

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



TRABAJO FIN DE MÁSTER

INTELIGENCIA ARTIFICIAL APLICADA A
SÚPER-RESOLUCIÓN EN TERMOGRAFÍA

**Titulación: Máster Universitario en Ingeniería de
Telecomunicación**

Autor: D. Antonio David Galván Hernández

Tutores: D. Víctor Alexis Araña Pulido

D. Jaime Roberto Ticay Rivas

D. Víctor Alonso Eugenio

Fecha: enero de 2022

ESCUELA DE INGENIERÍA DE
TELECOMUNICACIÓN Y ELECTRÓNICA



TRABAJO FIN DE MÁSTER

INTELIGENCIA ARTIFICIAL APLICADA A
SÚPER-RESOLUCIÓN EN TERMOGRAFÍA

HOJA DE FIRMAS

Firma de los tutores

Fdo.: _____

Fdo.: _____

Fdo.: _____

Firma del alumno

Fdo.: _____

Fecha: enero de 2022

ESCUELA DE INGENIERÍA DE
TELECOMUNICACIÓN Y ELECTRÓNICA



TRABAJO FIN DE MÁSTER

INTELIGENCIA ARTIFICIAL APLICADA A
SÚPER-RESOLUCIÓN EN TERMOGRAFÍA

HOJA DE EVALUACIÓN

Calificación: _____

Presidente

Secretario

Vocal

Fdo.: _____

Fdo.: _____

Fdo.: _____

Fecha: enero de 2022

Agradecimientos

A mi familia, que siempre ha sido paciente conmigo.

A mis amigos, por estar ahí y sacarme de la monotonía.

*A mis tutores, Víctor Araña, Jaime Ticay y Víctor Alonso, ya que sin ellos no
estaría escribiendo este texto el día de hoy.*

- Antonio David Galván Hernández -

Índice general

Acrónimos	XVIII
I Memoria	1
1. Introducción	2
1.1. Antecedentes	3
1.1.1. Conceptos sobre el espectro infrarrojo y la termografía	4
1.1.2. El concepto de súper-resolución	10
1.2. Objetivos	15
1.3. Motivación	16
1.4. Estructura de la memoria	16
2. Marco teórico	18
2.1. Termografía	18
2.2. Cámaras Termográficas	24
2.2.1. Consideraciones sobre el sensor	24
2.2.2. Consideraciones sobre la óptica	26
2.3. Súper-resolución	27
2.3.1. Modelo de observación	29
2.3.2. Métodos de súper-resolución	30
2.3.3. Súper-resolución en el dominio espacial	32
2.4. Métodos de reconstrucción	33
2.4.1. Método basado en algoritmos de interpolación	33
2.4.1.1. Métodos de interpolación clásicos	36
2.4.2. Métodos de regularización mediante aproximaciones Bayesianas	37

2.4.2.1.	Problemas inversos	37
2.4.2.2.	Aproximaciones bayesianas	39
2.4.3.	Método de retroproyección iterativa	40
2.4.3.1.	IBP en la súper-resolución	40
2.4.4.	Método de proyección sobre conjuntos convexos	41
2.4.4.1.	POCS en la súper-resolución	42
2.4.5.	Método de Papoulis-Gerchberg	45
2.4.5.1.	Bases del método	45
2.4.5.2.	El método de Papoulis-Gerchberg en la súper-resolución	48
2.4.6.	Método de súper-resolución robusta	49
2.4.6.1.	Uso de la mediana de píxeles	50
2.4.6.2.	El proceso para la detección de sesgos	51
2.5.	Contextualización de las redes neuronales	52
2.5.1.	Redes Neuronales	53
2.5.1.1.	Aprendizaje de una neurona	54
2.5.1.2.	Funciones de activación	56
2.5.1.3.	Cálculo del error en la predicción	57
2.5.1.4.	Optimización del error	59
2.5.1.5.	Propagación del error calculado	61
2.5.2.	Redes neuronales convolucionales	62
2.5.2.1.	La operación de convolución	62
2.6.	Contexto de las redes neuronales en la súper-resolución	65
2.6.1.	Método SRCNN	65
2.6.1.1.	Formulación del método	65
2.6.2.	Tipos de redes neuronales	68
2.6.2.1.	Redes neuronales lineales	68
2.6.2.2.	Redes neuronales recursivas	68
2.6.2.3.	Redes neuronales residuales	69
2.6.2.4.	Redes generativas adversarias	70
2.6.3.	Metodologías de muestreo aplicadas a la súper-resolución	71
2.6.3.1.	Muestreo de la imagen	71
2.6.4.	Funciones de pérdida	75

2.6.4.1.	Pérdida de píxeles	75
2.6.4.2.	Pérdida de contenido	75
2.6.4.3.	Pérdida adversaria	76
2.6.5.	Métricas de calidad	76
2.6.5.1.	PSNR	76
2.6.5.2.	SSIM	77
3.	Estado de la técnica	78
3.1.	Métodos basados en redes neuronales	78
3.1.1.	VDSR	78
3.1.1.1.	Proceso de entrenamiento y función de pérdida	79
3.1.2.	SRGAN	80
3.1.2.1.	Proceso de entrenamiento y función de pérdida	81
3.1.3.	EDSR	82
3.1.3.1.	Proceso de entrenamiento y función de pérdida	83
3.1.4.	MDSR	84
3.1.4.1.	Proceso de entrenamiento y función de pérdida	85
3.1.5.	WDSR	85
3.1.5.1.	Proceso de entrenamiento y función de pérdida	86
3.1.6.	ESRGAN	87
3.1.6.1.	Cambios en el generador y discriminador	88
3.1.6.2.	Pérdida VGG modificada	88
3.1.6.3.	Proceso de entrenamiento y función de pérdida total	89
3.1.7.	Real-ESRGAN	89
3.1.7.1.	Proceso de entrenamiento y función de pérdida	91
4.	Bases de datos	92
4.1.	Bases de datos para la súper-resolución espacial	93
4.1.1.	Bases de datos para el entrenamiento de los métodos	93
4.1.2.	Base de datos para la evaluación de los métodos	93
4.2.	Base de datos para la súper-resolución radiométrica	95
4.2.1.	Base de datos para el entrenamiento y evaluación	95
4.2.1.1.	Calibración de la superficie radiante	96

4.2.1.2.	Obtención de muestras para el entrenamiento y la evaluación	99
4.3.	Resumen de las bases de datos propuestas	102
5.	Experimentación	103
5.1.	Propuesta para la súper-resolución espacial	103
5.1.1.	Métodos basados en reconstrucción	104
5.1.2.	Métodos basados en redes neuronales	104
5.1.2.1.	Implementaciones elegidas	105
5.1.3.	Consideraciones adicionales	108
5.2.	Propuesta para la súper-resolución radiométrica	109
5.2.1.	El problema de las lecturas de una cámara	110
5.2.2.	Solución propuesta	111
5.2.3.	Entrenamiento de la red neuronal	113
5.2.3.1.	Estrategia para el pre-entrenamiento	116
5.2.3.2.	Entrenamiento total de la red	117
5.3.	Experimentación con la súper-resolución espacial	119
5.3.1.	Pre-procesado para los métodos de MATLAB	120
5.3.2.	Pre-procesado para los métodos de Python	121
5.4.	Experimentación con la súper-resolución radiométrica	122
5.5.	Experimentación con ambas súper-resoluciones	123
6.	Resultados	125
6.1.	Resultados del experimento 1	125
6.2.	Resultados del experimento 2	130
6.3.	Resultados del experimento 3	137
6.4.	Resultados del experimento 4	148
6.5.	Resultados del experimento 5	153
7.	Conclusiones y líneas futuras	160
7.1.	Conclusiones	160
7.2.	Líneas futuras	165

II Bibliografía	167
Bibliografía	178
III Presupuesto	179
Presupuesto	180
P.1. Introducción	180
P.2. Recursos Materiales	180
P.2.1. Equipos Hardware	181
P.2.2. Equipos Software	182
P.3. Recursos Humanos	182
P.4. Redacción del documento	183
P.5. Derechos de visado del COIT	184
P.6. Gastos de tramitación y envío	185
P.7. Aplicación de impuestos y coste final	185
IV Anexos	187
A. Resultado visual de los métodos	188
A.1. Cámara A615	188
A.2. Cámara Lepton	196
B. Creación del demostrador	201

Índice de figuras

1.1. Espectro electromagnético según temperatura, longitud de onda y frecuencia. Extraída de: [3]	3
1.2. Influencia del conjunto óptica-sensor en una cámara	6
1.3. Representación básica del funcionamiento de una cámara termográfica .	7
1.4. Esquema simplificado del estado de la técnica en cuestión a súper-resolución	11
1.5. a) Superficie de estudio, una bobina inductiva sobre una pieza de latón. b) Imagen termográfica original e imagen de súper-resolución. Extraída de: [28]	13
1.6. Simplificación de un método de súper-resolución	15
2.1. Efectos de la atmósfera en las medidas de una cámara termográfica . .	23
2.2. Funcionamiento básico de un detector de microbolómetro. Adaptada de [39]	25
2.3. Representación de la correlación entre FOV y distancia focal	26
2.4. Representación de distintos tipos de FOV con la herramienta Fieldometer	27
2.5. Modelo de observación genérico de una imagen	29
2.6. Relación de solapamiento entre la imagen LR y HR. Adaptada de [44] .	31
2.7. Formación de una imagen SR a partir de un conjunto de imágenes LR. Adaptada de [42]	34
2.8. Estimaciones de una interpolación lineal según el número de muestras .	36
2.9. Representación gráfica del algoritmo POCS. Adaptada de [53]	42
2.10. Proyecciones sobre dos conjuntos convexos cerrados. Adaptada de [56] .	43
2.11. Ejemplo de convergencia de un algoritmo POCS. Adaptada de [56] . . .	44
2.12. Método de Papoulis-Gerchberg (I). Extraída de [60]	45

2.13. Método de Papoulis-Gerchberg (II). Extraída de [60]	46
2.14. Método de Papoulis-Gerchberg (III). Extraída de [60]	46
2.15. Método de Papoulis-Gerchberg (IV). Extraída de [60]	47
2.16. Funcionamiento simplificado de una neurona (I). Adaptada de [65] . . .	54
2.17. Funcionamiento simplificado de una neurona (II). Adaptada de [66] . .	55
2.18. Generalización de una red neuronal profunda. Adaptada de [67]	56
2.19. Representación de la función sigmoide y ReLU	57
2.20. Ejemplo de una convolución de matrices	63
2.21. Proceso de convolución con distintos rellenos y parámetros de paso . .	64
2.22. Arquitectura del método SRCNN. Extraída de [73]	66
2.23. Conexión de salto en una ResNet. Adaptada de [77]	69
2.24. Diseño de una red GAN. Adaptada de [79]	70
2.25. Muestreo en el proceso de súper-resolución. Adaptada de [80]	73
2.26. Ejemplos de muestreo basado en aprendizaje. Adaptada de [80]	74
3.1. Arquitectura de VDSR. Extraída de [82]	79
3.2. Arquitectura de SRGAN. Extraída de [83]	81
3.3. Arquitectura de EDSR. Extraída de [85]	83
3.4. Modificación del bloque SRResNet propuesta en [85]	84
3.5. Arquitectura de MDSR. Extraída de [85]	85
3.6. Arquitectura de WDSR. Extraída de [87]	86
3.7. Arquitectura de ESRGAN. Extraída de [88]	87
3.8. Discriminador que implementa ESRGAN. Extraída de [88]	88
3.9. Degradación de la imagen llevada a cabo en el método Real-ESRGAN [89]	90
3.10. Arquitectura de Real-ESRGAN. Extraída de [89]	90
3.11. Diseño genérico de una U-NET. Adaptada de [90]	91
4.1. Cámaras usadas para la creación de la primera base de datos	94
4.2. Muestras extraídas de la base de datos DASSR	95
4.3. Sistema de cámaras y superficie radiante propuesta	96
4.4. Validación de temperaturas con la sonda Fluke 51 II	97
4.5. Experimento realizado para la calibración de la vitrocerámica	98
4.6. Rangos de temperatura de la vitrocerámica PCE-5000	99

4.7. Procedimiento para la creación de la base de datos	100
5.1. Comparativa entre imágenes termográficas de 16 bits y 8 bits	109
5.2. Red neuronal planteada para la súper-resolución radiométrica	112
5.3. Arquitectura de la red neuronal diseñada con Keras	114
5.4. Muestras elegidas de la base de datos TID	115
5.5. Mecanismo de Kfold implementado	117
5.6. Gráfica resultante del proceso de entrenamiento	118
5.7. Flujo de trabajo contemplado en MATLAB para la súper-resolución es- pacial	121
5.8. Flujo de trabajo contemplado en Python para la súper-resolución espacial	122
5.9. Flujo de trabajo contemplado para la súper-resolución radiométrica . .	123
5.10. Flujo de trabajo contemplado para la conjunción de súper-resoluciones	124
6.1. Comparación de escenas observadas entre cámaras con distinta óptica .	127
6.2. Representación gráfica de las Tablas 6.1 y 6.2 - Escala x4	129
6.3. Representación gráfica de las Tablas 6.3 y 6.5 - Escala x2	134
6.4. Representación gráfica de las Tablas 6.4 y 6.6 - Escala x4	134
6.5. Resultados del método de interpolación en la cámara A615	135
6.6. Resultados del método EDSR en la cámara A615	136
6.7. Resultados del método Real-ESRGAN en la cámara A615	136
6.8. Representación gráfica de las Tablas 6.7 y 6.9 - Escala x2 (I)	142
6.9. Representación gráfica de las Tablas 6.8 y 6.10 - Escala x4 (I)	142
6.10. Representación gráfica de las Tablas 6.7 y 6.9 - Escala x2 (II)	143
6.11. Representación gráfica de las Tablas 6.8 y 6.10 - Escala x4 (II)	143
6.12. Resultados del método de interpolación en la cámara Lepton	144
6.13. Resultados del método EDSR en la cámara Lepton	144
6.14. Resultados del método Real-ESRGAN en la cámara Lepton	144
6.15. Representación gráfica de las Tablas 6.11 y 6.12 - Escala x2 (I)	147
6.16. Representación gráfica de las Tablas 6.11 y 6.12 - Escala x2 (II)	147
6.17. Representación gráfica de la Tabla 6.13	149
6.18. Muestras evaluadas del experimento 4 - Distancias: 2 a 18 metros . . .	149
6.19. Representación gráfica de la Tabla 6.14	150

6.20. Muestras evaluadas del experimento 4 - Distancias: 3 a 19 metros . . .	151
6.21. Representación gráfica de la Tabla 6.15	152
6.22. Muestras evaluadas del experimento 4 - Distancias: 21 a 31 metros . . .	152
6.23. Representación gráfica de las Tablas 6.16 y 6.17	155
6.24. Muestras evaluadas del experimento 5 - Distancias: 2 a 18 metros . . .	155
6.25. Representación gráfica de las Tablas 6.18 y 6.19	157
6.26. Muestras evaluadas del experimento 5 - Distancias: 3 a 19 metros . . .	157
6.27. Representación gráfica de las Tablas 6.20 y 6.21	159
6.28. Muestras evaluadas del experimento 5 - Distancias: 21 a 31 metros . . .	159
A.1. Resultados del método de interpolación en la cámara A615	188
A.2. Resultados del método IBP en la cámara A615	189
A.3. Resultados del método POCS en la cámara A615	189
A.4. Resultados del método de Papoulis-Gerchberg en la cámara A615 . . .	190
A.5. Resultados del método de súper-resolución robusta en la cámara A615 .	190
A.6. Resultados del método VDSR en la cámara A615	191
A.7. Resultados del método SRGAN en la cámara A615	191
A.8. Resultados del método SRGAN_TH en la cámara A615	192
A.9. Resultados del método EDSR en la cámara A615	192
A.10. Resultados del método EDSR_TH en la cámara A615	193
A.11. Resultados del método MDSR en la cámara A615	193
A.12. Resultados del método WDSR en la cámara A615	194
A.13. Resultados del método WDSR_TH en la cámara A615	194
A.14. Resultados del método ESRGAN en la cámara A615	195
A.15. Resultados del método Real-ESRGAN en la cámara A615	195
A.16. Resultados del método de interpolación en la cámara Lepton	196
A.17. Resultados del método IBP en la cámara Lepton	196
A.18. Resultados del método POCS en la cámara Lepton	196
A.19. Resultados del método de Papoulis-Gerchberg en la cámara Lepton . .	197
A.20. Resultados del método de súper-resolución robusta en la cámara Lepton	197
A.21. Resultados del método VDSR en la cámara Lepton	197
A.22. Resultados del método SRGAN en la cámara Lepton	198
A.23. Resultados del método SRGAN_TH en la cámara Lepton	198

A.24.Resultados del método EDSR en la cámara Lepton	198
A.25.Resultados del método EDSR_TH en la cámara Lepton	198
A.26.Resultados del método MDSR en la cámara Lepton	199
A.27.Resultados del método WDSR en la cámara Lepton	199
A.28.Resultados del método WDSR_TH en la cámara Lepton	199
A.29.Resultados del método ESRRGAN en la cámara Lepton	200
A.30.Resultados del método Real-ESRRGAN en la cámara Lepton	200
B.1. Visualización de la herramienta <i>Qt Designer</i>	201
B.2. Visualización general del demostrador	202
B.3. Proceso de visualización de la imagen LR del demostrador	203
B.4. Proceso de para aplicar la súper-resolución espacial del demostrador . .	203
B.5. Paletas de colores implementadas en el demostrador	204
B.6. Valores de los píxeles deseados y su transformación en temperatura (I)	205
B.7. Valores de los píxeles deseados y su transformación en temperatura (II)	205
B.8. Rutina para almacenar imágenes SR en el demostrador	206
B.9. Proceso de almacenamiento por parte de matplotlib y borrado del lienzo	207

Índice de tablas

4.1. Características de las cámaras mostradas en la Figura 4.1	94
4.2. Características de las cámaras mostradas en la Figura 4.3	97
4.3. Bases de datos usadas en el TFM	102
5.1. Entrenamiento y evaluación del método SRGAN	106
5.2. Entrenamiento y evaluación de los métodos EDSR y WDSR-B	107
5.3. Hiperparámetros de la red neuronal propuesta	114
5.4. Resultados del mecanismo Kfold implementado	117
5.5. Resultados del entrenamiento planteado	118
6.1. Resultados del experimento 1 - Métodos de MATLAB en escala x4 . . .	126
6.2. Resultados del experimento 1 - Métodos de Python en escala x4 . . .	128
6.3. Resultados del experimento 2 - Métodos de MATLAB en escala x2 . . .	130
6.4. Resultados del experimento 2 - Métodos de MATLAB en escala x4 . . .	131
6.5. Resultados del experimento 2 - Métodos de Python en escala x2	132
6.6. Resultados del experimento 2 - Métodos de Python en escala x4	133
6.7. Resultados del experimento 3 (I) - Métodos de MATLAB en escala x2 .	138
6.8. Resultados del experimento 3 (I) - Métodos de MATLAB en escala x4 .	139
6.9. Resultados del experimento 3 (I) - Métodos de Python en escala x2 . .	140
6.10. Resultados del experimento 3 (I) - Métodos de Python en escala x4 . .	141
6.11. Resultados del experimento 3 (II) - Métodos de MATLAB en escala x2	145
6.12. Resultados del experimento 3 (II) - Métodos de Python en escala x2 . .	146
6.13. Resultados del experimento 4 - Distancias: 2 a 18 metros	148
6.14. Resultados del experimento 4 - Distancias: 3 a 19 metros	150
6.15. Resultados del experimento 4 - Distancias: 21 a 23 metros	151
6.16. Resultados del experimento 5 - Distancias: 2 a 18 metros (Valor original)	153

