.03 |
| Img 39 | 20.32 | 21.56 | 20.97 | 21.56 | 21.67 | 21.36 | 22.81 | 21.76 | 21.93 | 23.16 | 22.67 |
| Img 40 | 22.4 | 27.43 | 23.39 | 26.79 | 26.44 | 25.51 | 28.59 | 27.97 | 26.19 | 30.1 | 30.72 |
| Img 41 | 24.08 | 28.37 | 24.37 | 27.23 | 27.86 | 25.93 | 31.69 | 29.53 | 27.79 | 31.12 | 33.06 |
| Img 42 | 23.9 | 28.36 | 24.13 | 26.81 | 27.36 | 25.43 | 31.63 | 29.3 | 27.38 | 31.68 | 33.75 |
| Img 43 | 19.9 | 23.65 | 21.55 | 24.33 | 23.74 | 23.58 | 26.08 | 24.38 | 23.62 | 27.34 | 27.52 |
| promedio | 22.15 | 25.56 | 23.1 | 24.98 | 25.24 | 24.01 | 26.7 | 25.83 | 24.8 | 27.93 | 28.39 |

Máximo Mínimo

Tabla 4.9: Resultados en términos de PSNR para imágenes de texto obtenidos por diferentes métodos de Single Image Super Resolution.

A) Interpolación bicubic, B) Kim and Kwon (2010), C) Irani and Peleg (1991), D) Glasner et al. (2009), E) Timofte et al. (2014), F) Yang and Yang (2013), G) Dong et al. (2014), H) e I) Modelos basados en redes neuronales artificiales y diccionarios respectivamente entrenados con imágenes naturales, J) e K) Modelos basados en redes neuronales artificiales y diccionarios respectivamente entrenados con imágenes de texto. Los demás algoritmos no fueron entrenados con una clase en específico.

4.8.2. Factor de escala 4x

En esta Sección se muestran algunos resultados visuales (ya que no se cuenta con la imagen real) de diferentes algoritmos de Single Image Super Resolution para un factor de escala de 4x. Una razón importante para realizar estos experimentos es que la mayoría de los algoritmos comienzan a generar artefactos no deseados que se pueden observar más fácilmente en factores de escala mayores a 2x. Además, puede notarse con mayor facilidad que algoritmos logran recuperar mayor cantidad de detalles.

Para los experimentos mostrados en esta Sección se construyeron 2 bases de datos (una para cada modelo propuesto) con las siguientes características; Los parámetros para el modelo de degradación (o adquisición) se definieron como $s = 4$ y $\sigma = 1.6$ para ambas bases de datos, la cantidad de pares de parches en la base de datos fue de aproximadamente 700000 pares de parches en cada base de datos, el tamaño de parche se definió como $a = 2$ para el modelo basado en diccionarios y para $a = 2$ para las redes neuronales artificiales. Los parámetros para el algoritmo de Back-Projection fueron; un kernel gaussiano con $\sigma = 1.6$ usado como el kernel de back-projection B , la cantidad de iteraciones de 250 y una tolerancia de error de $\text{RMSE} < 0.0001$. Para la función de pesos definida en la Eq. (3.6) se usó el caso especial para el modelo basado en diccionarios, donde sólo se toma en cuenta el par de diccionarios tal que el centroide del grupo usado para entrenar ese par de diccionarios es el mas parecido, y para el modelo basado en redes neuronales artificiales se tomó $h = 12$. Se usó $\lambda = 0.1$ para el modelo basado en diccionarios, tanto para el entrenamiento como para la reconstrucción.

En las Figuras Fig. 4.24, Fig. 4.25, Fig. 4.26 y Fig. 4.27 se muestran algunas imágenes resultados de diferentes algoritmos del estado del arte. Cada una de las imágenes mostradas está referenciada por una letra que representa el algoritmo con el que se obtuvo esa imagen, la relación entre las letras y el algoritmo está definida en el texto debajo de cada Figura. Para cada Figura se detallan observaciones sobre los resultados.

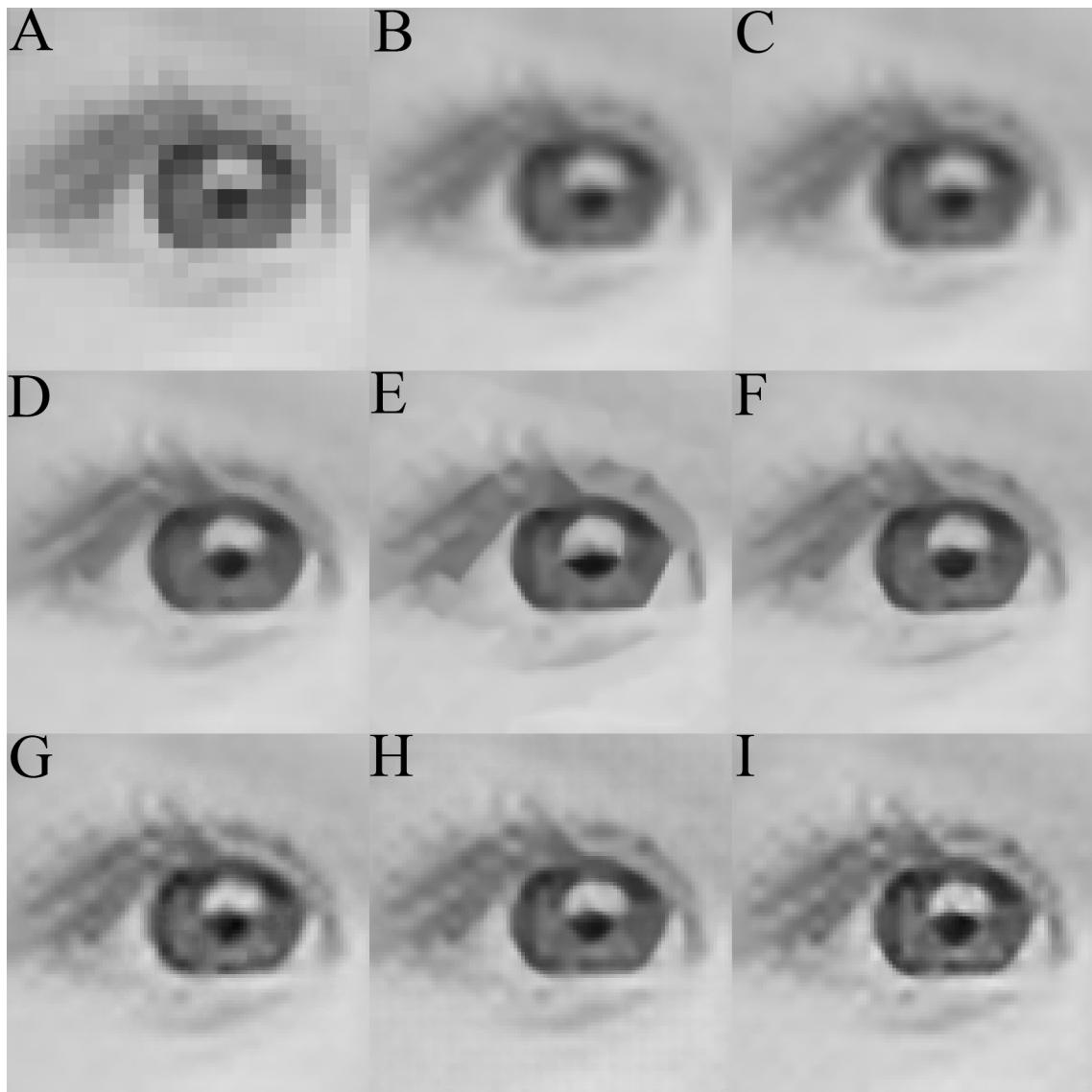


Figura 4.24: Comparación entre diferentes algoritmos. Imagen de un ojo de un bebe. A) Imagen de baja resolución, B) Interpolación bicubic, C) Irani and Peleg (1991), D) Kim and Kwon (2010), E) Glasner et al. (2009), F) Dong et al. (2014), G) Yang and Yang (2013), H) Modelo basado en redes neuronales artificiales e I) Modelo basado en diccionarios.

En la Fig. 4.24 se puede observar que los resultados de la imagen B y C no logran recuperar gran cantidad de detalles, además estos resultados tienen los bordes borrosos. La imagen E por el contrario tiene bordes duros que llegan a perder suavidad y forma, de manera que, para líneas que son curvas las comienza a convertir en una especie de líneas rectas sobre definidas (duras). Las imágenes D y F logran recuperar mejor los bordes, sin embargo, en la zona de las pestañas no logran recuperar demasiada información como lo logran las imágenes G,H e I. Por otro lado, existe un problema notable con las imágenes G e I (especialmente I) y es que parecen generar una serie de líneas en la parte superior derecha del ojo (en la zona de las pestañas) que no existen en realidad en la imagen. Estos

artefactos son introducidos por el uso de la interpolación bicubic como aproximación a la solución inicial y pueden observarse ligeramente en la imagen B y C.

En la Fig. 4.25 se puede observar que los resultados son muy similares entre sí, sin embargo, existe un factor en común para todos los resultados, el forraje de los árboles no se recupera correctamente. Recuperar la información del forraje de los árboles es una tarea complicada para cualquiera de los algoritmos actuales de Single Image Super Resolution, debido a que es una zona con textura formada por gradientes altos a través de los bordes y resulta complicado recuperar esta información. Un posible trabajo a futuro inspirado por estos problemas en los algoritmos, es entrenar distintos conjuntos de parámetros usando distintas clases de texturas, de manera que sea posible identificar cual conjunto de parámetros es el más adecuado acorde a la textura de un cierto parche de baja resolución. Entre los mejores resultados (visuales) se encuentran las imágenes G, H e I que logran recuperar más información en la zona del forraje de los árboles.

En la Fig. 4.26 se pueden observar algunas diferencias entre los diferentes resultados. Las imágenes B y C contienen bordes borrosos en comparación con los demás resultados. Los algoritmos propuestos en este trabajo de tesis (imagen H e I) parecen generar buenos resultados teniendo buena definición y forma en las letras, sin embargo, existe una especie de línea negra sobre la línea blanca a lo largo del microchip. Esto es un artefacto no deseado generado por el algoritmo.

En la Fig. 4.27 se puede observar nuevamente que el algoritmo que generó la imagen F (Glasner et al., 2009) genera bordes sobre definidos (duros) y tiende a provocar un efecto de “caricaturización” en la imagen resultante. Las imágenes B, C y D generan resultados con bordes un poco borrosos, por otro lado, las imágenes E, G,H e I parecen generar buenos resultados. En esta imagen en particular (tomada con un celular de 3 mega píxeles) se puede notar el efecto de “edge halo” al rededor de los bordes entre el cabello y el fondo gris. Esto es un artefacto no deseado que se busca disminuir en los resultados y queda como trabajo a resolver en futuras investigaciones.

En la Fig. 4.27 se puede observar uno de los problemas que presenta el algoritmo basado en diccionarios y es que, debido a la interpolación bicubic (imagen B) que es usada como estimación inicial, el modelo basado en diccionarios no es capaz de recuperar de manera correcta algunas estructuras de la imagen, por ejemplo, las antenas de la mariposa y algunas otras estructuras similares. El modelo basado en redes neuronales artificiales tiene este mismo problema, pero logra recuperar mejorar este tipo de estructuras. Este ti-



Figura 4.25: Comparación entre diferentes algoritmos. Imagen de una construcción entre árboles.