6.17. Resultados del experimento 5 - Distancias: 2 a 18 metros (Valor predicho)	154
6.18. Resultados del experimento 5 - Distancias: 3 a 19 metros (Valor original)	156
6.19. Resultados del experimento 5 - Distancias: 3 a 19 metros (Valor predicho)	156
6.20. Resultados del experimento 5 - Distancias: 21 a 31 metros (Valor original)	158
6.21. Resultados del experimento 5 - Distancias: 21 a 31 metros (Valor predicho)	158
7.1. Precio de mercado de las cámaras de FLIR utilizadas	164
P.1. Coste de los recursos hardware	181
P.2. Coste de los recursos software	182
P.3. Coste de los recursos humanos	183
P.4. Presupuesto parcial	184
P.5. Presupuesto con el coste de redacción	185
P.6. Presupuesto total	186

Acrónimos

Siglas	Descripción
Adam	Adaptive Moment Estimation
ADC	Analog-to-Digital Conversion
AGC	Automatic Gain Control
CNN	Convolutional Neural Networks
COIT	Colegio Oficial Ingenieros de Telecomunicaciones
DASSR	Database for the Assessment of Spatial Super-Resolution
EDSR	Enhanced Deep Super Resolution
EITE	Escuela de Ingeniería de Telecomunicación y Electrónica
ESRGAN	Enhanced Super Resolution Generative Adversarial Network
FOV	Field Of View
FPA	Focal Plane array
GAN	Generative Adversarial Networks
HG	High Gain
HR	High Resolution
IBP	Iterative Back Projection
IDeTIC	Instituto para el Desarrollo Tecnológico y la Innovación en Comunicación
IFOV	Instantaneous Field Of View
ILR	Interpolated Low Resolution
IRFPA	Infrared Focal Plane array
irRCNN	infrared Residual Convolutional Neural Network
JPEG	Joint Photographic Experts Group
LeakyRelu	Leaky Rectified Linear Unit

LG	Low Gain
LR	Low Resolution
LWIR	Longwave Infrared
MAE	Mean Absolute Error
MAP	Maximum A Posteriori
MAPE	Mean Absolute Percentage Error
MDSR	Multi-scale Deep Super Resolution
MSE	Mean Square Error
ML	Maximum Likelihood
MWIR	Midwave Infrared
SGD	Stochastic Gradient Descent
SNR	Signal to Noise Ratio
SRCNN	Super-Resolution Convolutional Neural Network
SRGAN	Super Resolution Generative Adversarial Network
TID	Thermal IDeTIC Database
TIFF	Tagged Image File Format
POCS	Projection Onto Convex Set
PSNR	Peak-Signal-to-Noise Ratio
PReLU	Parametric ReLU
RaGAN	Relativistic GAN
ReLU	Rectified Linear Unit
RGB	Red Green Blue
RMSE	Root Mean Squared Error
ROI	Region Of Interest
ROIC	Readout Integrated Circuit
RRDB	Residual-in-Residual Dense Block
SR	Superresolved
SSIM	Structural Similarity Index Measure
SWIR	Shortwave Infrared
TCR	Temperature Coefficient of Resistance
TFM	Trabajo Fin de Máster
ULPGC	Universidad de Las Palmas de Gran Canaria

VDSR	Very Deep Super Resolution
VGA	Video Graphics Array
VPA	Vacuum Package Assembly
WDSR	Wide Activation for Image Super Resolution

Resumen

Las cámaras termográficas son los dispositivos más utilizados para determinar, sin la necesidad de contacto, la temperatura de un objeto incluso cuando se encuentra a una gran distancia. No obstante, una de las limitaciones clásicas de estas cámaras es su baja resolución espacial en comparación con las cámaras de luz visible. En este aspecto, el paradigma cambia cuando entran en escena los métodos de súper-resolución, especialmente los basados en inteligencia artificial. La idea de estos métodos es mejorar la resolución espacial de las imágenes mediante técnicas de aprendizaje profundo, consiguiendo resultados rápidamente y a un precio menor que cambiando el hardware.

Este proyecto propone el estudio e implementación de distintos métodos de súper-resolución aplicados a imágenes termográficas de baja resolución. A partir de la aplicación de estos métodos de súper-resolución se realizará un estudio para poder estimar las pérdidas empíricas de estos al procesar una imagen termográfica, así como su resultado visual en comparación con las imágenes originales de baja resolución. De la misma manera, se propone un método basado en redes neuronales capaz de corregir los datos radiométricos obtenidos de una cámara de baja resolución a partir de la distancia que existe entre esta y el blanco de estudio. Este planteamiento aportará valor al estado de la técnica, ya que no sólo se mejorará la calidad visual de las imágenes, sino el valor radiométrico que representan los píxeles.

Los resultados finales son derivados de una gran campaña de medidas para la recolección de datos y la creación de las muestras de entrenamiento y evaluación. De esta forma, dos bases de datos de imágenes termográficas son usadas para la evaluación de los métodos del estado de la técnica y el propuesto en este trabajo, discutiendo a partir de estos resultados que métodos mejoran la calidad visual de una cámara termográfica de bajo coste, así como la mejora en la lectura de la temperatura.

Abstract

Thermal imaging cameras are the most widely used devices to determine, without the need for contact, the temperature of an object even when it is at a great distance. However, one of the classic limitations of these cameras is their low spatial resolution compared to visible light cameras. In this respect, the paradigm shifts when super-resolution methods enter the scene, especially those based on artificial intelligence. These methods aim to improve the spatial resolution of images using deep learning techniques, achieving results quickly and at a lower cost than changing the hardware.

This project proposes the study and implementation of different super-resolution methods applied to low-resolution thermographic images. From the application of these super-resolution methods, a study will be carried out to estimate the empirical losses of these methods when processing a thermographic image, as well as their visual result in comparison with the original low-resolution images. In the same way, a method based on neural networks capable of correcting the radiometric data obtained from a low-resolution camera based on the distance between the camera and the study target is proposed. This approach will add value to the state of the art since not only the visual quality of the images will be improved, but also the radiometric value represented by the pixels.

The final results are derived from an extensive measurement campaign for data collection and the creation of training and evaluation samples. In this way, two thermographic image databases are used to evaluate the state-of-the-art methods and the one proposed in this work, discussing from these results which methods improve the visual quality of a low-cost thermographic camera, as well as the improvement in the temperature reading.

Parte I

Memoria

Capítulo 1

Introducción

La fotografía, que se considera la técnica fundamental para la obtención de imágenes y su extracción de características, ha evolucionado a lo largo del tiempo junto a la comprensión humana de la teoría de la luz. En la actualidad, se ha conseguido explicar cómo interacciona la luz con su entorno, ya sea mediante un modelo matemático o por su comprensión más intuitiva. De esta forma, también se ha podido avanzar en los procedimientos técnicos que permiten captar esta interacción y plasmarla a ojos de las personas, independientemente del paso del tiempo o la localización geográfica.

Desde los tiempos de Isaac Newton, donde se realizaron los primeros experimentos para descomponer la luz blanca, se ha estudiado este fenómeno y extraído sus propiedades básicas [1]. De hecho, en aquella época (década de 1660) se estimaba que la luz visible o luz blanca era la única que conformaba lo que hoy día se conoce como espectro electromagnético. A partir de ese momento comenzaron a surgir conceptos como pueden ser la longitud de onda, frecuencia e intensidad de radiación, entre otros. No obstante, el descubrimiento del espectro electromagnético no se le puede atribuir a los esfuerzos de un solo científico ni a un período temporal específico. Si bien es cierto que Newton experimentaba con la luz visible y su descomposición en colores básicos como el rojo, naranja, amarillo, verde, azul y violeta [1], no es hasta períodos posteriores donde se demostró de forma práctica que existían otras regiones del espectro donde sucedían otro tipo de interacciones [2].

Por este motivo, el espectro electromagnético no se considera un descubrimiento de un solo siglo, sino una acumulación de conocimientos a lo largo del tiempo. El espectro

electromagnético se puede definir como una distribución energética de la radiación electromagnética y sus respectivas longitudes de onda, como puede observarse en la Figura 1.1. Este trabajo centrará sus esfuerzos en la región infrarroja del espectro.

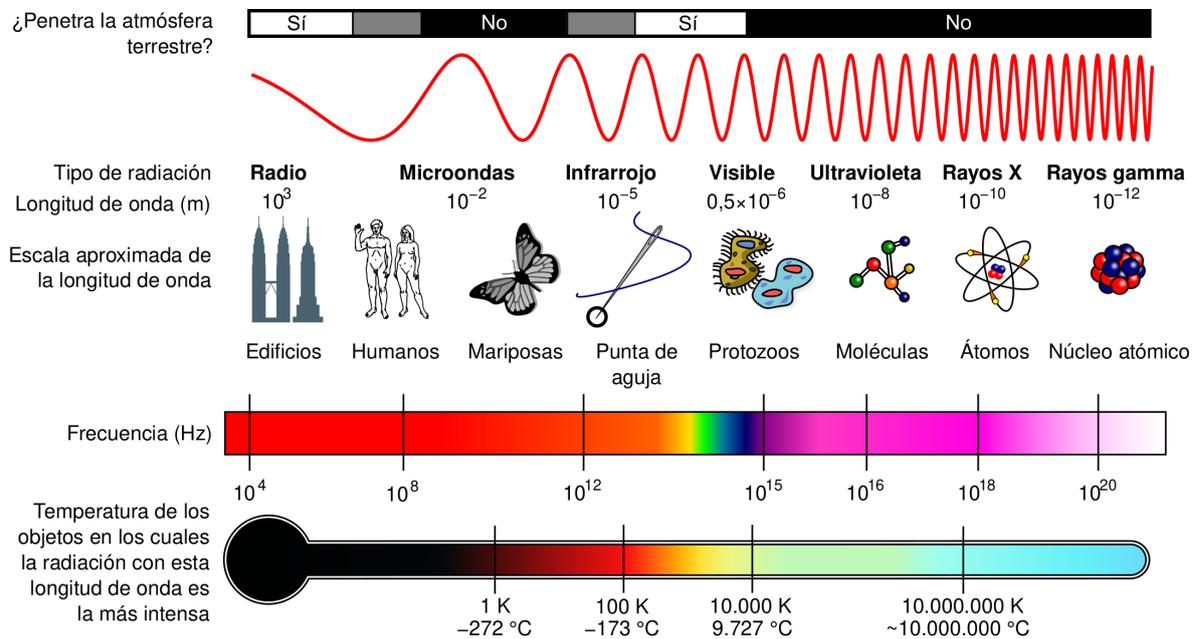


Figura 1.1: Espectro electromagnético según temperatura, longitud de onda y frecuencia. Extraída de: [3]

1.1. Antecedentes

El espectro visible ha sido durante mucho tiempo la región donde, por razones de peso, predominan las técnicas de fotografía tradicionales. La luz visible, junto a las partes de una cámara (lente y sensor, entre otros), permiten extraer las características físicas que conforman y caracterizan al objeto de estudio, normalmente en espacios de color como podría ser el RGB (*Red-Green-Blue*). No obstante, las condiciones de visibilidad adversas como la noche, la niebla o la calima, dificultan en gran medida la captación de estos datos. Por otro lado, para aplicaciones como medicina, astronomía o incluso en la observación espacial de fauna y flora, la tecnología visible presenta algunas limitaciones, siendo incapaz de extraer ciertas características que podrían resultar de interés. En este aspecto, han surgido otro tipo de sensores que permiten, captando diferentes longitudes de onda, suplir estos problemas. Algunos ejemplos se basan en la tecnología infrarroja, multiespectral e hiperspectral, entre otras. Este TFM (Trabajo

de Fin de Máster) tiene su base teórica en la tecnología capaz de captar variaciones energéticas en las regiones del infrarrojo, más concretamente en el infrarrojo de onda larga (7.5 a 13.5 μm), también denotado como LWIR (*Longwave Infrared*).

1.1.1. Conceptos sobre el espectro infrarrojo y la termografía

La región infrarroja ha sido objeto de estudio durante muchos años debido a la capacidad que tienen estas longitudes de onda para cargar con la energía térmica de un cuerpo dado. Desde el punto de vista energético, todo cuerpo por encima del 0 absoluto o $-273.15\text{ }^{\circ}\text{C}$ emite este tipo de radiación [4], aunque en situaciones prácticas lo conveniente es que un cuerpo se encuentre a una temperatura mayor que la del ambiente para poder distinguirlo de su entorno. Por ello, tener en posesión una herramienta que permita la captación de esta energía se considera de gran utilidad dependiendo de la aplicación que se busque, especialmente en condiciones nocturnas, ya que suele expresar emisión y reflexión de calor. Por otro lado, al contrario que la luz visible, la energía infrarroja tiene unas longitudes de onda incapaces de distinguirse por el ojo humano, por ello los sensores tradicionales de las cámaras de luz visible no pueden ser usados.

Uno de los campos más estudiados que incumbe la tecnología infrarroja es la termografía, ligada con la ciencia de la termodinámica. Por un lado, la termodinámica se puede definir como la ciencia que permite modelar los efectos de cambios de temperatura, presión y volumen de un cuerpo determinado. En este sentido, las partículas de un cuerpo se mueven de forma desordenada, el estudio de estas partículas y el calor generado por su movimiento componen la base de esta ciencia. Por otro lado, la termografía se considera la técnica más usada para estimar valores de temperatura a distancia y sin necesidad de contacto directo con el objeto de estudio, gracias a la emisión pasiva de radiación infrarroja que manifiesta un cuerpo [5]. En este aspecto, las herramientas más utilizadas en el campo de la termografía son las cámaras termográficas.

La connotación de cámaras termográficas suele acarrear dos significados:

1. Estas cámaras, debido a la naturaleza del sensor que implementan, trabajan en la región del espectro infrarrojo denominado “infrarrojo térmico” que se corresponde con el rango del infrarrojo de onda larga. Sin embargo, también existen muchas

cámaras que trabajan en el rango del infrarrojo de onda media o MWIR (*Midwave Infrared*), abarcando otros ámbitos de aplicación.

2. Una cámara termográfica es capaz de captar la emisión infrarroja pasiva de un objeto dado sin necesidad de una excitación externa artificial. Esta característica convierte a las cámaras termográficas en un recurso valioso para la monitorización remota de temperatura.

El continuo desarrollo de la tecnología termográfica está impulsando su uso en diversos campos como la medicina, la vigilancia remota, el control de plantas industriales y el cuidado de plantaciones agrícolas, entre otros [4], [6]. Los sensores de estas cámaras se diseñan específicamente para captar radiación infrarroja de una determinada longitud de onda, filtrando aquellas otras que no sean de interés para la aplicación buscada. Siguiendo con esta línea, una de las condiciones necesarias para asegurar una correcta medición de la temperatura a una determinada distancia no radica solo en la fabricación y caracterización del sensor, sino en el grupo de características derivadas del conjunto óptica-sensor que conforman la cámara.

Este aspecto puede quedar resumido en los dos siguientes puntos:

- Por un lado, se tiene en consideración la resolución del sensor integrado en la cámara, que es el principal agente que determina la cantidad de píxeles con la que esta es capaz de representar el entorno. Cuanto mayor sea la resolución más precisión se obtendrán en las medidas, ya que se captan mayores detalles que terminan caracterizando a la escena.
- De la misma forma, también es necesario estudiar la fabricación de su óptica, cuyas características físicas modelan como incide la radiación infrarroja sobre el sensor. Por ejemplo, la lente que integra una cámara influye en la capacidad de enfoque de esta y la cantidad de área que es capaz de abarcar.

Es decir, según la configuración de la cámara que se esté usando, la representación de la escena observada queda definida de una forma específica, como se puede observar en la Figura 1.2 y de forma similar a lo que sucede con las cámaras de luz visible.

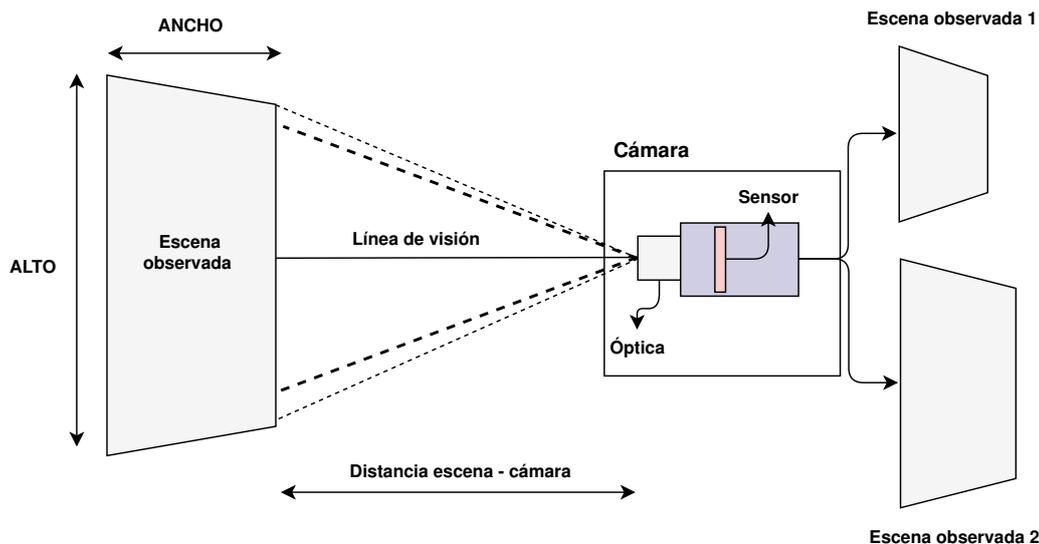


Figura 1.2: Influencia del conjunto óptica-sensor en una cámara

Respecto a la caracterización de la escena observada, las cámaras termográficas proporcionan la digitalización del flujo del campo electromagnético en el rango térmico, denominado en mucha bibliografía como cuentas térmicas. De forma inmediata, mediante aproximaciones de distintas ecuaciones matemáticas, se estima la temperatura de la superficie u objeto estudiado. Por ejemplo, en el caso del fabricante FLIR se puede obtener a partir de los términos de RBFO o similares [7]. Estos términos se implementan en una función de transferencia que transforman las cuentas térmicas en un valor concreto de temperatura.

Existen muchos factores que intervienen en el proceso de medida termográfica, por lo que es complicado asegurar que el valor que se está captando sea equivalente a la temperatura real de la escena observada [8]. Uno de los parámetros con mayor peso a la hora de establecer esta medida es la distancia entre la cámara y el blanco de estudio, algo complicado de modelar, porque afecta de forma diferente según las distintas configuraciones posibles de una cámara concreta. No obstante, en líneas generales se puede observar un continuo decaimiento en el valor del flujo electromagnético captado por una cámara termográfica a medida que aumenta la distancia, debido fundamentalmente al efecto de la atmósfera terrestre en la ventana del infrarrojo de onda larga [9]. Este es un factor que se nota en mayores proporciones sobre aquellas cámaras termográficas con una resolución de sensor menor, ya que la imagen termográfica queda conformada

por un menor número de píxeles. De la misma forma, la óptica que se integra en la cámara tiene mucho peso en los datos captados por esta, ya que concentra y encamina de forma particular la radiación infrarroja al sensor, dependiendo de las características físicas y fabricación de la lente. Por ejemplo, un gran angular, que es una óptica con un gran FOV (*Field Of View*), es capaz de abarcar mayor superficie en una iteración de medida. Mientras que un teleobjetivo, una óptica con un FOV reducido, se suele relacionar con aquellas lentes que enfocan muy bien el objeto de estudio, aunque este se encuentre a grandes distancias. Esta composición interviene en la cantidad de radiación que absorbe el objetivo (conjunto que incluye la lente, sistema de enfoque y obturación) de la cámara, y, por lo tanto, la cantidad de radiación que el sensor es capaz de integrar y convertir en una matriz de píxeles [10].

El proceso general de formación de una imagen termográfica, referenciado en la Figura 1.3, podría simplificarse en tres etapas:

1. Captación de la energía infrarroja por parte de la lente de la cámara.
2. Conversión del infrarrojo a valores digitales mediante un conversor ADC (*Analog-to-Digital Converter*). Dónde las referencias físicas son determinadas principalmente por el FOV.
3. Representación de estos valores digitales en formato matricial.

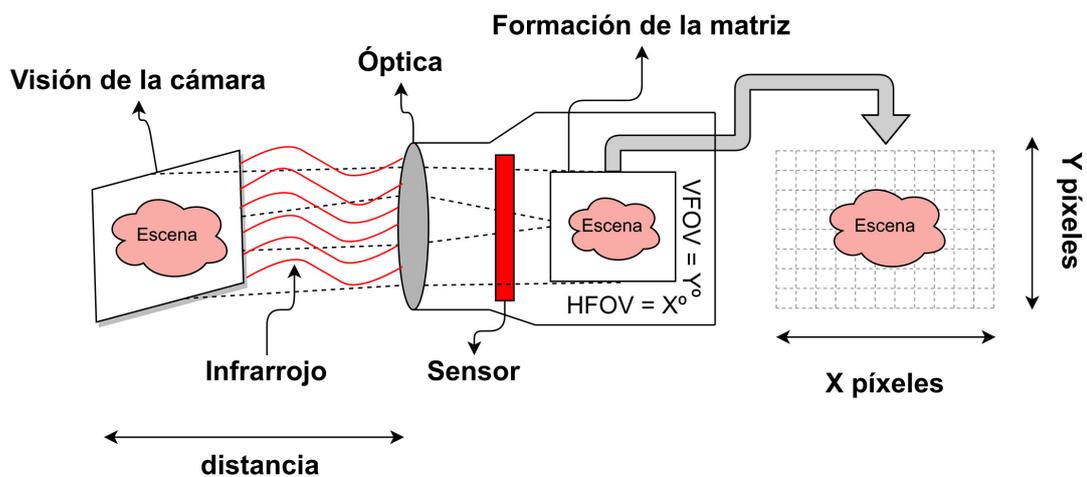


Figura 1.3: Representación básica del funcionamiento de una cámara termográfica

Respecto al ámbito de aplicación, las cámaras termográficas pueden implementarse en distintos campos de estudio, cada uno de ellos con requisitos y necesidades distintas. Esto se termina traduciendo en una oferta por parte de los fabricantes de cámaras termográficas bastante amplia, ya que dependiendo de la naturaleza de la demanda se fabrican cámaras con diferentes características hardware y software. Por este motivo, se puede optar por adquirir distintas gamas de cámaras en el mercado, cuyo precio varía enormemente dependiendo de las necesidades impuestas. Por ejemplo, cámaras con una resolución de sensor baja (80 x 60) tienden a costar menos que cámaras con una mayor resolución de sensor (640 x 480), ya que sus especificaciones y criterios de diseño son totalmente distintos, además de la óptica que tienen integrada. Sin embargo, no existen cámaras termográficas en el mercado que lleguen a la resolución actual de las cámaras de luz visible. Por lo tanto, cuando se habla de resoluciones altas en el campo de la termografía se suele referir a estándares como VGA (*Video Graphics Array*) más que otros formatos superiores. De hecho, la mayoría de las cámaras termográficas del mercado no alcanzan 1 megapíxel de resolución y aquellas que si lo alcanzan suelen tener un precio demasiado alto para ser accesible al público general [11], [12].

De este modo, debido a la longitud de onda larga y la baja resolución del sensor que implementan las cámaras termográficas, las imágenes que se generan suelen ofrecer pérdida en los detalles de la textura de un objeto, pérdida de contexto y pérdida de la información en los bordes, entre otros [13]. A pesar de la innovación tecnológica actual en el campo de la termografía, principalmente en la fabricación de sensores debido a la alta demanda en los últimos años, la capacidad de generar una imagen termográfica sigue sufriendo de una limitación relacionada con la resolución espacial del sensor, siendo esta menor si se compara con imágenes de las cámaras tradicionales de luz visible.

Desde un punto de vista hardware, incrementar el número de píxeles por área suele ser una tarea difícil por dos razones [14]:

1. Disminuir el tamaño del píxel degradaría la relación señal/ruido, puesto que una menor cantidad de señal sería registrada por este, aumentando el ruido de disparo en consecuencia (*shot noise*). Por otro lado, se ha demostrado que los píxeles de

menor tamaño (en relación con el tamaño de la apertura de la lente) son más sensibles a los efectos de difracción en comparación con los píxeles de mayor tamaño.

2. Con la intención de lograr una medición de datos precisa, los sensores infrarrojos suelen encapsularse en paquetes individuales al vacío, lo que supone un proceso largo y con mucho coste asociado.

Debido a estas limitaciones, el coste de una cámara termográfica es comparativamente más elevado que el de una cámara de luz visible con una resolución similar. Por este motivo, se dice que existe una limitación a la hora de fabricar sensores infrarrojos con una densidad de píxeles alta, también llamados IRFPA (*Infrared Focal Plane array*) [15], derivando estos en un ciclo de diseño largo y caro [16].

En líneas generales, una cámara termográfica de bajo coste y poca resolución no es capaz de ofrecer una cantidad de detalles comparable a otra con mayor resolución y alto coste, ya sea desde el punto de vista de la cantidad de píxeles que describen la escena como por el valor radiométrico de estos píxeles. Este hecho es aún más apreciable cuanto mayor es la distancia que existe entre la cámara y el blanco de estudio. Una solución a este problema podría radicar en el cambio de la óptica que viene por defecto en la cámara, garantizando una mejora en los resultados finales si se utiliza una lente de altas prestaciones. No obstante, este acercamiento tiene un defecto, el cambio de una óptica afecta directamente a la calibración de la cámara termográfica por parte de su fabricante. Esto implica no solo un proceso delicado de diseño e integración de una lente, sino también una metodología propia y específica de calibración. Por este motivo, se continúa en la búsqueda de soluciones que permitan mejorar el desempeño general de las cámaras de bajo coste mediante la incorporación de metodologías software, disminuyendo de forma significativa el coste de adquisición de estas para futuros proyectos y mejorando así los resultados finales de las medidas.

Entre las soluciones más competitivas actualmente se encuentra el uso de técnicas de súper-resolución. La súper-resolución es una técnica muy utilizada en el campo de la luz visible, sobre todo en las tres últimas décadas [17]. Este concepto se relaciona con la generación de una imagen de alta resolución a partir de una o varias imágenes de

baja resolución. Una de las soluciones que plantea esta técnica es la compensación de los detalles visuales de alta frecuencia que se pierden completamente en imágenes de baja resolución [17], [18]. Se debe reseñar que la súper-resolución no debe confundirse con la interpolación clásica, la restauración o el renderizado de imágenes, ya que estas técnicas no permiten recuperar las componentes de alta frecuencia de las imágenes [19].

1.1.2. El concepto de súper-resolución

El principio fundamental de la súper-resolución se basa en un concepto sencillo, este es el de diseñar e implementar una función capaz de transformar una imagen a su entrada con una resolución $a \times b$ en una imagen de mayor resolución $A \times B$. Sin embargo, distintos métodos de súper-resolución pueden ser aplicados a las imágenes, dependiendo fundamentalmente del dominio de estudio.

Según el dominio de estudio se puede encontrar [17]:

- **Dominio frecuencial.** Las imágenes de baja resolución son llevadas al dominio de la frecuencia y luego se estima la imagen de súper-resolución en este dominio. Por último, se vuelve a transformar la imagen de súper-resolución reconstruida al dominio espacial. Dependiendo de la transformación empleada para convertir las imágenes al dominio de la frecuencia, estos métodos se dividen generalmente en métodos basados en la transformada de Fourier y métodos basados en la transformada de Wavelet.
- **Dominio espacial.** Donde se agrupa la gran mayoría de métodos existentes. Estos métodos se estiman en función del número de observaciones disponibles, los métodos de súper-resolución espacial pueden dividirse generalmente en dos grupos:
 - **Métodos basados en una sola imagen.** En este enfoque, la relación entre las imágenes de baja resolución y las correspondientes imágenes de alta resolución se aprende de una base de datos de pares de imágenes de baja y alta resolución conocidas [20].
 - **Métodos basados en múltiples imágenes.** Consiste en la reconstrucción de una imagen de alta resolución partir de varias iteraciones de imágenes de

baja resolución con distintos desplazamientos geométricos y condiciones de medida.

A su vez, los algoritmos que se aplican en estos métodos suelen ser:

- **Algoritmos basados en aprendizaje.** Donde proliferan los algoritmos de aprendizaje profundo.
- **Algoritmos basados en reconstrucción de imágenes.** Muy usados en los métodos de múltiples imágenes, puesto que necesitan una gran cantidad de imágenes con cierto desplazamiento geométrico para poder reconstruir una imagen de alta resolución.

Un esquema resumen del estado de la técnica en cuestiones a súper-resolución se presenta en la Figura 1.4.

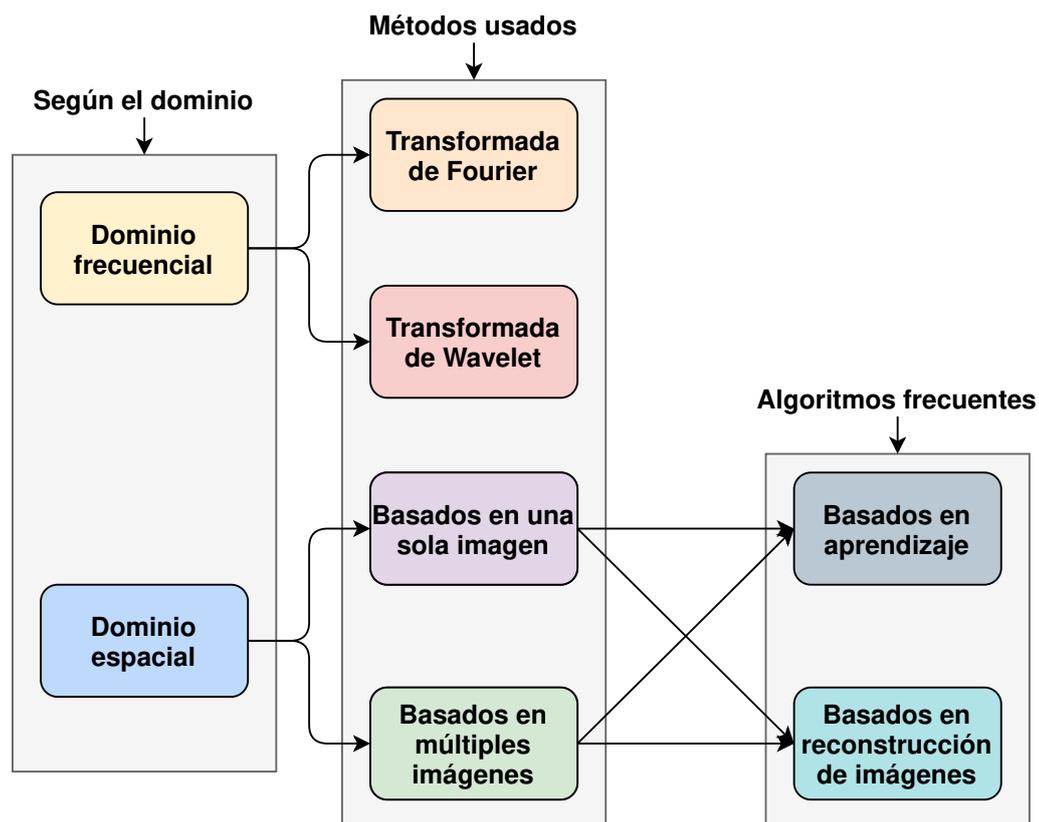


Figura 1.4: Esquema simplificado del estado de la técnica en cuestión a súper-resolución

De esta manera y con el paso del tiempo, las técnicas de súper-resolución han evolucionado de forma constante para otros ámbitos de trabajo como pueden ser:

- **La vigilancia.** Donde su mayor problema es que los escenarios estudiados no son

estáticos y se necesita corrección constante para el desenfoque del movimiento [21]. La súper-resolución es de gran importancia en este aspecto, puesto que posibilita un mejor reconocimiento de patrones.

- **Observación de la tierra vía satélites.** Donde las técnicas de súper-resolución ayudan a dar un nuevo enfoque tanto a las imágenes hiperespectrales como multi-espectrales de satélites [22], existiendo ejemplos de aplicación a la serie de satélites Landsat [23].
- **Diagnosis médica.** Donde los campos de aplicación se cuentan por muchos, tanto para el estudio de la anatomía general como para resonancias magnéticas, entre otros. La idea base de la aplicación de súper-resolución en medicina se basa en el hecho de aumentar la precisión en las medidas preservando la imagen isotrópica en 3D [24], [25].
- **Observaciones astronómicas.** Donde la capacidad de la óptica del telescopio se ve limitada a partir de ciertas distancias. Algunos estudios han planteado técnicas básicas de súper-resolución para aumentar la precisión y calidad visual de las imágenes del cielo estrellado [26] e incluso para aumentar la resolución de imágenes de la superficie de la luna [27].

El ámbito de aplicación que compete a este TFM se encuentra en la termografía. En la actualidad, existen varias aproximaciones para el uso de súper-resolución en el estudio de imágenes termográficas, estos estudios son normalmente enfocados a cámaras de baja resolución espacial y bajo coste. Por ejemplo, Piotr Wieçek y Dominik Sankowski proponen un método propio de súper-resolución basado en aprendizaje [28], más concretamente implementando su propia *irRCNN* (*infrared Residual Convolutional Neural Network*) y ofreciendo métricas como la relación PSNR (*Peak-Signal-to-Noise Ratio*) y el SSIM (*Structural Similarity Index Measure*), comparando las imágenes originales con las resultantes de aplicar el método. El aprendizaje del modelo de red neuronal propuesto y su desempeño se observan a través de distintas gráficas donde las anteriores métricas son usadas respecto el número de muestras de entrenamiento que el modelo procesa. En estos experimentos se usó una base de datos de más de 800 imágenes de alta resolución (2040 x 1404 y 1024 x 1356) en formato RGB.

Uno de los resultados que ofrecen los autores del uso de su método se observa en la Figura 1.5, donde:

- La parte izquierda muestra la superficie de estudio, mientras que, a la derecha, se muestran dos imágenes termográficas.
- La primera empezando por la izquierda representa la imagen original (160 x 120) y la segunda muestra el resultado de aplicar el método propuesto, una imagen de súper-resolución con un factor de escala x6 (960 x 720).

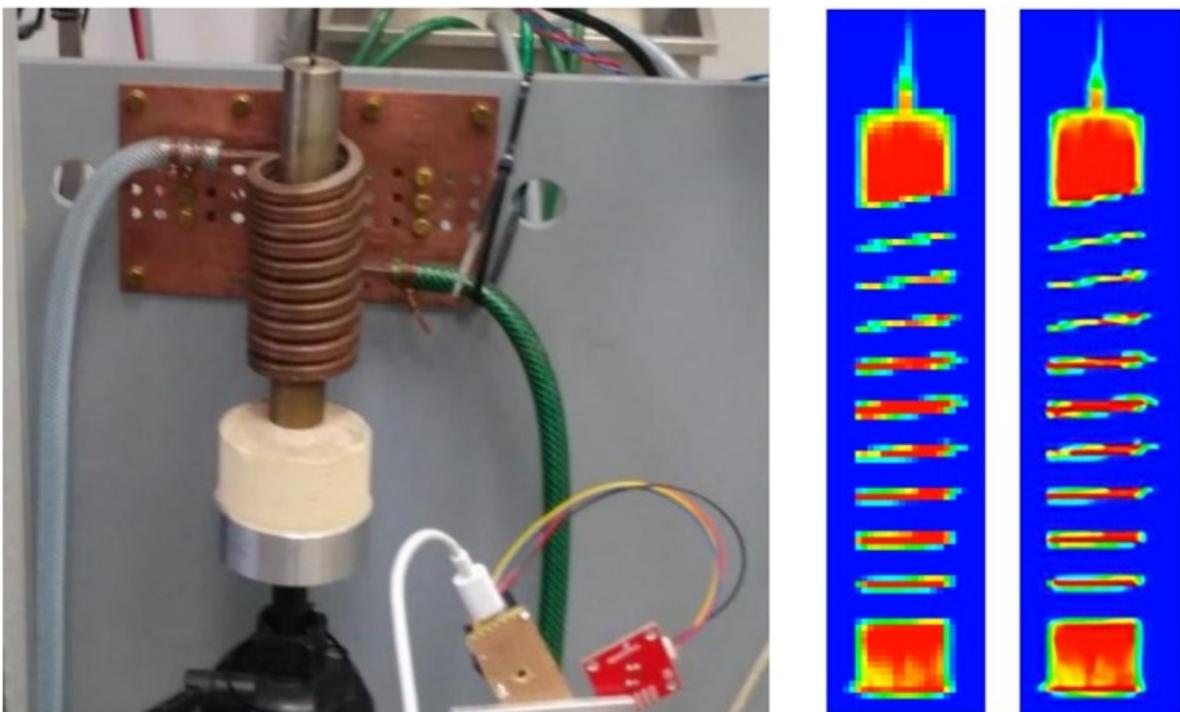


Figura 1.5: a) Superficie de estudio, una bobina inductiva sobre una pieza de latón. b) Imagen termográfica original e imagen de súper-resolución. Extraída de: [28]

Siguiendo con esta línea, otro de los artículos consultados plantea una red convolucional multinivel basada en aprendizaje supervisado para aplicar súper-resolución en distintas escalas [29]. Este aprendizaje supervisado se utiliza para asegurar, durante el proceso de entrenamiento, la similitud entre la imagen original y la imagen resultante de aplicar el modelo de red neuronal. Más concretamente, en este artículo se referencian tres modelos de redes neuronales entrenados, capaces de obtener imágenes de alta resolución en tres escalas distintas (x2, x3, x4) y usando métricas como la PSNR y el SSIM. En este acercamiento se usaron dos conjuntos de datos distintos, uno que

servía fundamentalmente como base de datos de distintas imágenes termográficas de varias resoluciones (baja 160 x 120, media 320 x 240 y alta 640 x 480) destinadas al entrenamiento y otro cuya finalidad residía en probar la robustez de la estructura y el método propuesto. Los resultados mostrados por los autores plantean que el uso de un aprendizaje supervisado a la hora de aplicar métodos de súper-resolución mejoran los valores obtenidos respecto a cuando se aplican este tipo de estrategias, al menos en los estudios realizados durante el artículo.