A) Imagen de baja resolución, B) Interpolación bicubic, C) Irani and Peleg (1991), D) Dong et al. (2014), E) Dong et al. (2011), F) Kim and Kwon (2010), G) Yang and Yang (2013), H) Modelo basado en redes neuronales artificiales e I) Modelo basado en diccionarios.

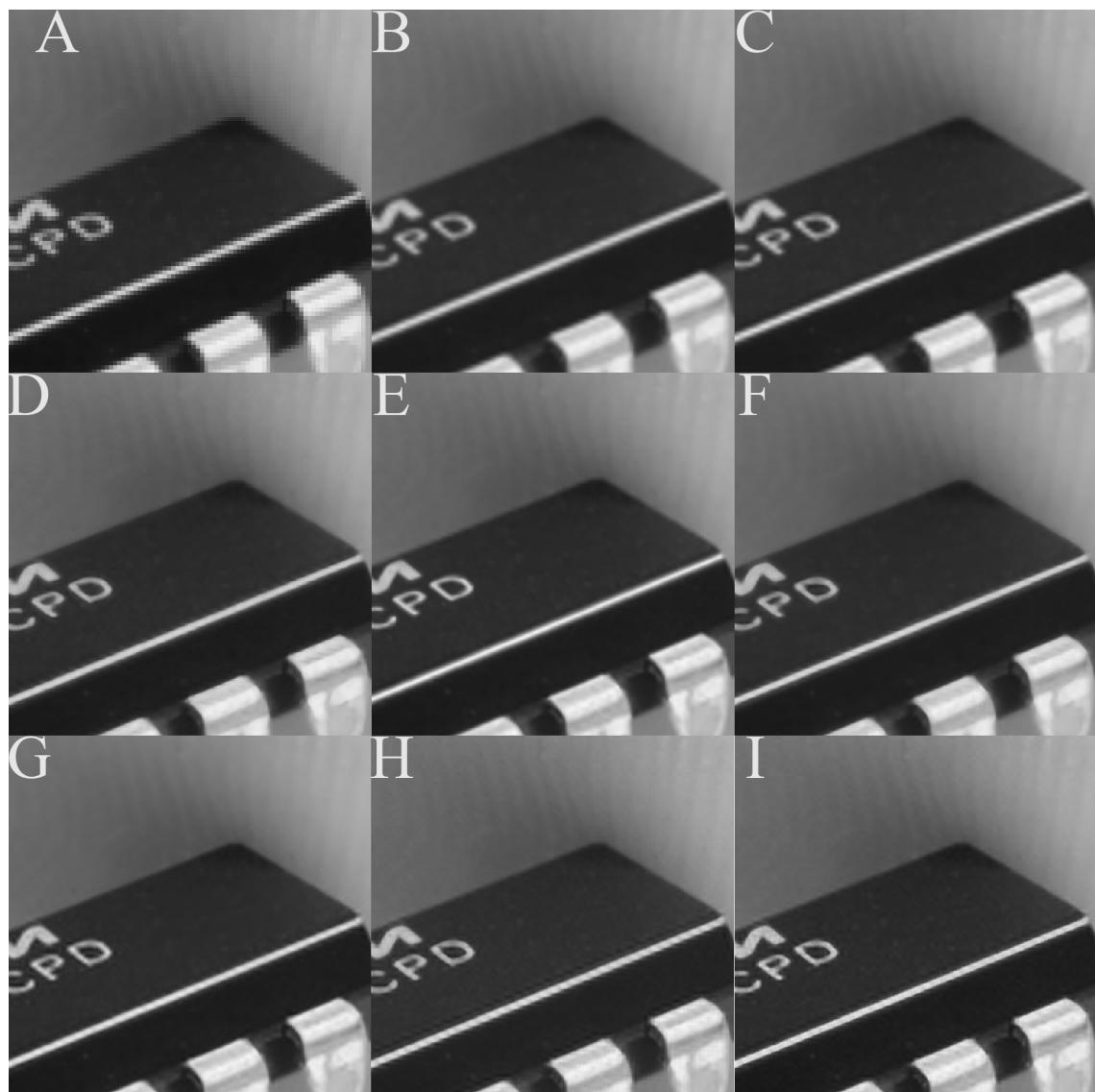


Figura 4.26: Comparación entre diferentes algoritmos. Imagen de un microchip.
A) Imagen de baja resolución, B) Interpolación bicubic, C) Irani and Peleg (1991), D) Dong et al. (2014), E) Dong et al. (2011), F) Kim and Kwon (2010), G) Yang and Yang (2013), H) Modelo basado en redes neuronales artificiales e I) Modelo basado en diccionarios.



Figura 4.27: Comparación entre diferentes algoritmos. Imagen de la cara de un adolescente.
A) Imagen de baja resolución, B) Interpolación bicubic, C) Irani and Peleg (1991), D)Dong et al. (2014), E) Dong et al. (2011) , F)Glasner et al. (2009) , G) Yang and Yang (2013), H) Modelo basado en redes neuronales artificiales e I) Modelo basado en diccionarios.

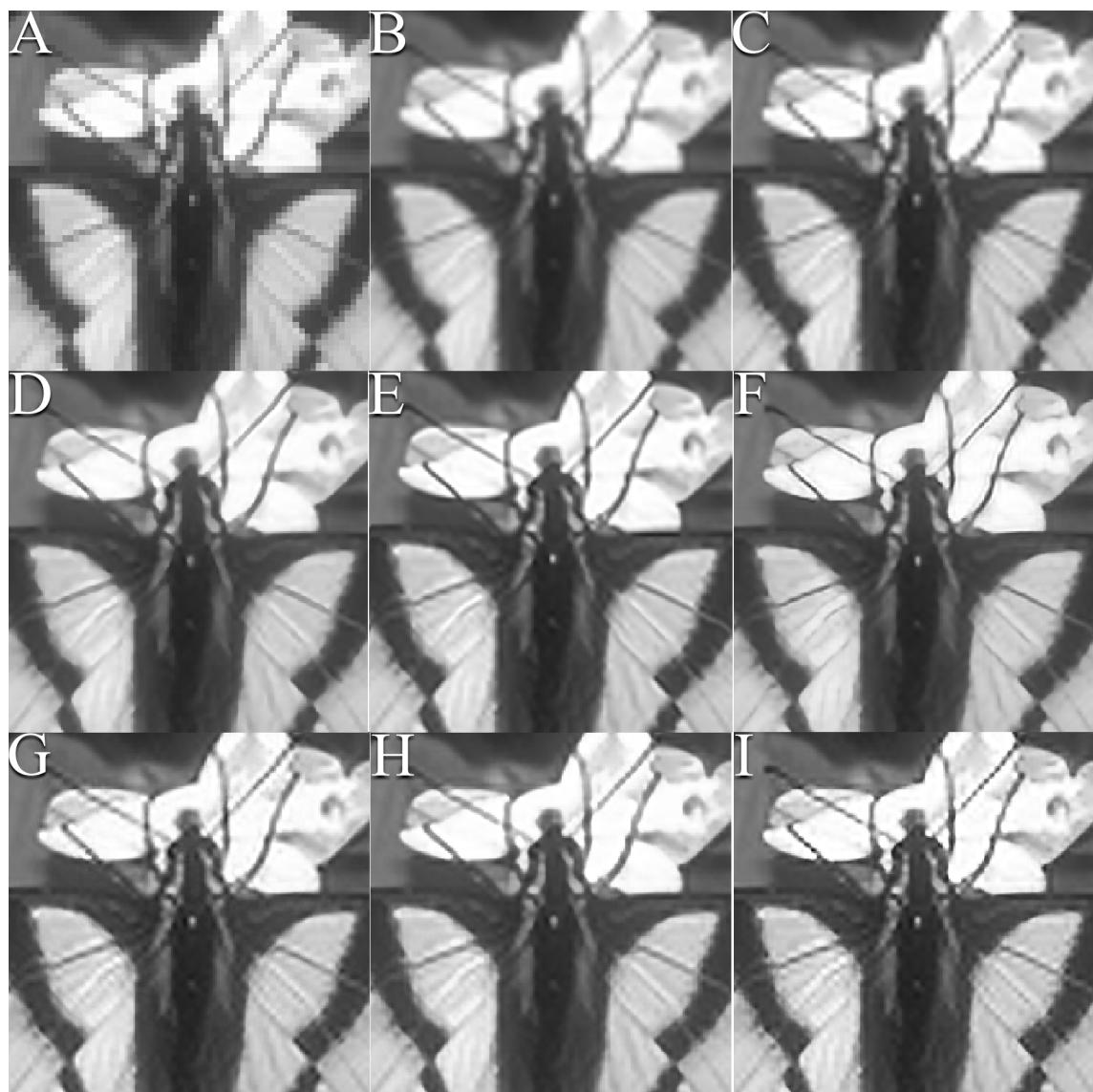


Figura 4.28: Comparación entre diferentes algoritmos. Imagen de una mariposa.
A) Imagen de baja resolución, B) Interpolación bicubic, C) Irani and Peleg (1991), D)
Kim and Kwon (2010), E) Glasner et al. (2009), F) Dong et al. (2014), G) Yang and Yang
(2013), H) Modelo basado en redes neuronales artificiales e I) Modelo basado en
diccionarios.