Finalmente, en distintas referencias revisadas también se han encontrado aplicaciones de súper-resolución termográfica [30], [31]. No obstante, el factor de mérito se sigue caracterizando por métricas como la PSNR, el SSIM y el RMSE (*Root Mean Squared Error*), entre otras. Sin embargo, no se encuentra ninguna referencia directa a la calidad radiométrica de los píxeles. Es decir, no se ofrecen evidencias de que se mejore la precisión de los valores radiométricos captados por la cámara a través de un procesado software, teniendo como modelo de referencia una cámara termográfica de altas prestaciones y una resolución alta.

La propuesta que se presenta para este TFM recoge las ideas base de las referencias analizadas anteriormente, incluyendo además nuevas variables en el estado de la técnica, como podría ser la distancia entre cámara y blanco. Por un lado, se realizará una búsqueda de distintos métodos de súper-resolución presentes en el estado de la técnica, analizándolos y aplicándolos a imágenes termográficas de una base de datos propia. Por otro lado, a estas imágenes SR (*Superresolved*) se les aplicará una red neuronal capaz de mejorar el valor de los píxeles resultantes del proceso de súper-resolución anterior, teniendo en cuenta que estos píxeles son equivalentes al valor de temperatura del cuerpo estudiado. Es decir, se pretende estudiar, para distintas distancias de medida, el desempeño de un método de súper-resolución propio, introduciendo el término de distancia como un parámetro más y obteniendo un rango de operación estimado. De la misma manera, se compararán las imágenes obtenidas de aplicar súper-resolución (previamente imágenes de baja resolución) con otras imágenes de cámaras termográficas tomadas en el mismo intervalo temporal. Un ejemplo de esto podría ser la comparación de una imagen termográfica de 640 x 480 con otra imagen de 640 x 480 cuya resolución original era menor. De esta forma, la métrica que marcaría el desempeño de un método

de súper-resolución concreto, además de las comentadas anteriormente, podría ser la similitud entre los valores de píxeles (en $^{\circ}\text{C}$) de estas dos imágenes.

1.2. Objetivos

Este TFM tiene como objetivo fundamental el estudio e implementación de métodos de súper-resolución basados en aprendizaje profundo que generen imágenes de alta resolución a partir de imágenes de baja resolución obtenidas de distintas cámaras termográficas. El funcionamiento de este tipo de métodos se basa en obtener, a partir de una imagen de baja resolución espacial (x) y de una transformación determinada (h), una imagen de mayor resolución (y), tal como se ilustra en la Figura 1.6. Por otro lado, se pretende incluir en el estado de la técnica de la súper-resolución de imágenes termográficas un nuevo término corrector de temperatura. De esta forma, no solo se espera aumentar el número total de píxeles que integran un blanco de estudio, sino que su valor radiométrico se ajuste lo más posible a la realidad. Para ello se planteará un modelo de red neuronal determinado que aprenda el mapeo entre los datos de entrada y salida, además de una base de datos de imágenes tomadas por varias cámaras termográficas, obteniendo una referencia clara de detalles y valores radiométricos con los que entrenar la red neuronal.

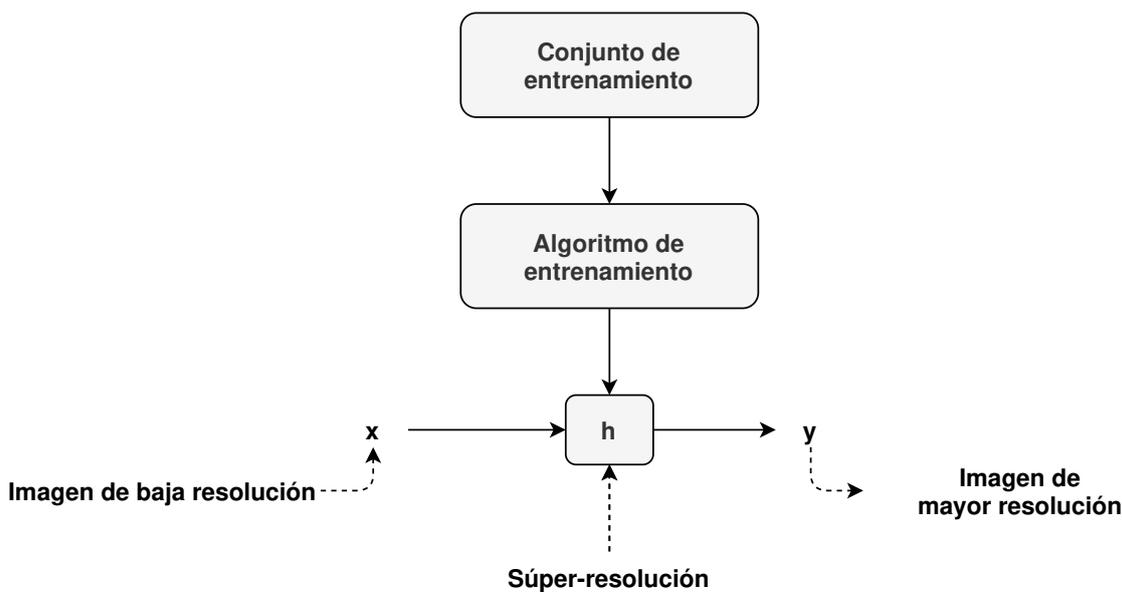


Figura 1.6: Simplificación de un método de súper-resolución

Por ello, el hilo conductor de este TFM tendrá sus bases en cinco puntos fundamentales:

- O1. Estudio y evaluación de distintos métodos de súper-resolución aplicables a imágenes de cámaras de luz visible y cámaras termográficas.
- O2. Análisis y preparación de distintos escenarios para la captación de imágenes termográficas respecto a un foco emisor determinado.
- O3. Creación de una base de datos compuesta por imágenes termográficas.
- O4. Diseñar, entrenar y evaluar la red neuronal propuesta para unos entornos controlados, además de una posterior discusión de los resultados.
- O5. Documentar y presentar el trabajo realizado.

1.3. Motivación

La motivación principal de este TFM es la consecución del título de Máster Universitario en Ingeniería de Telecomunicación que habilita profesionalmente para ejercer la profesión de Ingeniero en Telecomunicación y permite acceder al mercado laboral con dicha acreditación oficial.

La motivación didáctica de este proyecto es el estudio de distintos métodos de súper-resolución, con la intención de implementar uno o varios métodos basados en aprendizaje profundo capaces de aumentar la resolución espacial de una cámara termográfica de bajo coste, mejorando de la misma forma la precisión del valor radiométrico de cada uno de sus píxeles.

1.4. Estructura de la memoria

Este documento está organizado en una memoria descriptiva, bibliografía, presupuesto y anexos. A continuación, se desglosan los puntos que conforman el documento:

- Capítulo 1. Breve introducción del tema, exponiendo los antecedentes y objetivos, además de la motivación que ha iniciado este trabajo.

- Capítulo 2. Principios teóricos que sustentan y basan a la termografía infrarroja, las cámaras termográficas, la súper-resolución y las redes neuronales.
- Capítulo 3. Estado de la técnica de los distintos métodos de súper-resolución espacial basados en redes neuronales.
- Capítulo 4. Preparación de las bases de datos que se usarán en este TFM para el entrenamiento de las redes neuronales y del protocolo de medidas seguido para la creación de las bases de datos que se destinarán a su evaluación.
- Capítulo 5. Propuestas para la súper-resolución espacial y radiométrica, el entrenamiento de las redes neuronales y exposición teórica de los distintos experimentos que se realizarán para evaluar los métodos de súper-resolución, además del procesado específico seguido en cada uno.
- Capítulo 6. Presentación de los resultados y discusiones a partir de estos.
- Capítulo 7. Conclusiones de este trabajo y líneas futuras que se plantean.
- Anexos. Esta memoria tiene dos anexos:
 - El anexo A presenta algunos resultados visuales de aplicar los métodos de súper-resolución espacial nombrados en este TFM para dos cámaras termográficas diferentes.
 - El anexo B incluye el diseño e implementación de un demostrador que une los distintos métodos de súper-resolución espacial basados en aprendizaje profundo nombrados en este TFM, además de la red neuronal propuesta para la corrección de temperatura.

Capítulo 2

Marco teórico

2.1. Termografía

La termografía es una técnica que permite estimar la temperatura de un cuerpo dado a cierta distancia y sin necesidad de contacto directo con este, gracias a la captación de la intensidad de radiación infrarroja que emiten los cuerpos al encontrarse por encima del 0 absoluto o $-273.15\text{ }^{\circ}\text{C}$ [4]. El uso de esta técnica para estudiar la temperatura permite detectar diferencias térmicas muy pequeñas en la oscuridad, a través de niebla, humo y polvo, entre otras condiciones atmosféricas adversas [32]. Para definir las bases de la termografía es necesario explicar el concepto de cuerpo negro, una idea muy presente en el estudio termográfico a lo largo del tiempo.

La radiación es la continua emisión de energía desde la superficie de cualquier cuerpo. Un cuerpo negro es un concepto teórico de radiador ideal, una idea establecida fundamentalmente por el físico Gustav Robert Kirchoff. Este se puede definir como un cuerpo capaz de absorber todas las longitudes de onda que conforman la radiación incidente a este y que, de la misma manera, emite radiación con una igual eficiencia, incluso en el vacío. Un cuerpo negro absorbe toda la energía incidente e impide su reflexión, por lo que es una premisa muy presente en la teoría física detrás de la termografía.

Existen cuatro leyes fundamentales que describen la forma de radiación de un cuerpo [4], [32]:

- **La ley de radiación térmica de Kirchoff.** Esta ley tiene su base en el de-

nominado equilibrio térmico, un estado donde toda la radiación absorbida por un cuerpo es igual a la emitida. Gustav Robert Kirchoff formuló una serie de postulados en este aspecto:

1. El cociente entre la emisividad y absortividad de un cuerpo es una cantidad independiente a la naturaleza del material, dependiendo únicamente de la temperatura a la que se encuentre este. Es decir, se estima que un cuerpo es mejor emisor de radiación según su eficiencia al absorberla.
 2. En equilibrio térmico, un cuerpo emite y absorbe radiación con una emisividad ϵ_λ igual a su absortividad α_λ , donde la emisividad no puede ser mayor a 1 por la ley de conservación de la energía. En la nomenclatura usada, λ implica que los coeficientes de emisividad y absortividad dependen de la longitud de onda.
- **Ley de radiación de Planck.** Max Planck fue capaz de modelar la función de distribución espectral de la radiación de un cuerpo negro a una temperatura determinada. La intuición detrás de la ley de Planck es que la curva que representa esta función denota que un cuerpo en equilibrio térmico emite radiación en todo el conjunto de sub-bandas del espectro electromagnético. Esta distribución espectral se modela según la ecuación 2.1.

$$W_{b\lambda} = \frac{2\pi \cdot h \cdot c^2}{\lambda^5 \cdot \left(e^{\frac{hc}{\lambda K_B T}} - 1 \right)} \quad (W/m^2) \quad (2.1)$$

Las constantes de esta ecuación son:

- La constante de Planck, $h = 6.6 \cdot 10^{-34} J \cdot s$.
 - La constante de Boltzmann, $K_B = 1.38 \cdot 10^{-23} J/K$.
 - La velocidad de la luz, $c = 3 \cdot 10^8 m/s$.
- **Ley de desplazamiento de Wien.** La materia emite radiación a lo largo del espectro electromagnético, pero se ha demostrado que no lo hace en la misma cantidad para todas las longitudes de onda que lo componen. El pico de emisión en una determinada longitud de onda λ es inversamente proporcional a la temperatura del cuerpo emisor. Cuanto mayor sea la temperatura de un cuerpo, la

longitud de onda asociada a la energía radiada se desplaza hacia longitudes de onda más cortas. La ley de Wien define que, en equilibrio térmico, el producto de la temperatura y la longitud de onda donde se consigue la máxima potencia emitida siempre es constante, expresándose como $T \cdot \lambda = 2.898 \cdot 10^{-3} \text{ m} \cdot \text{K}$. De esta forma, la longitud de onda donde la radiación de un cuerpo será máxima se puede calcular según la ecuación 2.2.

$$\lambda_{max} = \frac{0.0028976 \text{ m} \cdot \text{K}}{T} \quad (2.2)$$

Donde T es la temperatura del cuerpo negro en Kelvin y λ_{max} es la longitud de onda del pico de emisión en metros.

- **Ley de Stefan-Boltzmann.** De forma teórica, al integrar toda el área bajo la curva que describe la función de Planck, es posible obtener la potencia total por unidad de área emitida por un cuerpo en todas las longitudes de onda. La ley de Stefan-Boltzman propone integrar la función de Planck a través de todas las longitudes de onda, eliminando así la dependencia de estas. Con esta propuesta, se demuestra que el total de potencia emitida por un cuerpo es directamente proporcional a la cuarta potencia de su temperatura (T^4) multiplicada por una constante física σ , que depende de h , c y K_B vistas en la ley de Planck. Esta ley permite ilustrar de forma intuitiva que dependencia existe entre la temperatura de un cuerpo y la potencia por unidad de área que este emite, como se muestra en la ecuación 2.3.

$$W_b = \sigma \cdot T^4 \text{ (W/m}^2\text{)} \quad (2.3)$$

Donde $\sigma = 5.67 \cdot 10^{-8} \text{ (W/m}^2 \cdot \text{K}^4\text{)}$.

La idea teórica de cuerpo negro sustenta el cálculo de la potencia radiante de un objeto. No obstante, en la realidad no existen los cuerpos negros, por ello se deben tener en cuenta los términos de reflexión y transmisión existentes en todo proceso de transferencia térmica. De esta manera, se entiende que los cuerpos reales no son capaces de emitir radiación con la misma eficiencia que un cuerpo negro. Se estima que para que exista conservación de la energía se debe cumplir la igualdad de la ecuación 2.4.

$$\alpha_\lambda + \rho_\lambda + \tau_\lambda = 0 \quad (2.4)$$

Donde los términos mostrados son respectivamente la absorptividad, reflectividad y transmisividad de un determinado cuerpo.

Teniendo en cuenta las características del cuerpo estudiado, es posible estimar si será un buen emisor de radiación a partir de los factores nombrados anteriormente. La emisividad de un cuerpo es la relación que existe entre la energía emitida por este frente a la radiación que emitiría un cuerpo negro a la misma temperatura. Este factor se muestra en la ecuación 2.5.

$$\epsilon_\lambda = \frac{W_\lambda}{W_{\lambda B}} \quad (2.5)$$

Donde el factor del numerador es la energía emitida por el cuerpo estudiado mientras que el denominador denotaría la energía emitida por un cuerpo negro.

La emisividad es dependiente de la longitud de onda y la temperatura a la que se someta un cuerpo. Sin embargo, los sensores actuales suelen operar en una región muy estricta del espectro electromagnético. Por este motivo, una aproximación muy usada es plantear este factor como una constante $\epsilon_\lambda = \epsilon$, a los cuerpos donde se toma esta estimación se les llama cuerpos grises. Sin embargo, se debe recalcar que en realidad los cuerpos son radiadores selectivos, su emisividad cambia dependiendo de la longitud de onda y temperatura. De esta manera, el cálculo de la radiación de un cuerpo no negro al introducir el factor de emisividad puede expresarse según la ecuación 2.6.

$$W_b = \epsilon_\lambda \cdot \sigma \cdot T^4 (W/m^2) \quad (2.6)$$

El flujo de radiación que emite un cuerpo recorre cierta distancia a través de la atmósfera. Esto, irremediablemente, deriva en una atenuación percibida por parte del sistema de captación. Un buen modelado de estos factores permite una corrección más precisa a la hora de calcular la temperatura de un objeto.

Los factores más comunes que suelen intervenir en el proceso de medida termográfica suelen ser [8], [9]:

- **La emisividad.** Es decir, la eficiencia con la que un cuerpo emite radiación.
- **La reflectividad.** Un sistema de captación no solo integra la señal que le llega del objeto, sino de toda una serie de reflexiones recibidas por parte de su entorno que deben ser ponderadas. Este factor, al igual que la emisividad, es multiplicativo y oscila entre 0 y 1.
- **Temperatura del entorno.** La integración total de un sensor toma en cuenta no solo la radiación de un cuerpo a una determinada temperatura, sino también las distintas radiaciones por parte de cuerpos que se encuentran a una temperatura ambiente.
- **Transmisividad de la atmósfera.** La atmósfera atenúa el paso de cualquier señal electromagnética debido fundamentalmente a su temperatura, densidad del aire y humedad relativa, entre otros parámetros. La transmisividad es otro factor que oscila entre 0 y 1 capaz de cambiar de forma notable los datos que le llegan a un dispositivo de medida. Sobre todo, cuanto mayor es la distancia entre el cuerpo estudiado y la cámara, puesto que la energía infrarroja recorre más distancia.