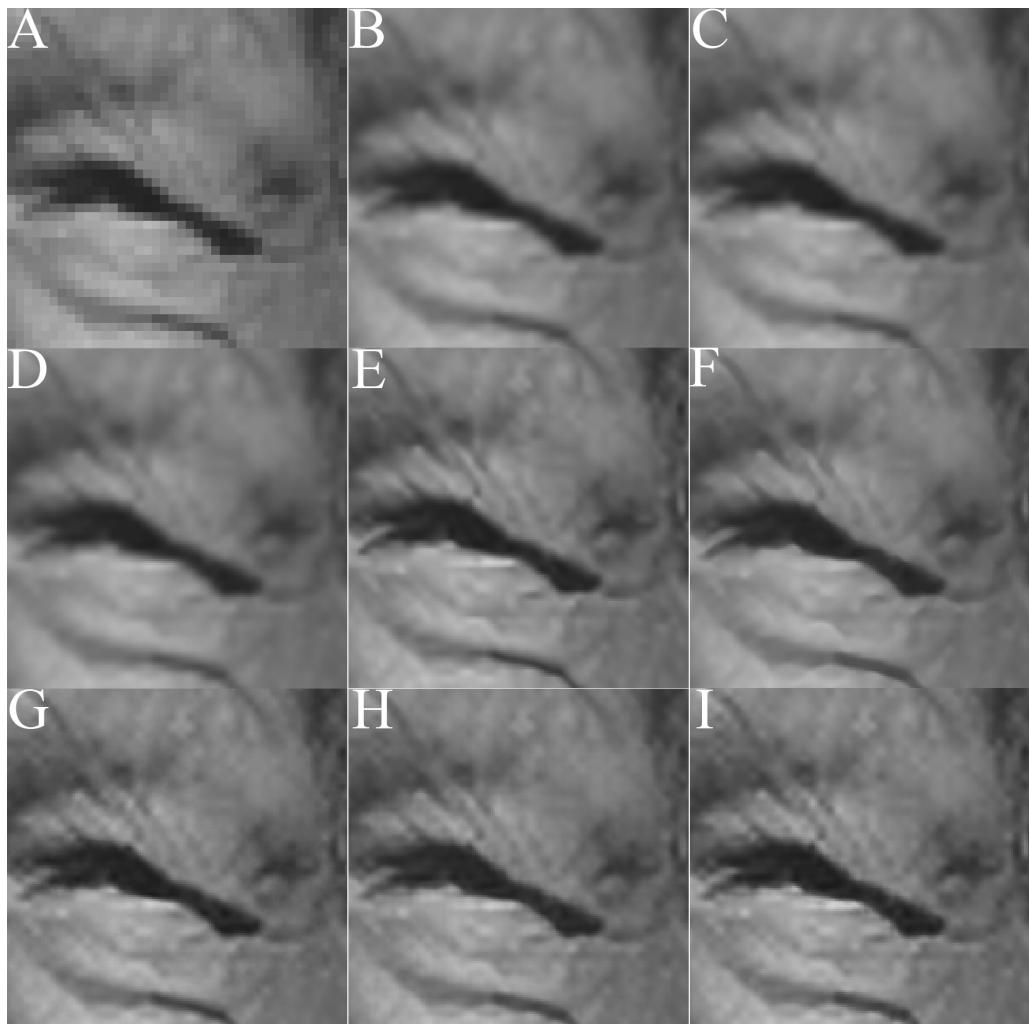


Figura 4.29: Comparación entre diferentes algoritmos. Imagen del ojo de un adulto mayor.
 A) Imagen de baja resolución, B) Interpolación bicubic, C) Irani and Peleg (1991), D)
 Dong et al. (2014), E) Dong et al. (2011), F) Kim and Kwon (2010), G) Yang and Yang
 (2013), H) Modelo basado en redes neuronales artificiales e I) Modelo basado en
 diccionarios.

po de problemas es algo en que se debe continuar trabajando y queda como trabajo a futuro.

En las Figuras Fig. 4.29 y Fig. 4.30 se pueden observar una imagen del ojo de un adulto mayor y otra de un edificio respectivamente. Son imágenes donde los métodos propuestos logran recuperar bien algunos detalles de alta frecuencia como se puede observar en la imagen del ojo del adulto mayor. Sin embargo, en la imagen del edificio se puede observar el efecto de edge halo producido por la interpolación bicubic y amplificado por el método de Single Image Super Resolution propuesto en este trabajo de tesis. Este problema aparece en algunas imágenes y queda abierta la investigación para tratar de reducir este efecto.

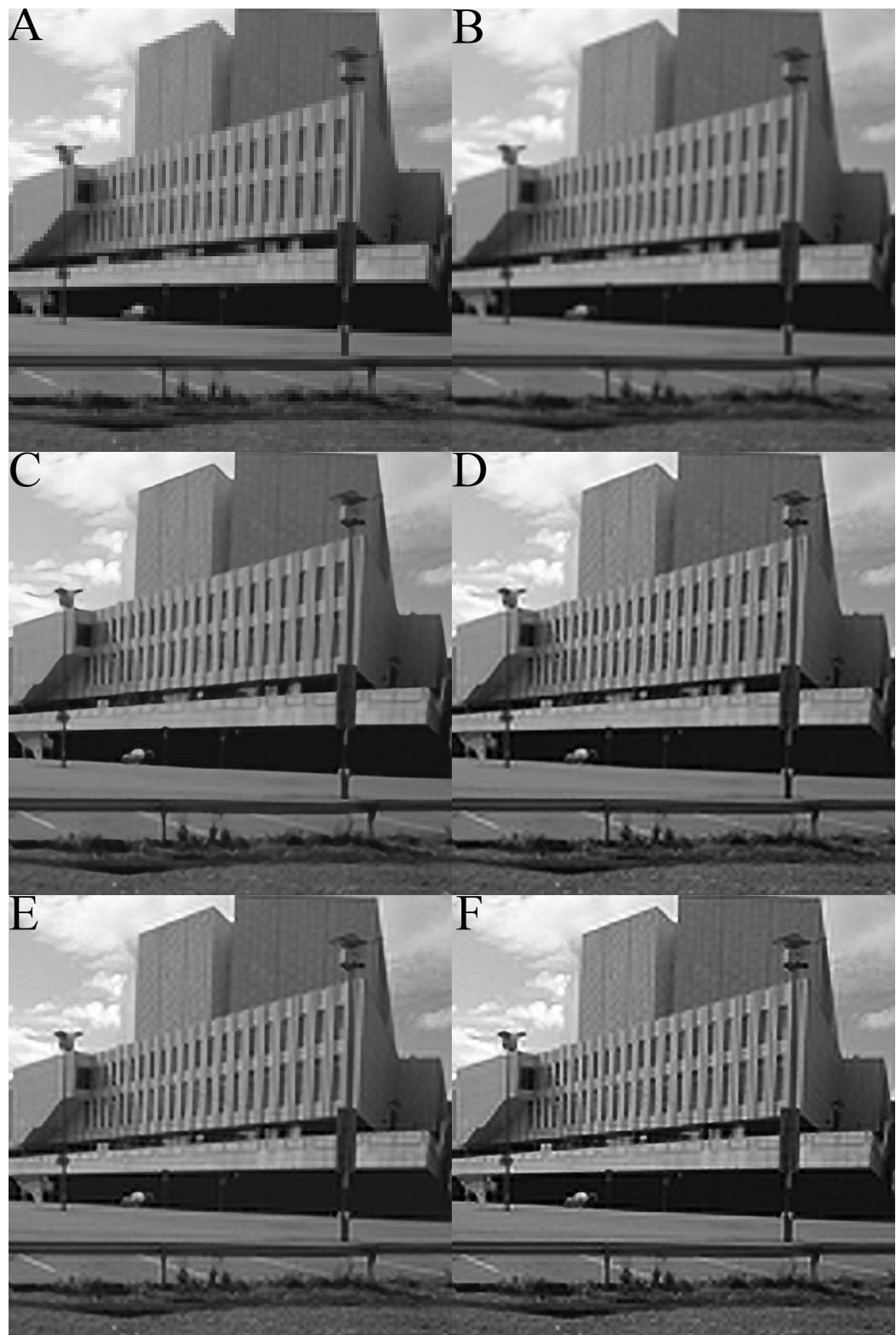


Figura 4.30: Comparación entre diferentes algoritmos. Imagen de un edificio.
A) Imagen de baja resolución, B) Interpolación bicubic, C) Dong et al. (2014), D) Yang and Yang (2013), E) Modelo basado en redes neuronales artificiales y F) Modelo basado en diccionarios.

Capítulo 5

Conclusiones y Trabajo Futuro

5.1. Conclusiones

A manera de un breve resumen sobre el trabajo presentado a lo largo de este documento y con el fin de resaltar los resultados obtenidos en este trabajo de tesis, se introduce esta Sección de conclusiones.

A lo largo del Capítulo 3, se presentó un esquema general de Single Image Super Resolution para el cual se propusieron dos modelos matemáticos diferentes para abordar el problema. El primer modelo está basado en un conjunto de redes neuronales artificiales que, a saber, es uno de los pocos trabajos realizados en este tema que hacen uso de esta técnica de aprendizaje automatizado. El segundo modelo está basado en un conjunto de pares de diccionarios, donde un diccionario es estimado con parches de baja resolución y otro con parches de alta resolución donde, a diferencia del primer modelo, ya se han presentado varios trabajos basados en propuestas similares.

Uno de los aportes en este trabajo de tesis, son los estudios realizados sobre el agrupamiento y como ayuda este a mejorar los resultados obtenidos. Los modelos propuestos en esta tesis hacen uso directamente del agrupamiento para aprovechar la información redundante a nivel de parches presente en las imágenes naturales. Esta es una propuesta que no es comúnmente vista en los trabajos del estado del arte. Aunque, existen algunos trabajos que hacen uso del agrupamiento (en investigaciones recientes), son pocos a saber, los que muestran algún soporte analítico y/o empírico (como lo es el caso de esta tesis) sobre el aporte o ventajas que se obtienen al agrupar los parches. Como conclusión sobre el uso del agrupamiento en el esquema de Super Resolution, se puede decir que efectivamente ayuda a mejorar los resultados obtenidos y que aún queda mucho trabajo por hacer en relación a este tema, como probar otras técnicas de agrupamiento y/o medidas de distancia (en el caso de K-means).

El modelo basado en redes neuronales artificiales está inspirado en una idea muy simple, aprender la transformación de un parche de baja resolución a alta resolución. La manera en que esta transformación es aprendida es mediante el uso de parches de baja y alta resolución como datos de entrenamiento del conjunto de redes neuronales artificiales. A pesar de que la idea es muy simple, los resultados obtenidos son competitivos con varios algoritmos del estado del arte y debido a que evaluar una red neuronal artificial es sencillo y barato computacionalmente (dependiendo de la cantidad de neuronas en la red) resulta ser un área prometedora para explorar más a fondo en un futuro. Los resultados obtenidos con este modelo a pesar de no ser los mejores, resultaron ser competitivos con varios algoritmos del estado del arte y además por la forma en que está planteado el modelo, es posible realizar una implementación que sea muy eficiente en términos computacionales. Es importante resaltar que durante la transformación que se realiza a los parches (centrarlos alrededor de cero restando la media) se propone restar la media del parche de baja resolución tanto a los parches de baja resolución como a los de alta y de esta forma reducir el sesgo obtenido durante el proceso de reconstrucción. Esta transformación no es usada, a saber, en los algoritmos presentados en el Capítulo 2 ni con los algoritmos con los que se comparan resultados. Lo común es que se centre alrededor de cero cada parche usando su media correspondiente y en el proceso de reconstrucción se recupere el parche de alta resolución usando como estimación la media del parche de baja resolución, lo cual introduce un sesgo en los resultados. En general, el cambio propuesto en la transformación de los parches generó mejores resultados que la transformación tradicional.

En cuanto al modelo basado en pares de diccionarios, consideramos que las aportaciones y conclusiones son bastante significativas. Una de las aportaciones importantes está en el uso de la norma l_F en relación con el factor de regularización al estimar los diccionarios de alta resolución ya que, a saber, esto es poco común en los trabajos del estado del arte. La mayoría de los trabajos que hacen uso de pares de diccionarios, estiman estos diccionarios usando técnicas de optimización sobre modelos que hacen uso de la norma l_1 buscando soluciones sparse. Otro de los resultados significativos es que a pesar de que se hace uso de la norma l_1 para estimar los diccionarios de baja resolución, los estudios realizados mostraron que factor de regularización λ genera mejores resultados cuando tiene un valor pequeño lo que sugiere que el hecho de buscar soluciones sparse no sería una estrategia bien fundamentada (al menos en nuestra propuesta). Otro resultado importante es que los diccionarios grandes no representaron mejores resultados en nuestra propuesta. Por el contrario, fueron los diccionarios con pocos átomos los que generaron mejores resultados, esto sugiere que el uso de diccionarios sobre-completos no es necesario (al menos en nuestro caso). Estos resultados experimentales son importantes ya que abren camino a nuevas ideas y caminos

que explorar en el tema de Single Image Super Resolution.