Al establecer una caracterización básica del entorno donde se sitúa el cuerpo que se desea estudiar, sale a la luz la capacidad de cada uno de estos agentes para condicionar las medidas finales de un sistema de captación. Por ejemplo, la aportación total que llega a una cámara termográfica puede ser dividida en 3 partes, como se referencia en la ecuación 2.7.

$$W_{total} = W_{obj} + W_{amb} + W_{atm} \quad (2.7)$$

Donde W_{obj} es la radiación debida a la emisión del objeto, W_{amb} representa la radiación que se refleja en el objeto debida a la temperatura del entorno y W_{atm} es el resultado de la componente radiométrica debida a la temperatura de la atmósfera.

Un despiece profundo de estos factores se presenta en [33] y se puede resumir en los siguientes puntos:

- La radiación emitida por el objeto ya ha sido descrita con anterioridad. El factor W_{obj} es el producto de la constante σ por ϵ_{obj} y T_{obj}^4 .

- Los cuerpos no negros tienen una reflectividad mayor que 0, por ello parte de la radiación emitida en su entorno es reflejada. Este coeficiente de reflexión se puede denotar como $\rho_{obj} = 1 - \epsilon_{obj}$.
- Por otro lado, se debe añadir el factor correspondiente a la propia radiación de la atmósfera y el término atenuante que introduce esta en el modelado del entorno. Esta relación se puede escribir como $\epsilon_{atm} = 1 - \tau_{atm}$.

De esta forma, una manera de parametrizar numéricamente estos hechos se muestra en la ecuación 2.8 y se representa visualmente en la Figura 2.1.

$$\begin{cases} W_{obj} = \epsilon_{obj} \cdot \tau_{atm} \cdot \sigma \cdot (T_{obj})^4 \\ W_{atm} = (1 - \tau_{atm}) \cdot \sigma \cdot (T_{atm})^4 \\ W_{amb} = (1 - \epsilon_{obj}) \cdot \epsilon_{amb} \cdot \tau_{atm} \cdot \sigma \cdot (T_{amb})^4 \end{cases} \quad (2.8)$$

La temperatura del objeto T_{obj}^4 podría ser despejada de la ecuación resultante. Con esta aproximación se describe en gran parte cual es la densidad total de potencia que emite un cuerpo para todas las longitudes de onda. No obstante, es usual que si el sistema de medidas opera en un rango concreto del espectro, como LWIR, se suele optar más por la fórmula que ofrece la ley de Planck o derivados de esta.

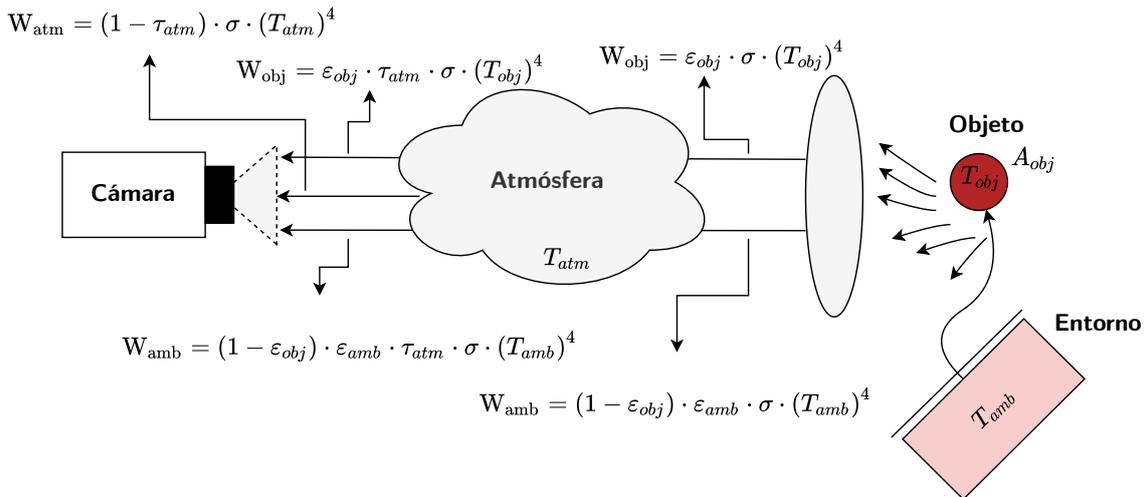


Figura 2.1: Efectos de la atmósfera en las medidas de una cámara termográfica

2.2. Cámaras Termográficas

Las cámaras termográficas se han convertido en los dispositivos por excelencia a la hora de captar y mostrar una representación bidimensional de la radiación térmica infrarroja emitida por un objeto. De forma clásica, los detectores capaces de captar radiación infrarroja se clasifican en detectores de fotones y detectores térmicos. Dentro de los segundos, los detectores térmicos no refrigerados son los más utilizados en las cámaras termográficas debido a sus pequeñas dimensiones, bajo consumo de energía, alta rentabilidad, además de una gran respuesta espectral y funcionamiento a largo plazo [34]. Los detectores infrarrojos no refrigerados se componen por un número reducido de sensores, en ocasiones sólo uno. Sin embargo, contar con un sólo sensor resulta inadecuado para aplicaciones donde se busque una imagen térmica, ya que se necesita de un sensor 2D, también conocido como FPA (*Focal Plane Array*), para la conformación de la imagen. En este sentido, existen tres tipos de tecnologías principales de FPA no refrigerado: sensores de microbolómetro [35], [36], sensores de termopila y sensores piroeléctricos [37]. En la actualidad, la sensibilidad de los sensores de microbolómetro ha demostrado ser mayor que los otros dos, contando además con un proceso de fabricación más sencillo [38].

2.2.1. Consideraciones sobre el sensor

Un microbolómetro es un dispositivo pasivo cuya resistencia varía según los cambios de temperatura. Estos cambios se transforman en una tensión medible a través de un conversor ADC. Un bosquejo de la estructura típica de un detector basado en microbolómetro se ilustra en la Figura 2.2. En esta figura, se puede observar como la radiación infrarroja incide sobre un material absorbedor, aumentando la temperatura del material suspendido sobre el puente aislado térmicamente, generando de esta forma un cambio en su resistencia interna y relacionado con el parámetro TCR (*Temperature Coefficient of Resistance*) o también conocido como α . Este cambio de valor de resistencia se transfiere eléctricamente al ROIC (*Readout Integrated Circuit*) para un posterior procesado.

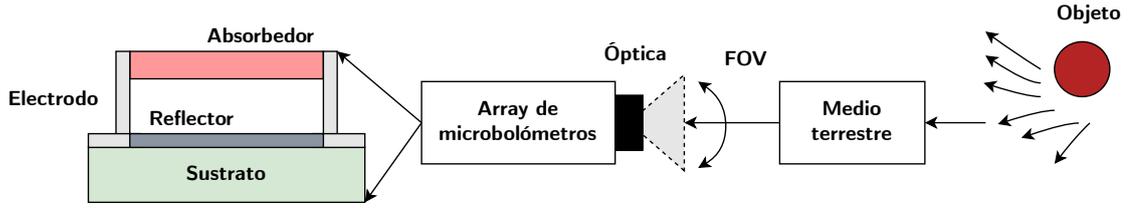


Figura 2.2: Funcionamiento básico de un detector de microbolómetro. Adaptada de [39]

Siguiendo con lo comentado anteriormente, el TCR se considera como una de las figuras de mérito que definen el rendimiento de un microbolómetro y puede ser definido según [40] por la ecuación 2.9.

$$\alpha = \frac{1}{R} \frac{dR}{dT} \quad (2.9)$$

Donde α es el TCR, R es la resistencia del microbolómetro y T es la temperatura. En este caso, α puede entenderse como la sensibilidad de la resistencia a los cambios de temperatura. De esta forma, valores grandes de α sugieren que un pequeño cambio de temperatura en el material sensor dará lugar a un mayor cambio en el valor de la resistencia.

La siguiente figura de mérito que caracteriza a un microbolómetro es la respuesta a los cambios de tensión R_v , que puede ser definida según [40] a partir de la ecuación 2.10.

$$R_v = \frac{I_b \alpha R \zeta}{G \sqrt{1 + \omega^2 \tau^2}} \quad (2.10)$$

Donde I_b es la corriente de polarización, ζ es el coeficiente de absorción del material, G es la conductancia térmica entre el detector y el sustrato, ω es la frecuencia angular y τ es el tiempo de respuesta térmica.

Finalmente, la tercera figura de mérito es la detectividad D^* , que proporciona la relación entre la respuesta del sensor y el ruido por unidad de ancho de banda [40]. Esta queda modelada por la ecuación 2.11.

$$D^* = \frac{R_v \sqrt{A_{eff} \Delta f}}{V_n} \quad (2.11)$$

Donde Δf es el ancho de banda del ruido del sensor, V_n es el ruido cuadrático medio total del sensor y A_{eff} es el área efectiva donde incide la radiación infrarroja.

2.2.2. Consideraciones sobre la óptica

Como ocurre con cualquier dispositivo de imagen, el tamaño proyectado del blanco estudiado depende de la resolución de la cámara, su FOV y la distancia [39]. El campo de visión o FOV de una cámara termográfica describe el área localizada y modelada finalmente por el sensor, como se referencia en la Figura 2.3.

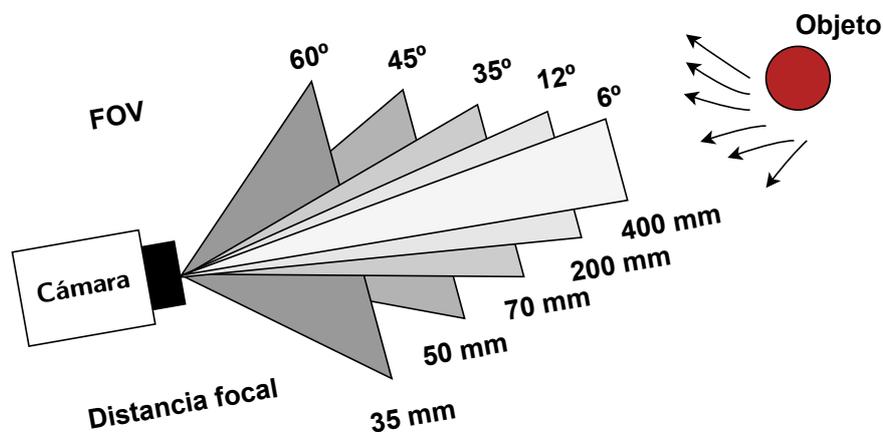


Figura 2.3: Representación de la correlación entre FOV y distancia focal

En situaciones donde sea deseable que la distribución de píxeles para un área determinada sea lo más precisa posible, se desea un factor de FOV pequeño, ya que de lo contrario se tendrá una mayor dispersión de los píxeles, necesitando de una mayor resolución del sensor para conseguir la misma precisión en los datos. Siguiendo con la misma línea, la distancia focal f de una cámara se considera inversa al FOV, representando esta la distancia entre la lente de la cámara y el punto donde todos los rayos captados por la óptica convergen, en este caso el sensor.

La relación entre FOV y distancia focal queda detallada de la siguiente forma:

- Distancias focales cortas: campo de visión amplio, menor enfoque del blanco que se desea estudiar.
- Distancias focales largas: campo de visión estrecho, mayor enfoque del blanco que se desea estudiar.

Un ejemplo ilustrativo se muestra en la Figura 2.4, donde se observan dos imágenes tomadas por cámaras de distinto FOV, la primera con 50° y la segunda con 18° .

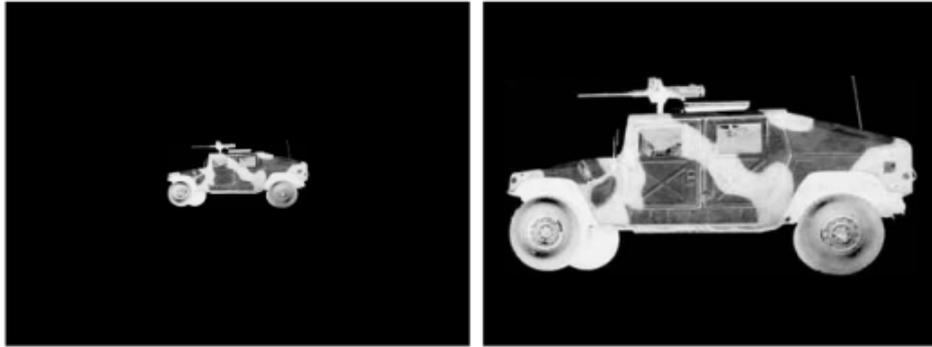


Figura 2.4: Representación de distintos tipos de FOV con la herramienta Fieldometer

2.3. Súper-resolución

El campo de la súper-resolución se encuentra en continua evolución y ya cuenta con una gran cantidad de publicaciones científicas en el panorama actual, ofreciendo soluciones transversales a distintas ramas del conocimiento. Este tipo de técnicas tienen como objetivo incrementar el nivel de píxeles y los detalles que caracterizan una imagen, aumentando la calidad visual de estas. Más concretamente, la súper-resolución es una idea que nace a partir de las tradicionales limitaciones físicas en los dispositivos empleados para capturar y extraer características de una escena observada, como pueden ser las cámaras de foto o vídeo. La tecnología de súper-resolución comprende múltiples variantes y ramificaciones de distintos métodos, que son principalmente clasificados según su dominio de acción, que puede ser espacial o frecuencial y según los datos de entrada, pudiéndose tratar de una imagen o varias.

En líneas generales, la resolución de una imagen puede ser clasificada según distintos criterios [41]:

- **Resolución espacial.** Relacionada con el nivel de detalle perceptible y presente en imágenes. Existe una relación entre el tamaño del píxel iluminado por la luz de un objeto y como queda representado este en formato matricial. Se estima que cuanto menor sea el tamaño del píxel, mayor será el detalle captado, aportando una mayor resolución espacial. Sin embargo, contar con un tamaño menor de

píxeles normalmente deriva en la necesidad de una mayor cantidad de estos para obtener un buen detalle de la escena.

- **Resolución espectral.** Este tipo de resolución queda decidida por el número y el ancho de las bandas espectrales captadas por el sensor que implementa una cámara, al aumentar el número de canales percibidos y contar con bandas más estrechas, mejor será este tipo de resolución.
- **Resolución radiométrica.** Junto a la resolución espacial esta es la que se considera de gran importancia en el transcurso del TFM. La resolución radiométrica se determina según el número de niveles donde se divide la radiación recibida y almacenada por el sensor. El concepto que más se relaciona con este tipo de resolución es la sensibilidad, la capacidad de discriminar pequeñas variaciones en la radiación captada, así como su valor exacto. De la misma manera, en termografía, la resolución radiométrica puede hacer referencia al número de niveles digitales usados para expresar los datos recogidos por el sensor. En general, cuando mayor es el número de niveles mayor es el detalle con que se podrá expresar esta información. El rango dinámico de los sensores infrarrojos normalmente es codificado en 16 bits.
- **Resolución temporal.** Representa fundamentalmente el intervalo o frecuencia con la que se capturan imágenes de una determinada escena. Dependiendo de la aplicación, resoluciones temporales altas podrían ser útiles para escenas que cambian en períodos relativamente cortos, como la videovigilancia, mejorando la probabilidad de obtener imágenes de buena calidad.

En la actualidad, la mayoría de las técnicas de súper-resolución se centran en mejorar las limitaciones de la resolución espacial, ya que es un problema presente en gran parte de las aplicaciones relacionadas con cámaras de luz visible. Desde el momento de captura de una imagen y su conversión digital, existe una pérdida intrínseca de calidad producida por distorsiones ópticas, desenfoque de movimiento y ruido del sensor, entre otros. De esta forma, la imagen capturada, incluso sin manipulación, suele no ser fiel al escenario que se está observando. Entre los objetivos de la súper-resolución también se encuentra subsanar este tipo de degradaciones, a este proceso se le llama restauración

de imágenes. El modelado de este tipo de degradaciones se realiza en un modelo de observación.

2.3.1. Modelo de observación

La cámara de luz visible es la forma más común de capturar una imagen. Aunque la imagen proyectada es de dominio analógico, a menudo se convierte en una representación digital para su posterior análisis. Esto significa que una determinada escena se puede representar como una señal continua de ancho de banda arbitrario que debe ser medida por un dispositivo de muestreo discreto [42]. El modelo de observación establece una relación entre la imagen HR (*High-Resolution*) original y las imágenes LR (*Low-Resolution*) observadas. De este modo, las imágenes LR observadas son la versión deformada, borrosa, diezmada y con ruido de la imagen HR original. Para poder parametrizar el rendimiento de muchos métodos de súper-resolución es necesario recurrir a un modelo de observación, esto es degradar una imagen HR de forma artificial para crear una imagen LR. En este punto es posible aplicar el método de súper-resolución y observar el desempeño general de este. Un ejemplo de modelo de observación, donde se muestran distintas degradaciones introducidas a la imagen, se ilustra en la Figura 2.5.

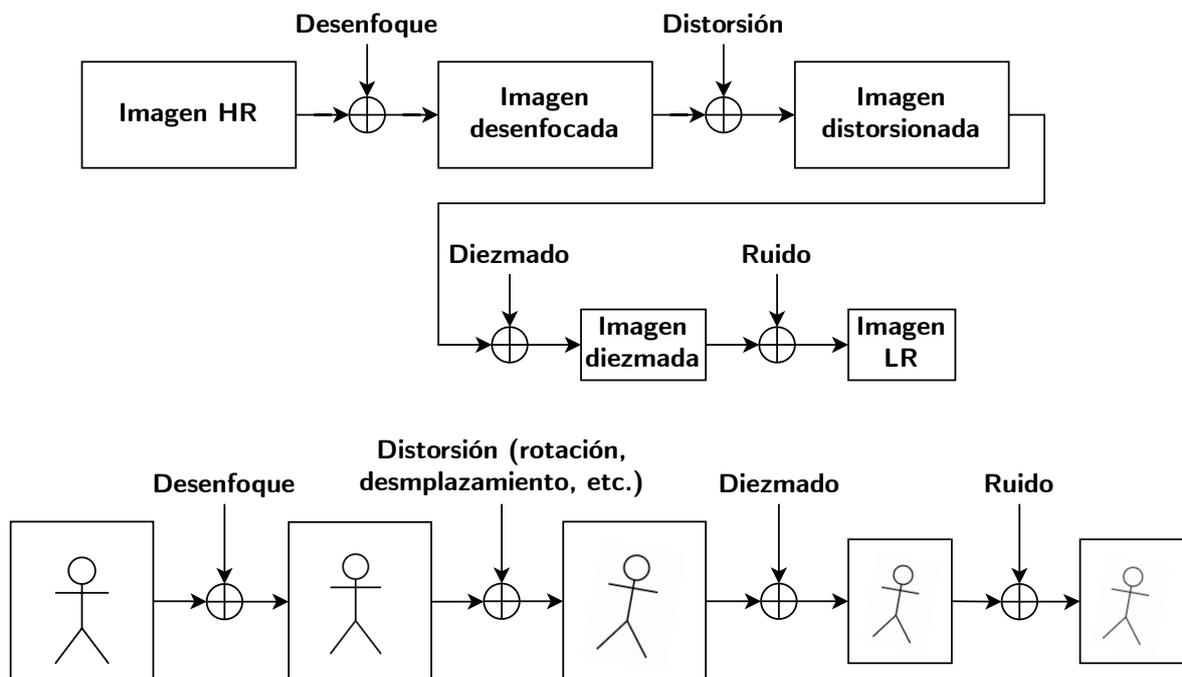


Figura 2.5: Modelo de observación genérico de una imagen

En este tipo de modelo de observación el sistema de captación recibe a su entrada una serie de señales degradantes, tras un determinado procesado digital se obtiene la imagen digitalizada en formato matricial, en este caso una imagen de baja resolución.