Una ventaja del esquema propuesto es que no es un proceso iterativo, es decir, no hay que procesar varias veces la imagen de baja resolución para generar una estimación de alta resolución. Varios algoritmos del estado del arte funcionan como un proceso iterativo y esto repercuten en el tiempo computacional y ocasiona que esos algoritmos sean menos eficientes con respecto a las propuestas presentadas en este trabajo de tesis.

En general, los resultados obtenidos por nuestras propuestas, si bien, no podemos decir que son mejores que los resultados de los algoritmos del estado del arte, sí podemos decir que son competitivos y que en ocasiones generan resultados que llegan a superar a varios de los algoritmos del estado del arte y cuando no, los resultados quedan bastante cerca de los resultados de los otros algoritmos. En cuanto a las aportaciones y resultados obtenidos de los estudios realizados en este trabajo, se abren nuevos caminos y posibilidades para continuar las investigaciones sobre el tema de Super Resolution dejando bastante trabajo a futuro por realizar.

5.2. Trabajo Futuro

En el tema de Single Image Super Resolution aún queda mucho trabajo por hacer. Esta sección es para explicar el trabajo a futuro que, en la opinión personal de los autores de esta tesis, falta por realizar para hacer más robusta la investigación presentada en este documento. Con el objetivo de servir de guía a quien desee abordar el tema de restauración de imágenes y específicamente Single Image Super Resolution, en la siguiente lista se detallan algunos de los puntos sobre el trabajo a futuro que se propone.

- **Mejorar la estimación inicial.**- Con estimación inicial hacemos referencia al preproceso de llevar una imagen de baja resolución I_l a una resolución espacial de mayor tamaño haciendo uso de la interpolación bicubic. Este tipo de interpolación introduce artefactos no deseados que gran mayoría de los algoritmos son magnificados afectando la calidad de la imagen estimada de alta resolución. Posiblemente enfocar algo de trabajo a crear un método similar a la interpolación bicubic pero tratando de minimizar ese tipo de artefactos.
- **Método para estimación de σ .**- El valor de σ en la mayoría de los algoritmos propuestos se desconoce y al menos ninguno de los trabajos citados en esta tesis explica una forma de abordar el problema y poder estimarlo. Trabajar en algún método que

dada una imagen de baja resolución determine o arroje una buena aproximación sobre el valor σ del modelo de degradación que la generó representaría una ventaja ya que sería posible eliminar un parámetro de varios algoritmos propuestos.

- **Algoritmo con agrupamiento y redes convolucionales.**- Aunque las redes convolucionales han demostrado su efectividad para resolver el problema de Single Image Super Resolution en el trabajo de Dong et al. (2014), sería interesante probar si el agrupamiento puede ofrecer mejores resultados.
- **Proponer nuevas restricciones.**- En este trabajo se hablo de dos restricciones, una local y otra global. Estas restricciones ayudan a que la solución obtenida este cerca de la solución real, sin embargo, es bien conocido que el problema de Single Image Super Resolution es altamente mal condicionado por lo que proponer nuevas restricciones o formas de regularizar la solución representarían una aportación importante.
- **Realizar pruebas para datos en 3D como imágenes de resonancia magnética.**- El área de imágenes médicas es una área de gran importancia y dentro de esta área se encuentran las imágenes de resonancia magnética o MRI por sus siglas en inglés. Este tipo de información es representado por un volumen 3D que los modelos presentados en este trabajo de tesis pueden abordar sin problemas, sólo se debe modificar el proceso de extracción de parches donde en lugar de extraer parches de forma cuadrada se extraerán parches en forma de cubo. Este trabajo requiere que se implementen métodos de optimización de forma eficiente que funcionen en paralelo ya que al ser los parches representados por un cubo la información a procesar crece considerablemente.
- **Realizar pruebas para videos.**- De manera similar a la propuesta para imágenes MRI, un vídeo de una misma escena (sin saltos de escenarios) podría procesarse con información en 3D donde los parches ahora serán representados por cubos tomando información de frames vecinos. Sería interesante ver los resultados que generarían los modelos propuestos sobre este tipo de información.
- **Realizar pruebas con bases de datos más grandes.**- En la Sección de experimentos y resultados se mostró el comportamiento que tienen los algoritmos propuestos cuando se incrementa el tamaño de la base de datos. Sin embargo sólo se realizaron

pruebas con a lo más 2 millones de pares de parches. Sería interesante probar con bases de datos con más de 5 millones de parches y ver si la tendencia de mejorar el PSNR se mantiene con mayor cantidad de datos o si existe algún punto donde se estabiliza y no hay un efecto significativo si se agregan más parches a la base de datos.

- **Verificar si hay regiones donde un determinado tamaño de parche funcione mejor.**- Estas pruebas consistirán en verificar si para cierto tipo de regiones o texturas en una imagen natural, existe un tamaño de parche adecuado. Si esto fuera verdad entonces se podrían entrenar redes neuronales artificiales o aprender pares de diccionarios con distintos tamaños de parches y adaptar los modelos para que seleccionen cual tamaño de parche es mejor para cierta zona de una nueva imagen.
- **Probar con otras técnicas de agrupamiento.**- Parte del trabajo futuro por realizar es probar con nuevas técnicas de agrupamiento y ver si es posible que se logren mejorar los resultados obtenidos con K-Means.
- **Uso de segmentación y clasificación de texturas.**- Un posible trabajo a futuro es entrenar distintos conjuntos de parámetros usando distintas clases de texturas, de manera que sea posible identificar cual conjunto de parámetros es el más adecuado acorde a la textura de un cierto parche de baja resolución. El resultado de los experimentos realizados sobre la base de datos de texto, sugieren que aprender parámetros con parches provenientes de una cierta clase ayudan a mejorar los resultados, es por ello que se propone segmentar y clasificar texturas y aprovechar esta información para mejorar los resultados.

Bibliografía

Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *ICCV, 2009*. URL <http://tinyurl.com/cqkpa5v>.

Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006. ISBN 013168728X.

Michal Irani and Shmuel Peleg. Improving resolution by image registration. *CVGIP: Graph. Models Image Process.*, 53(3):231–239, April 1991. ISSN 1049-9652. doi: 10.1016/1049-9652(91)90045-L. URL [http://dx.doi.org/10.1016/1049-9652\(91\)90045-L](http://dx.doi.org/10.1016/1049-9652(91)90045-L).

Gilad Freedman and Raanan Fattal. Image and video upscaling from local self-examples. *ACM Trans. Graph.*, 28(3):1–10, 2010. ISSN 0730-0301. doi: <http://doi.acm.org/10.1145/1531326.1531328>.

Michal Irani and Shmuel Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324 – 335, 1993. URL <http://tinyurl.com/hqvvu3b>.

Sina Farsiu, M. Dirk Robinson, Michael Elad, and Peyman Milanfar. Fast and robust multiframe super resolution. *IEEE Transactions on Image Processing*, 13(10):1327–1344, 2004. URL <http://tinyurl.com/hrdn55>.

Jianchao Yang, John Wright, Thomas S. Huang, and Yi Ma. Image super-resolution via sparse representation. *Trans. Img. Proc.*, 19(11):2861–2873, November 2010. ISSN 1057-7149. doi: 10.1109/TIP.2010.2050625. URL <http://dx.doi.org/10.1109/TIP.2010.2050625>.

Weisheng Dong, Lei Zhang B, Guangming Shi A, Senior Member, Xiaolin Wu C, and Senior Member. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization, 2011.

S. Wang, L. Zhang, Y. Liang, and Q. Pan. Semi-coupled dictionary learning with applications in image super-resolution and photo-sketch synthesis. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012.

L. An and B. Bhanu. Improved image super-resolution by support vector regression. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 696–700, 2011. doi: 10.1109/IJCNN.2011.6033289.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *CoRR*, abs/1501.00092, 2015. URL <http://arxiv.org/abs/1501.00092>.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, 2001. ISSN 0036-1445. URL <http://tinyurl.com/hz7vttz>.

Florent Couzinie-Devy, Julien Mairal, Francis R. Bach, and Jean Ponce. Dictionary learning for deblurring and digital zoom. *CoRR*, abs/1110.0957, 2011. URL <http://arxiv.org/abs/1110.0957>.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, December 1989. ISSN 0899-7667. doi: 10.1162/neco.1989.1.4.541. URL <http://dx.doi.org/10.1162/neco.1989.1.4.541>.

Viren Jain and Sebastian Seung. Natural image denoising with convolutional networks. In *Advances in Neural Information Processing Systems 21*, pages 769–776. Curran Associates, Inc., 2009. URL <http://papers.nips.cc/paper/3506-natural-image-denoising-with-convolutional-networks.pdf>.

David Eigen, Dilip Krishnan, and Rob Fergus. Restoring an image taken through a window covered with dirt or rain. *Computer Vision, IEEE International Conference on*, 0:633–640, 2013. ISSN 1550-5499.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV*, chapter Learning a Deep Convolutional Network for Image Super-Resolution, pages 184–199. Springer International Publishing, Cham, 2014. URL <http://tinyurl.com/zreanqv>.

- Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 60–65, Washington, DC, USA, 2005. IEEE Computer Society. URL <http://dx.doi.org/10.1109/CVPR.2005.38>.
- A. Rajavelu, M.T. Musavi, and M.V. Shirvaikar. A neural network approach to character recognition. *Neural Networks*, 2(5):387 – 393, 1989. ISSN 0893-6080. doi: [http://dx.doi.org/10.1016/0893-6080\(89\)90023-3](http://dx.doi.org/10.1016/0893-6080(89)90023-3). URL <http://www.sciencedirect.com/science/article/pii/0893608089900233>.
- Kenneth O. Stanley, Bobby D. Bryant, and Risto Miikkulainen. Evolving neural network agents in the nero video game. In *Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games (CIG'05)*, Piscataway, NJ, 2005. IEEE. URL <http://nn.cs.utexas.edu/?stanley:cig05>.
- Vladik Kreinovich Carlos Ramirez and Miguel Argaez. Why 11 is a good approximation to 10: A geometric explanation. *Journal of Uncertain Systems*, 7, 2013.
- Saurabh Karsoliya. Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture. *International Journal of Engineering Trends and Technology*, 3: 714–717, 2012.
- Oscar Dalmau Cedeño, Teresa E. Alarcón, and Graciela González. Kernel multilogit algorithm for multiclass classification. *Computational Statistics & Data Analysis*, 82:199–206, 2015. doi: 10.1016/j.csda.2014.09.007. URL <http://dx.doi.org/10.1016/j.csda.2014.09.007>.
- Kwang In Kim and Younghée Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(6): 1127–1133, June 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.25. URL <http://dx.doi.org/10.1109/TPAMI.2010.25>.
- Chih-Yuan Yang and Ming-Hsuan Yang. Fast direct super-resolution by simple functions. In *Proceedings of IEEE International Conference on Computer Vision*, 2013.
- Radu Timofte, Vincent De Smet, and Luc J. Van Gool. Adjusted anchored neighborhood regression for fast super-resolution. In *Computer Vision - ACCV 2014 - 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part IV*, pages 111–126, 2014. URL <http://tinyurl.com/zg79cnx>.
- Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In *Proceedings of European Conference on Computer Vision*, 2014.