Algunas de estas degradaciones se corresponden con:

- Los diversos desenfoques que se suelen modelar mediante la convolución de la imagen con un filtro paso bajo.
- Transformaciones geométricas que incluyen la traslación, la rotación y el escalado.
- Alteraciones arbitrarias en el brillo o color de una imagen.
- Imperfecciones al digitalizar y submuestrear la señal por parte del sensor.

Se podría decir entonces que las técnicas de súper-resolución no sólo se encargan de mejorar la relación de píxeles por área en una determinada imagen, sino que también es muy usada en el campo de la restauración de imágenes digitales.

2.3.2. Métodos de súper-resolución

Desde un punto de vista práctico, las técnicas de súper-resolución se han visto limitadas por la potencia de cómputo de los ordenadores a lo largo del tiempo. Es decir, si bien su concepción se remonta a tres décadas anteriores, no es hasta hace relativamente poco tiempo que se han podido llevar del terreno teórico al práctico. Al menos desde un punto de vista comercial y no tanto ligado con el entorno de la investigación. El avance de la tecnología actual ha permitido que se propongan diferentes tipos de algoritmos y métodos de súper-resolución para solucionar una amplia gama de problemas. El uso de estos algoritmos o el planteamiento de su uso dependen del ámbito de aplicación, ya que muchos de estos han sido especialmente diseñados para suplir cierto tipo de degradaciones específicas. En realidad, la evolución de este tipo de técnicas ha derivado en dos dominios claros de aplicación: el dominio de la frecuencia y el dominio del espacio. El primero, como su nombre indica, intenta solucionar problemas desde el enfoque de trabajo de un dominio frecuencial. Uno de los primeros trabajos que se plantean en este dominio viene dado por Tsai y Huang [43], donde su acercamiento planteaba la relación de una imagen HR con varias imágenes LR usando una serie de ecuaciones relacionadas con las propiedades de la transformada de Fourier.

El método planteado se basaba en tres principios con base matemática:

1. El uso de la propiedad de desplazamiento de la transformada de Fourier.
2. La existencia de una relación entre los coeficientes de la transformada discreta de Fourier de las imágenes LR con la transformada continua de Fourier de la imagen HR que se desea obtener.
3. Se establece la suposición de que la imagen original, una imagen HR, tiene su ancho de banda limitado.

En este punto, el trabajo realizado por Tsai y Huang demuestra que, aplicando los tres principios anteriores, imágenes LR con solapamiento (muestreadas sin cumplir el criterio de Nyquist) pueden usarse para obtener una imagen HR sin solapamiento, un ejemplo de esta relación se muestra en la Figura 2.6. Este método ha sido mejorado con trabajos posteriores, considerando otro tipo de degradaciones, cambiando el enfoque para la obtención de coeficientes o introduciendo estimadores de movimientos. Sin embargo, la mayoría de las innovaciones en el campo de la súper-resolución actual se realizan hoy sobre el dominio espacial, más que el enfoque desde el dominio frecuencial.

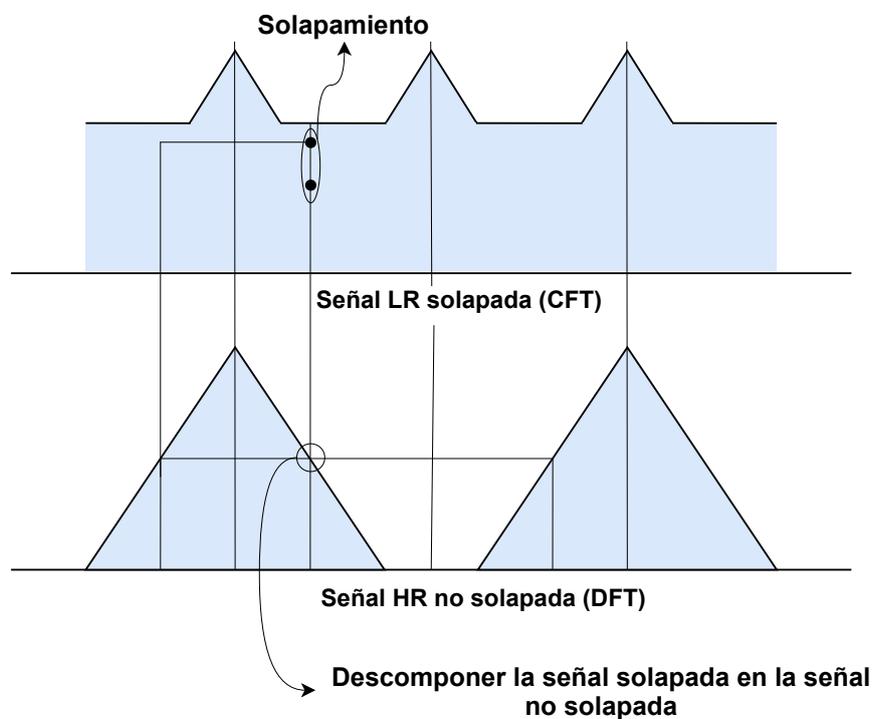


Figura 2.6: Relación de solapamiento entre la imagen LR y HR. Adaptada de [44]

2.3.3. Súper-resolución en el dominio espacial

Este tipo de súper-resolución suele trabajar con los valores numéricos de los píxeles que componen una matriz digital de datos, aprovechando distintas técnicas de procesamiento de imagen tanto a nivel de píxeles individuales como a nivel de grupo de píxeles, permitiendo así mejorar la imagen resultante. En este aspecto, las técnicas usadas en el dominio espacial son aplicadas teniendo en cuenta la cantidad de datos entrantes, pudiendo ser técnicas basadas en una sola imagen o en múltiples imágenes. Las primeras suelen establecer un mapeo directo entre una imagen LR y una imagen HR, ya sea mediante una base de datos bien clasificada, diccionarios clave-valor o registros temporales similares. Es decir, suele ser condición impuesta que, al usar estas técnicas, ambas imágenes se hayan capturado en un mismo lapso. Por otro lado, las técnicas basadas en múltiples imágenes suelen sentar las bases del proceso de reconstrucción, entendiendo este como la recuperación de una imagen HR a partir de varias imágenes LR, teniendo en cuenta que estas presentan algunos desplazamientos geométricos y/o fotométricos relativos respecto a la imagen HR objetivo.

En la actualidad, los métodos de súper-resolución de una sola imagen se suelen implementar a través de una aproximación basada en aprendizaje. Estos métodos tratan de percibir la información que falta en las imágenes SR utilizando la relación entre las imágenes LR y HR de una base de datos de entrenamiento [17]. En este trabajo, dentro de los métodos basados en una sola imagen, se estudiarán aquellos basados en inteligencia artificial. La inteligencia artificial proporciona una flexibilidad y posibilidades superiores a las técnicas tradicionales, a cambio de una potencia de cálculo mayor. En este sentido, también se estudiarán algunos métodos usados de forma clásica que no necesitan de un entrenamiento previo ni están basados en aprendizaje a partir de inteligencia artificial. Este hecho proporcionará una perspectiva interesante para poder comparar ambos tipos de métodos.

En la siguiente sección se estudiarán los métodos de reconstrucción más utilizados en el dominio espacial. Sin embargo, los métodos basados en redes neuronales que se han usado en este trabajo se enunciarán en el capítulo del estado de la técnica.

2.4. Métodos de reconstrucción

Los métodos de reconstrucción son, en su mayoría, algoritmos que basan su funcionamiento en conseguir una imagen SR a partir de un conjunto de imágenes LR. Es decir, son algoritmos basados en múltiples imágenes. Los métodos de reconstrucción intentan abordar los artefactos de solapamiento que están presentes en las imágenes LR observadas, debido al proceso de submuestreo que se realiza en toda cámara digital. Dentro del dominio espacial y los procesos de reconstrucción a partir de múltiples imágenes existe una gran cantidad de métodos, en este trabajo se estudiarán las bases de los más importantes y presentes en el estado de la técnica.

2.4.1. Método basado en algoritmos de interpolación

La estrategia que siguen los métodos basados en algoritmos por interpolación consiste en el uso de un conjunto de imágenes LR para poder generar una imagen HR. Este conjunto de imágenes contiene un solapamiento con diferente información frecuencial. De esta forma, aplicando ciertas condiciones iniciales, es teóricamente posible suplir parte de este solapamiento para reconstruir las frecuencias altas perdidas. Normalmente, para poder generar un conjunto de imágenes LR y poder probar estos algoritmos se suele considerar un movimiento global de traslación y rotación, creado artificialmente, con la intención de emular el desplazamiento de una cámara [45].

Los métodos de reconstrucción por interpolación pueden dividirse en tres pasos distintos:

- **Registro.** En las referencias consultadas, se llama registro a toda técnica que puede llegar a usarse para estimar las traslaciones relativas entre todo el conjunto de imágenes de entrada. En el campo de la súper-resolución de imágenes, el registro es necesario para determinar estos desplazamientos relativos entre imágenes con una precisión alta, existiendo de la misma manera una gran cantidad de algoritmos distintos para aplicar en este proceso y ofreciendo ciertas ventajas y desventajas según el contexto de uso.
- **Reconstrucción.** La reconstrucción es el proceso siguiente a la etapa de registro, ya que todos los píxeles del conjunto registrado pueden combinarse para formar

una imagen final. En este proceso de fusión suelen utilizarse diferentes filtros, dependiendo de las características intrínsecas de las imágenes estudiadas [17].

- **Desenfoque.** Finalmente, es posible aplicar un proceso de desenfoque que restablezca las frecuencias altas que se perdieron previamente en el proceso de obtención de imágenes de baja resolución.

Al finalizar el proceso de registro, se estima que todos los píxeles del conjunto pueden llegar a combinarse hasta formar una imagen final, como se observa en la Figura 2.7. No obstante, la imagen resultante, debido al movimiento global de traslación, pierde gran parte de su uniformidad. Con la finalidad de reconstruir los datos de la imagen en todos los puntos de una cuadrícula dada es necesario interpolar y volver a muestrear los datos.

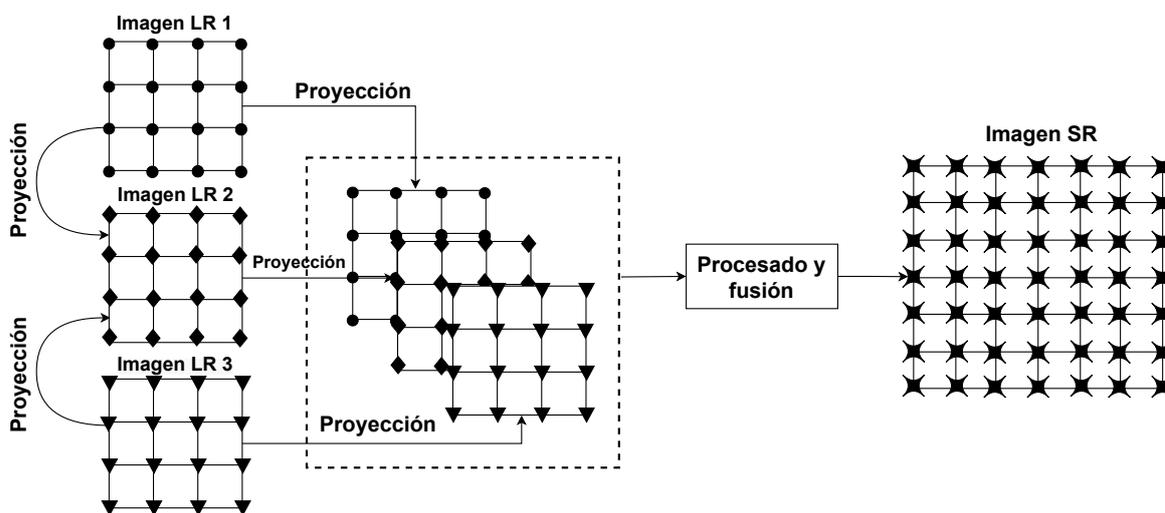


Figura 2.7: Formación de una imagen SR a partir de un conjunto de imágenes LR. Adaptada de [42]

Desde un punto de vista del procesamiento digital, las imágenes que capta una cámara son muestras de una escena continua que esta ha observado. Estas muestras se posicionan en un conjunto discreto de ubicaciones que puede interpretarse como una cuadrícula rectangular, pudiendo ser esta regular o irregular. Es decir, matrices de datos digitales. De esta forma, en una gran parte de las situaciones se desea obtener el valor de un punto concreto en una ubicación determinada, dada por estas cuadrículas. Esto necesita de una reconstrucción de la función continua o, en otras palabras, una representación de la escena capturada $g(x)$, que en última instancia es la imagen HR.

El proceso de interpolación se define matemáticamente de la siguiente forma [45]:

1. Dada una función continua que representa la escena capturada $g(x)$ es posible definir una imagen de entrada muestreada y discretizada, a la que se llamará $f_1(p)$, perteneciendo $f_1(p)$ a toda una cuadrícula de localizaciones P estudiadas.
2. De esta forma, se considera que $f_1(p)$, para todo P , tomará una serie de valores conocidos como $g(p)$. Mientras que para las regiones que no son P , serán desconocidos a priori. Esto queda descrito matemáticamente en la ecuación 2.12.

$$f_1(p) = \begin{cases} g(p) & \text{en la cuadrícula} \\ \text{indefinido} & \text{fuera de la cuadrícula} \end{cases} \quad (2.12)$$

3. El objetivo que tienen los métodos basados en interpolación es estimar una imagen continua $\tilde{g}(x)$ a partir de $f_1(p)$.
4. La operación de interpolación puede llegar a considerarse como una función de filtrado e implementarse como una convolución espacial. De esta manera, la función continua reconstruida $\tilde{g}(x)$ es una combinación lineal de valores de entrada, como se muestra en la ecuación 2.13.

$$\tilde{g}(x) = \sum_p f_1(p) H_p(x - p), p \in P \quad (2.13)$$

Donde $H_p(\cdot)$ puede ser cualquier función de interpolación asociada al punto de entrada p . La elección del método de interpolación suele estar ligado a la parte práctica de la experimentación, ya que se puede demostrar que un método concreto funciona mejor que otro para un conjunto de datos específico.

De forma resumida, el enfoque de súper-resolución basado en la interpolación trata de construir una imagen SR proyectando las imágenes LR adquiridas, luego fusiona toda la información disponible de cada imagen, debido a que cada imagen LR proporciona una cantidad de información adicional sobre la escena y finalmente conforma la nueva imagen. En algunas referencias consultadas, a los métodos por interpolación se les conoce también como métodos directos [17]. Estos métodos directos incluyen en su repertorio unos algoritmos conocidos como algoritmos no paramétricos. La idea detrás

de estos algoritmos es combinar los pasos de estimación de movimiento y traslación con la de fusión de imágenes. El uso potencial que se les ha dado a estos algoritmos suele encontrarse en secuencias de vídeo, ya que reaccionan de forma óptima a características relacionadas con las componentes faciales o al cambio de estas expresiones faciales.

2.4.1.1. Métodos de interpolación clásicos

La interpolación ocurre en gran parte del procesado de una imagen digital. Este proceso interviene en las operaciones de cambio de tamaño o reajuste de una imagen. La primera consiste en el cambio de un mallado concreto a otro, siendo necesario si se desea aumentar o disminuir el número total de píxeles de una matriz. El reajuste se utiliza de distinta forma dependiendo del escenario estudiado (corregir distorsión de las lentes y rotar imágenes, entre otros.). El proceso de interpolación es un tipo de estimación, esto implica cierta pérdida de calidad cada vez que se efectúa sobre una imagen. De esta manera, la interpolación es el proceso que estima valores de puntos desconocidos usando datos conocidos, siendo el ejemplo más sencillo la interpolación lineal. De forma general, cuantos más datos sean conocidos de antemano, más preciso puede llegar a ser el algoritmo, como se observa en la Figura 2.8.

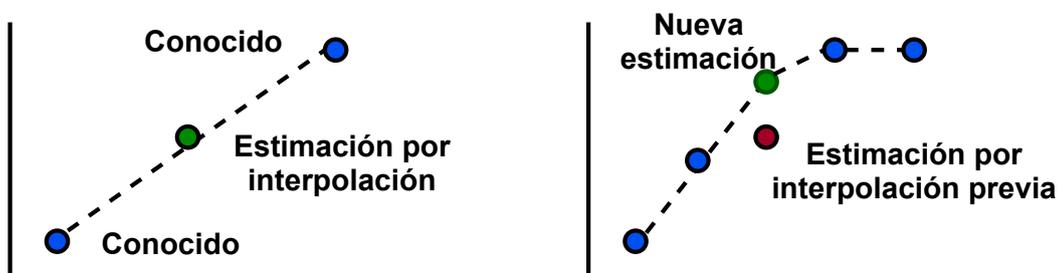


Figura 2.8: Estimaciones de una interpolación lineal según el número de muestras

Cuando la interpolación lineal es aplicada a dos direcciones del espacio se le llama interpolación bilineal, el producto de dos interpolaciones lineales. En el campo del procesado de imagen este es uno de los métodos más sencillos para volver a muestrear una matriz de datos, intentando de esta forma aproximar un determinado píxel basándose en los valores de los píxeles circundantes. Por este motivo, se debe tener en cuenta que la interpolación nunca puede añadir detalles a la imagen que no estén ya presentes. Los métodos de interpolación tradicionales usados en el campo de la súper-resolución son la interpolación del vecino más cercano, interpolación bilineal y bicúbica [46]. De

forma general, cuantos más píxeles circundantes tomen en consideración estos métodos, mayor será su desempeño, pero más capacidad de cómputo necesitarán.

Una explicación breve de estos métodos se ofrece en los siguientes puntos:

- **Vecino más cercano.** El método más sencillo de interpolación para imágenes y el que menos consumo de tiempo necesita, ya que sólo toma en consideración un píxel, el más cercano al punto de interpolación. Este algoritmo por aproximación simplemente aumenta el tamaño del píxel.
- **Bilineal.** La interpolación bilineal estima un píxel desconocido a través de sus vecinos, tomando el conjunto de cuatro píxeles adyacentes y realizando la media ponderada de estos cuatro para obtener el valor estimado del desconocido.
- **Bicúbica.** Similar a la interpolación bilineal, pero tomando y promediando los dieciséis píxeles adyacentes para estimar el píxel desconocido.

Existen otros métodos de interpolación de mayor orden. Es decir, toman en consideración muchos más píxeles conocidos. No obstante, estos no son tan usados en el campo de la súper-resolución debido a que requieren de una mayor capacidad de cómputo.

2.4.2. Métodos de regularización mediante aproximaciones Bayesianas

El campo de la súper-resolución ha sido estudiado y mejorado a lo largo del tiempo, una de las mejoras más significativas son los algoritmos de regularización. Estos algoritmos nacen en un momento importante, ya que el estado de la técnica llevaba a una conclusión general, el cálculo de súper-resolución para imágenes digitales se consideraba un problema inverso y, además, mal planteado.

2.4.2.1. Problemas inversos

Los problemas inversos son aquellos que tratan de encontrar la causa de algunos resultados determinados. En el contexto de la súper-resolución, se podría extrapolar que los resultados son imágenes LR y las causas son imágenes HR. Se tiene en posesión imágenes LR y se quiere encontrar la imagen HR que las ha producido, de esta forma se podría reconstruir la imagen SR [47].

- La estimación de una imagen SR es un problema inverso y se sabe que está mal planteado. Esto es así porque pequeños errores en las imágenes observadas pueden causar grandes cambios en la reconstrucción [47], [48].

En este contexto, según la clasificación formal realizada por Jacques Hadamard [47], [49], un problema bien planteado es un problema cuya solución tiene las siguientes propiedades:

- **Existencia de la solución.** Es decir, para todos los datos, la solución del problema o modelo debe existir.
- **Unicidad de la solución.** Es decir, para todos los datos, si la solución existe, debe ser única.
- **Estabilidad del proceso de solución.** La solución del problema o del modelo debe depender continuamente de los datos.

Por otro lado, se dice que un problema está mal planteado, desde el punto de vista de esta formulación, cuando no se satisfacen las propiedades mencionadas anteriormente.

En este sentido, los problemas clásicos de la súper-resolución suelen estar mal planteados, puesto que un pequeño cambio en los datos observados puede provocar un gran cambio en la imagen estimada. La parte importante del proceso de súper-resolución es, por lo tanto, modificar el problema original de forma que la solución sea una aproximación significativa y cercana a la escena real, pero, al mismo tiempo, sea menos sensible a los errores de las imágenes observadas. El procedimiento para lograr este objetivo y estabilizar el proceso de reconstrucción se conoce como regularización [47], [48]. Desde esta perspectiva y con la seguridad de que el cálculo de súper-resolución era un problema inverso mal planteado, se desarrollaron una gran cantidad de algoritmos que implementan términos de regularización para abordar esta cuestión [42].

- Se entiende por regularización a la incorporación del conocimiento previo de la imagen HR desconocida. Desde el punto de vista bayesiano, la información que puede extraerse de las observaciones LR sobre la señal desconocida está contenida en la distribución de probabilidad de la incógnita.

- Los dos enfoques más populares de la súper-resolución basada en la teoría bayesiana son el enfoque ML (*Maximum Likelihood*) y la estimación MAP (*Maximum A Posteriori*).

2.4.2.2. Aproximaciones bayesianas

Las aproximaciones bayesianas se basan fundamentalmente en modelos probabilísticos, donde las entradas a un sistema no siempre producen la misma salida. Se considera entonces que la imagen HR deseada y las diferencias de traslación y rotación de las imágenes de menor resolución son variables estocásticas, permitiendo modelar la información relativa a la solución dentro de un marco bayesiano, donde las evidencias u observaciones se emplean para actualizar el valor de los parámetros. La premisa, desde el punto de vista bayesiano, es que la información puede extraerse de las observaciones LR y que la señal desconocida HR está contenida en la distribución de probabilidad de la incógnita. Entonces, la imagen HR desconocida puede estimarse explotando la información proporcionada tanto por las imágenes LR observadas como por el conocimiento previo de la imagen HR desconocida [42].

El teorema bayesiano describe la relación entre dos probabilidades condicionales. El teorema expresa la probabilidad posterior de una hipótesis H (la probabilidad de H después de observar la evidencia E) en términos de la probabilidad a priori de E y la probabilidad condicional de E dada H .

- En el campo de la súper-resolución, la imagen HR se considera la hipótesis y las imágenes LR son la evidencia. La reconstrucción de la imagen SR que emplea el enfoque bayesiano tiene que encontrar la estimación de la imagen HR dadas las observaciones LR.

El teorema de bayes se describe matemáticamente en la ecuación 2.14.

$$P(H | E) = \frac{P(E | H)P(H)}{P(E)} \quad (2.14)$$

En este caso:

- $P(H)$ se denomina probabilidad a priori de H , ya que no depende de E .

- $P(H | E)$ es la probabilidad condicional de H en términos de E .
- $P(E | H)$ es la probabilidad condicional de E en términos de H .
- $P(E)$ es la probabilidad a priori de E .

Extrapolado a la súper-resolución se puede obtener la probabilidad de f (la imagen HR), en función de g_k (la imagen LR), según se confirma en la ecuación 2.15.

$$P(f | g_k) = \frac{P(g_k | f)P(f)}{P(g_k)} \quad (2.15)$$

2.4.3. Método de retroproyección iterativa

El método de retroproyección iterativa o IBP (*Iterative Back Projection*) se considera uno de los primeros métodos desarrollados para el campo de la súper-resolución basada en el dominio espacial [17]. Este método recibe el nombre de iterativo ya que siempre empieza con una conjetura inicial para la imagen HR, refinando esta conjetura en un proceso iterativo [17], [50].

2.4.3.1. IBP en la súper-resolución

En un inicio se puede decir que f es la imagen HR que se desea conseguir, de esta manera la conjetura inicial quedaría definida como $f^{(0)}$. El siguiente paso es simular o generar de forma sintética una serie de imágenes LR que se denotarán como $g_k^{(0)}$. Por último, el error entre las imágenes LR simuladas y las observadas se puede calcular mediante distintas variantes de la ecuación 2.16.

$$Error_{IBP} = \frac{1}{K} \sum_{k=1}^K g_k - g_k^{(n)} \quad (2.16)$$

Donde n es el número de iteraciones.

El error que se obtiene de la ecuación anterior es proyectado hacia atrás, a las coordenadas de la imagen HR para mejorar la suposición inicial. Este proceso se repite durante un número determinado de iteraciones o hasta que no se consiga ninguna mejora. Si el valor de un píxel no cambia durante un número específico de iteraciones, su valor se considerará resuelto. Este proceso es iterativo y reduce el error residual de

la simulación al mínimo [51]. Una descripción matemática del algoritmo iterativo de retroproyección se ilustra en la ecuación 2.17.

$$f^{(n+1)} = f^{(n)} - \lambda \Delta_L$$

$$\Delta_L = \frac{1}{K} \sum_{k=1}^K H_k^{BP} \left(g_k - g_k^{(n)} \right) \quad (2.17)$$

Donde $f^{(n+1)}$ es la imagen SR obtenida en la iteración $n + 1$ y $f^{(n)}$ es la imagen SR obtenida en la iteración n . El factor H_k^{BP} es la operación de retroproyección y λ es el paso de gradiente [52].

El principal problema de algunos métodos IBP es que la respuesta de la iteración puede converger a una de las posibles soluciones o puede oscilar entre ellas. Sin embargo, esto puede solucionarse incorporando un conocimiento a priori sobre la solución. Entonces uno se encontraría ante un algoritmo iterativo y, además, regularizador [17].

2.4.4. Método de proyección sobre conjuntos convexos

El algoritmo de proyecciones sobre conjuntos convexos o POCS (*Projection Onto Convex Sets*) permite encontrar un punto en la intersección de dos conjuntos convexos cerrados. Una forma sencilla de ver el concepto de POCS gráficamente se muestra en la Figura 2.9. En esta figura se parte de un punto Ω_0 y, a partir de distintas proyecciones sobre los conjuntos convexos, se llega al punto Ω° que minimiza el error. Es decir, se llega a la solución requerida. De esta forma, el enfoque POCS describe un enfoque iterativo y alternativo al IBP, con la intención principal de incorporar el conocimiento previo sobre la solución en el proceso de reconstrucción, añadiendo un término de regularización extra. Con las estimaciones de los parámetros de registro, este algoritmo es capaz de resolver simultáneamente el problema de restauración e interpolación para estimar una imagen SR.

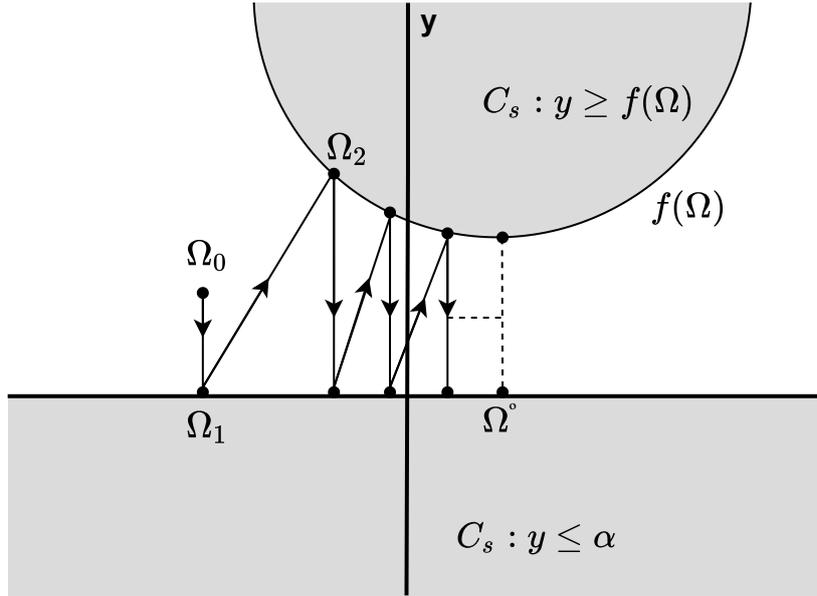


Figura 2.9: Representación gráfica del algoritmo POCS. Adaptada de [53]

El principal resultado de aplicar POCS es la capacidad de definir conjuntos convexos que representan restricciones estrictas sobre la imagen requerida.

2.4.4.1. POCS en la súper-resolución

En el problema de la súper-resolución se considera un número finito de conjuntos convexos que contienen la información de la imagen deseada, esta imagen es entonces un punto dentro de los conjuntos convexos [54]. Cada conjunto contiene información individual del problema, que corresponde con la información obtenida de una imagen LR. La solución al problema global se obtiene de la intersección de dichos conjuntos [41]. Es decir, en los métodos basados en POCS cada imagen LR impone un conocimiento a priori sobre la solución final [17]. Se asume que este conocimiento a priori es un conjunto convexo cerrado Ω_k , que se define matemáticamente según la ecuación 2.18.

$$\Omega_k = \{f \mid \delta_l \leq |dh_k w_k f - g_k| \leq \delta_u\} \quad (2.18)$$

Donde g_k es la imagen LR, d , h_k y w_k son factores que degradan la imagen, f es la solución y δ_l , δ_u son las incertidumbres de límite inferior y superior del modelo de conjuntos convexos cerrados.

Teniendo este grupo de K conjuntos convexos, se puede utilizar la siguiente iteración del procedimiento para estimar la imagen SR desconocida a partir de la ecuación 2.19.

$$f^{(n+1)} = P_m P_{m-1} \dots P_2 P_1 f^{(n)} \quad (2.19)$$

Donde $f^{(n+1)}$ es la estimación de la imagen SR y P el operador de proyección.

El método de POCS requiere la definición de conjuntos de restricciones convexos cerrados dentro de un espacio vectorial bien definido, que contienen la imagen SR deseada. Una estimación de la imagen SR se define entonces como un punto en la intersección de estos conjuntos de restricciones y se determina proyectando sucesivamente una estimación inicial arbitraria sobre los conjuntos de restricciones. A cada conjunto de restricciones se le asocia un operador de proyección P , que asigna un punto arbitrario dentro del espacio al punto más cercano dentro del conjunto [55]. Un ejemplo que ilustra el concepto de proyección se muestra en la Figura 2.10.

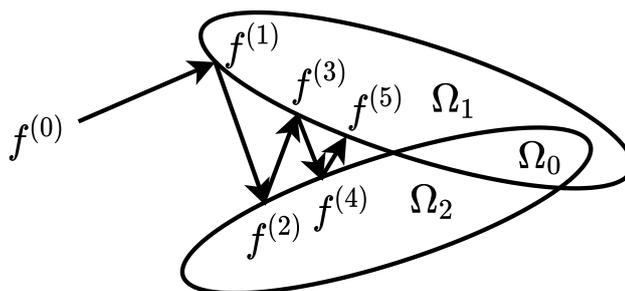


Figura 2.10: Proyecciones sobre dos conjuntos convexos cerrados. Adaptada de [56]

La intuición detrás del algoritmo POCS aplicado a súper-resolución es que las restricciones lineales y no lineales sobre la imagen final, además de los conjuntos de datos disponibles, pueden representarse a menudo como conjuntos convexos en un espacio vectorial, donde cada elemento de este espacio es una imagen o una característica concreta de esta que se usará en la siguiente proyección de la estimación [57].

De forma similar a lo que ocurre con el método IBP, el proceso de convergencia del algoritmo POCS es importante en el proceso de súper-resolución, ya que el tiempo de procesado y la existencia de una solución afecta al desempeño de los métodos iterativos. Un ejemplo gráfico del proceso de convergencia de un algoritmo POCS es mostrado en la Figura 2.11. En esta se observa cómo, para dos conjuntos convexos cerrados Ω_1 y Ω_2 ,

existe una intersección Ω_0 , que representa el punto de solución al que se quiere llegar. En este caso, $f^{(0)}$ es el punto de comienzo del algoritmo o la conjetura inicial.

Más concretamente, la proyección mostrada en la Figura 2.11 es la proyección relajada T_i , definida matemáticamente en la ecuación 2.20.

$$T_i = I + \lambda_i(P_i - I) \tag{2.20}$$

Donde I es el operador identidad y λ_i es un parámetro de relajación, que afecta a la velocidad de convergencia. Este parámetro de relajación que puede variar entre 0 y 2.

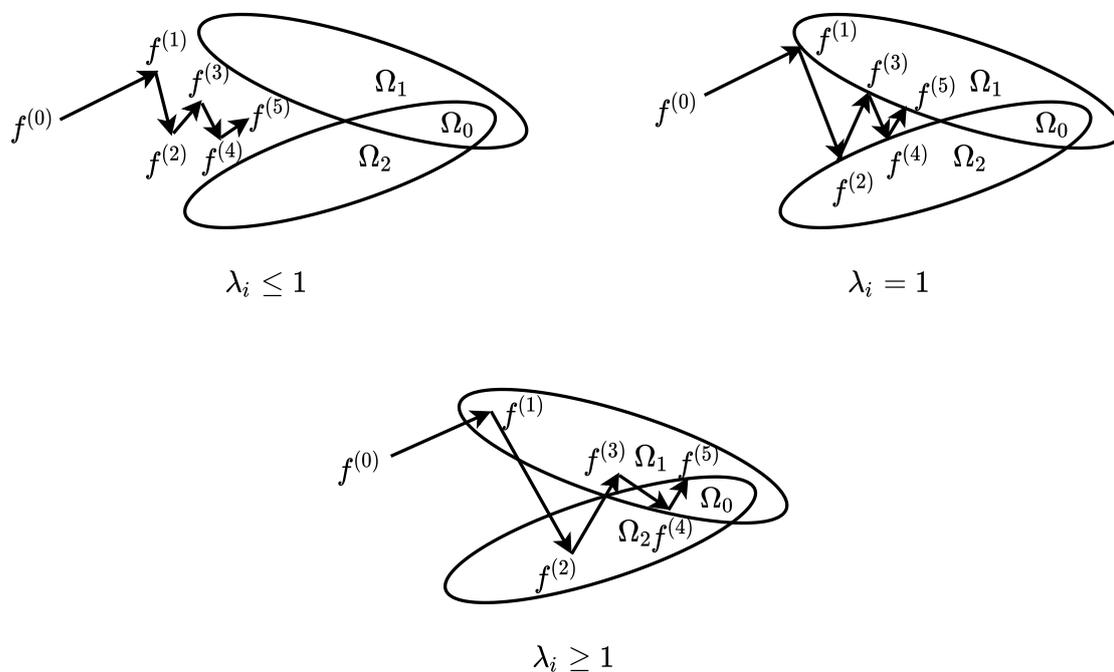


Figura 2.11: Ejemplo de convergencia de un algoritmo POCS. Adaptada de [56]

La teoría POCS ofrece un potente aparato matemático para resolver muchos problemas de reconstrucción que implican datos incompletos e inconsistentes. La principal ventaja de POCS es que presenta una forma natural de incorporar el conocimiento a priori en el proceso de reconstrucción [57]. Por lo tanto, el problema de reconstrucción de la imagen puede reformularse como el problema de encontrar un punto en un espacio vectorial que pertenezca a la intersección de los conjuntos convexos definidos por la información a priori.

2.4.5. Método de Papoulis-Gerchberg

El método de Papoulis-Gerchberg se basa fundamentalmente en los trabajos independientes de Papoulis [58] y Gerchberg [59]. A continuación, se explican las bases detrás de su trabajo.

2.4.5.1. Bases del método

Por un lado, Gerchberg propone un método para realizar la reconstrucción de una señal a través del límite de difracción de esta y una parte del espectro [60]. Por otro lado, el trabajo de Papoulis fue la extrapolación de una señal limitada en banda a partir de sólo una parte de la señal original, más concretamente, a través de la determinación de la transformada de una señal $f(t)$ mostrada en la ecuación 2.21.

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t} dt \quad (2.21)$$

Un conjunto de señales genéricas $f(t)$ y $F(\omega)$ se representan en la Figura 2.12 con la intención de explicar el concepto del método.

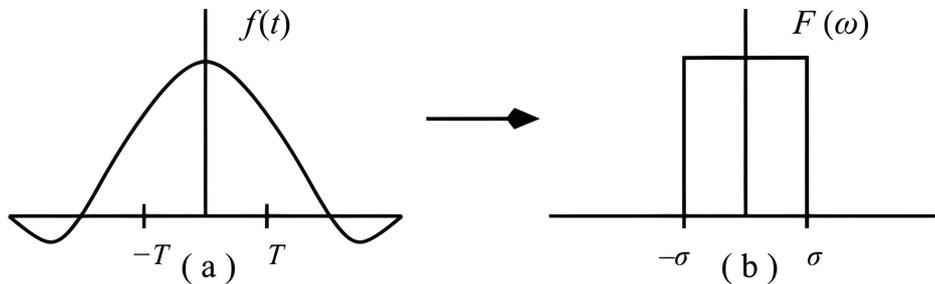


Figura 2.12: Método de Papoulis-Gerchberg (I). Extraída de [60]

En este caso, la señal $f(t)$ es equivalente a la señal que se quiere reconstruir en un ejemplo real. A partir de este punto, dado un segmento finito $p_T(t)$, la señal $f(t)$ queda limitada a un espacio concreto, conformándose la señal $g(t)$, que se describe en la ecuación 2.22.

$$g(t) = f(t)p_T(t), \text{ donde } p_T(t) = \begin{cases} 1, & |t| \leq T \\ 0, & |t| > T \end{cases} \quad (2.22)$$

Esta $g(t)$ es una versión truncada de $f(t)$. Es decir, se encuentra limitada en tiempo. A continuación, comienza el proceso iterativo de extrapolación de $g(t)$, alternando continuamente entre el dominio temporal y frecuencial. Tanto $g(t)$ como su transformada $G(w)$ se presentan en la Figura 2.13.

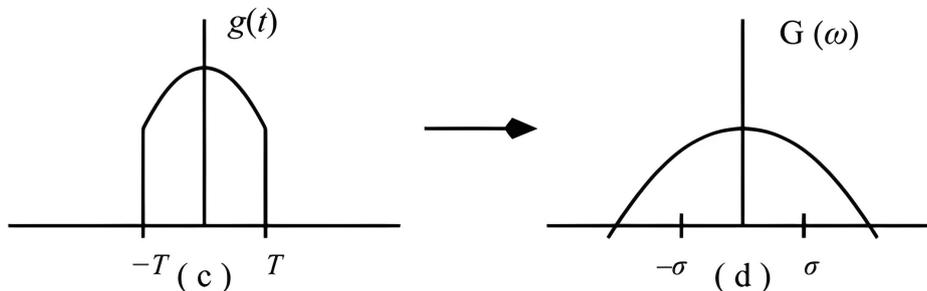


Figura 2.13: Método de Papoulis-Gerchberg (II). Extraída de [60]

En este punto, la estimación inicial del método de Papoulis-Gerchberg es $g(t) = g_0(t)$. El siguiente paso es el filtrado de $g(t)$ a través de un filtro paso bajo con una frecuencia de corte de σ , asumiendo que σ es el ancho de banda de la señal $f(t)$. Esta señal filtrada, denominada $F_1(w)$ en el dominio frecuencial, se consigue a partir de $G(w) = G_0(w)$. Es decir, a partir de la estimación inicial del método. Las señales $F_1(w)$ y su equivalente en tiempo $f_1(t)$, así como la etapa anterior, se representan en la Figura 2.14.

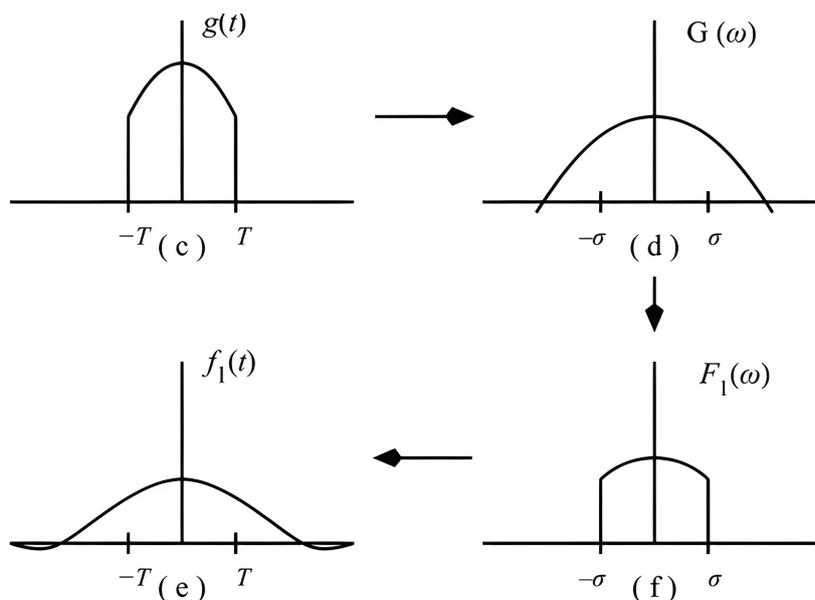


Figura 2.14: Método de Papoulis-Gerchberg (III). Extraída de [60]

La representación formal de este paso del método para su n -ésima iteración se puede

expresar según la ecuación 2.23.

$$F_n(\omega) = G_{n-1}(\omega)p_\sigma(\omega), \quad p_\sigma(\omega) = \begin{cases} 1, & |\omega| \leq \sigma \\ 0, & |\omega| > \sigma \end{cases} \quad (2.23)$$

El resultado de volver al dominio temporal, en este caso de $F_1(\omega)$ a $f_1(t)$, es una reducción del error $|f(t) - f_1(t)|^2$ fuera del segmento conocido de la señal. Sin embargo, la señal $f_1(t)$ no coincide con la señal observada $g(t)$ en la región $[-T, T]$. Esta parte de la señal necesita restablecerse al segmento original conocido, formando la señal $g_1(t)$ para la siguiente iteración. El cambio resultante en el dominio espectral debido a la introducción de componentes de mayor frecuencia puede verse en la Figura 2.15.

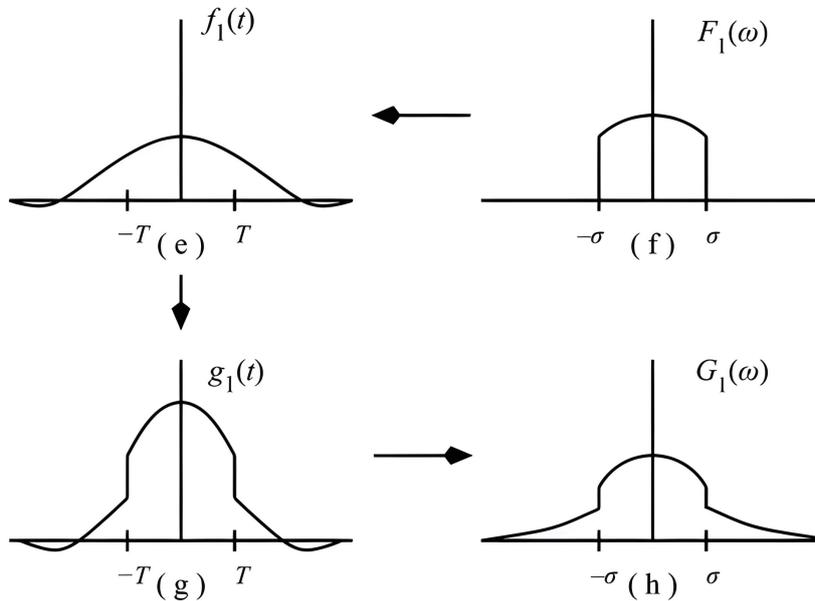


Figura 2.15: Método de Papoulis-Gerchberg (IV). Extraída de [60]

Generalizando esto para la n -ésima iteración se obtiene la ecuación 2.21.

$$g_n(t) = f_n(t) + [f(t) - f_n(t)]p_T(t) = \begin{cases} g(t), & |t| \leq T \\ f_n(t), & |t| > T \end{cases} \quad (2.24)$$

Este proceso se repite otra vez con la nueva $g_1(t)$ formada. En cada iteración se reduce el error cuadrático medio de la señal extrapolada. Así, con las sucesivas iteraciones, la señal extrapolada generada se aproxima a la señal deseada $f(t)$. La convergencia del método está garantizada y se muestra en [58]. Sin embargo, el proceso requiere un número infinito de iteraciones. Si se detiene el proceso después de r iteraciones, la

señal reconstruida viene dada por $f_r(t)$ en lugar de $f(t)$. Por otro lado, en situaciones prácticas, los datos medidos $g(t) = g_0(t)$ contendrán errores. No obstante, la propagación de estos errores puede controlarse mediante la finalización temprana del proceso iterativo [59]. De la misma forma, el proceso también supone que la señal $f(t)$ es de banda limitada, pero se ha comprobado que el método funciona razonablemente bien para señales con una energía suficientemente baja en sus componentes de frecuencia más alta [60].

2.4.5.2. El método de Papoulis-Gerchberg en la súper-resolución

El método de Papoulis-Gerchberg puede ser extrapolado de las señales unidimensionales a matrices de varias dimensiones. En este trabajo se comentarán dos acercamientos existentes tanto para múltiples imágenes LR como para una sola imagen LR.

- **Primer acercamiento.** El método de Papoulis-Gerchberg parte de una premisa similar a los métodos anteriormente estudiados. Una primera estimación de la imagen HR, en comparación con su equivalente LR, puede considerarse como una matriz de mayor dimensión en la que los valores de algunos píxeles son conocidos y otros desconocidos.
 1. Un primer punto de partida para aplicar este método a imágenes bidimensionales es suponer que los valores de los píxeles desconocidos se estiman inicialmente a cero.
 2. En el siguiente paso, se aplica a la imagen un filtro paso bajo con una frecuencia de corte normalizada σ , estableciendo esta como la máxima frecuencia permitida.
 3. De este modo, la imagen se difumina y los píxeles desconocidos obtienen algunos valores. Sin embargo, los valores de los píxeles conocidos también han sufrido cambios debido al filtrado.
 4. A continuación, se restauran los valores originales de los píxeles conocidos, creando componentes de alta frecuencia.
 5. Todo el proceso se repite hasta alcanzar un punto determinado, dependiendo del experimento que se esté llevando a cabo.

- **Segundo acercamiento.** Otra manera de entender el método de Papoulis-Gerchberg en el campo de la súper-resolución es pensar que este se trata de un caso especial del POCS, suponiendo entonces que la imagen pertenece a dos conjuntos lineales convexos. Mediante proyecciones repetidas, el método converge iterativamente a la imagen deseada en la intersección de los dos conjuntos [61]. Esta premisa, tal como se puede observar, parte de la definición formal de POCS aplicado a la súper-resolución. No obstante, la segunda etapa del método se desvincula de la premisa original del método POCS. En esta etapa, los valores desconocidos de los píxeles se estiman a cero durante la inicialización, realizando dos proyecciones distintas en cada iteración del método.

1. En primer lugar, la imagen se proyecta sobre el conjunto de señales de banda limitada, variantes de $g(t)$, estableciendo a cero aquellas frecuencias que son mayores a la frecuencia máxima permitida $f_{max} = \sigma$. Es decir, la que termina definiendo al filtro paso bajo aplicado.
2. A continuación, la imagen se proyecta sobre el conjunto de muestras desconocidas de la imagen SR, asignando de nuevo los valores correctos a las muestras conocidas de la imagen LR [61].

2.4.6. Método de súper-resolución robusta

En este trabajo se estudiará el concepto de súper-resolución robusta, más concretamente el método de súper-resolución propuesto por Assaf Zomet [62]. Los métodos de súper-resolución robusta parten de una premisa concreta, la estimación de una imagen HR desconocida no se basa exclusivamente en las imágenes LR, también resulta necesario tener en consideración otras muchas suposiciones, como los modelos de ruido, desenfoque o movimiento [63]. Sin embargo, estos modelos no siempre se adecuan a la realidad, siendo alguno de estos más complejos de parametrizar en la práctica que en el campo teórico. En este punto, cuando los supuestos fundamentales de los modelos de datos y ruido no describen fielmente los valores medidos, el rendimiento de un estimador se termina degradando. Además, la existencia de valores atípicos, definidos como aquellos datos cuyas características son diferentes al modelo asumido, producen estimaciones erróneas. Un método especializado para una clase limitada de datos y mo-

delos de ruido puede no ser el enfoque global más eficaz. La robustez de un método de súper-resolución se puede definir en función de que tan sensibles son sus estimaciones respecto a los valores atípicos y a los errores debidos por modelar mal el problema [63].

El método de súper-resolución descrito por Assaf Zomet se considera robusto frente a los valores atípicos causados por imprecisiones del modelo y objetos en movimiento, entre otros factores. Como base del método se propone el IBP, mejorándolo en el proceso con dos características nuevas, la introducción de la mediana como nuevo método de calcular el gradiente y un proceso para la detección de sesgos. Además, se propone que el cálculo iterativo de una imagen f se convierta en el cálculo de una imagen \vec{X} , que no es más que f reordenada en un vector, consiguiendo que las operaciones básicas del modelo de observación usado como la convolución, el muestreo y la deformación, sean lineales respecto las intensidades de las imágenes [62]. El cálculo de \vec{X} se puede representar según la ecuación 2.25.

$$\vec{X}^{(n+1)} = \vec{X}^{(n)} + \lambda \nabla L(\vec{X}) \quad (2.25)$$

Donde $\vec{X}^{(n+1)}$ y $\vec{X}^{(n)}$ son vectores y λ es un factor de escala que define el tamaño del paso en la dirección del gradiente.

2.4.6.1. Uso de la mediana de píxeles

Un enfoque directo para hacer robusto el método de súper-resolución propuesto es minimizar el error bajo una norma diferente que pueda manejar los valores atípicos. Normalmente los errores grandes en el proceso de retroproyección son debidos a estos valores, aunque también se relacionan con el solapamiento en las regiones que contienen frecuencias altas. En este acercamiento, para introducir robustez en el proceso, se propone una mediana escalada por píxeles, como se observa en la ecuación 2.26.

$$\nabla L(\vec{X})(x, y) \approx n \cdot \text{median} \left\{ \vec{B}_k(x, y) \right\}_{k=1}^n \quad (2.26)$$

Donde \vec{B}_k es una imagen diferencia que ha sido retroproyectada. En otras palabras, es el valor residual del algoritmo IBP.

Para una distribución simétrica la mediana puede aproximarse a la media con bastante precisión, dado un conjunto suficiente de muestras. El método asegura que, para ciertos escenarios donde se produzcan valores atípicos durante el proceso de retroproyección, la mediana ofrece mayor robustez que la media.

2.4.6.2. El proceso para la detección de sesgos

El objetivo del procedimiento para la detección de sesgos planteado en este método es detectar la mayoría de las mediciones atípicas que puedan sesgar el estimador de la mediana. Es decir, es un procesado extra para ayudar a la operación anterior. Los autores del método especifican que, el resultado del proceso para la detección de sesgos en una imagen LR, es en definitiva una máscara binaria.

Dada una estimación para la imagen SR, la máscara binaria resultante para la imagen LR de entrada, se determina a partir del error e_k^n definido en la ecuación 2.27. Esta máscara binaria puede calcularse mediante una convolución de e_k^n con un filtro paso bajo.

$$e_k^n = g_k - P_n(f^{(n)}) \quad (2.27)$$

Donde $P_n(f^{(n)})$ es $f^{(n)}$ pero con operaciones de desenfoque, deformación y diezmado.

La mayor parte de la energía de e_k^n se concentra cerca de los bordes de intensidad, donde cada valor positivo tiene un valor negativo vecino. Por lo tanto, el promedio local de e_k^n será cercano a cero.

A partir de este punto pueden suceder dos escenarios:

- Cuando e_k^n contiene valores distintos de cero debido a pequeñas regiones homogéneas atípicas, la imagen e_k^n tiende a ser no simétrica con respecto a cero, por lo que se esperará que el estimador de la mediana se vea afectado (sesgado) por los valores atípicos.
- Cuando e_k^n contiene valores altos, pero es simétrica con respecto a cero, se espera que el estimador de la mediana se vea menos afectado por estos valores atípicos.

2.5. Contextualización de las redes neuronales

En el campo del aprendizaje automático existen tres grandes vertientes siendo explotadas en la actualidad, estas son el aprendizaje supervisado, no supervisado y el reforzado [64].

- **Aprendizaje supervisado.** El aprendizaje supervisado es un enfoque en el que se entrena un sistema, proporcionándole las etiquetas o valores correctos que debe de predecir. Los problemas de aprendizaje supervisado se dividen en problemas de regresión y clasificación.
 - En un problema de regresión se intenta predecir los resultados de una salida continua. Es decir, asignar las variables de entrada a una función continua.
 - En un problema de clasificación lo que se quiere es predecir los resultados de una salida discreta. Es decir, asignar las variables de entrada a categorías discretas. El problema de la clasificación es similar al de regresión. No obstante, los valores que se desean predecir sólo adoptan un pequeño número de valores discretos, estos valores discretos son los que serán clasificados posteriormente en clases.
- **Aprendizaje no supervisado.** El aprendizaje no supervisado es un enfoque que permite abordar los problemas donde se cuenta con poca o ninguna información de cómo deberían ser los resultados. En este enfoque el sistema experimenta un conjunto de datos que contiene muchas características y luego aprende propiedades útiles de la estructura de este conjunto de datos. Sin embargo, se debe tener en cuenta que con este aprendizaje no hay retroalimentación basada en los resultados de la predicción.
- **Aprendizaje por refuerzo.** El aprendizaje por refuerzo es un enfoque que se basa en recompensar los comportamientos deseados y penalizar los no deseados durante el entrenamiento de un sistema. En esta aproximación se puede interpretar el entorno e interactuar con este, por lo que la retroalimentación se consigue gracias al proceso de prueba y error.

En este TFM se profundizará en el campo del aprendizaje profundo, que es un subcampo del aprendizaje automático. La mayor diferencia entre estos dos conceptos

es que, en el campo del aprendizaje automático, suele ser el humano quien le impone al modelo diseñado que características se desean extraer de los parámetros de entrada. No obstante, la aproximación del aprendizaje profundo es algo distinta, consistiendo en que estas características son extraídas directamente por la máquina. De esta forma, la máquina es capaz de realizar decisiones “inteligentes” por sí misma. En resumen, los modelos de aprendizaje automático clásicos son capaces de mejorar progresivamente en su función. Sin embargo, estos siguen necesitando de cierta orientación. Con un modelo de aprendizaje profundo, un algoritmo puede determinar por sí mismo si una predicción es precisa o no.

2.5.1. Redes Neuronales

El paradigma actual en el campo del aprendizaje profundo viene dado por la explotación de un modelo computacional capaz de emular el procesado de información del cerebro humano, a este modelo se le conoce como red neuronal. El núcleo de este tipo de aprendizaje profundo son las neuronas. Como se puede observar, el nombre de esta estructura está inspirado en el cerebro humano, imitando la forma en que las neuronas de las personas envían señales de una a otra. De hecho, la función que realiza una neurona en algoritmos de aprendizaje profundo puede quedar simplificada por tres factores: las entradas I , los pesos W y una función de activación σ .

- Las entradas son todos aquellos datos que procesan las neuronas a su entrada.
- Los pesos permiten decidir la importancia de las entradas, estimando si aumenta o disminuye la probabilidad de una hipótesis.
 - La hipótesis es la función creada por la red neuronal que se encarga de mapear los datos de entrada para conseguir una salida determinada. Al ponderar estas entradas y unirlas se llega a una posible solución.
- La función de activación define la salida de una neurona a partir de una serie de entradas. Este paso es muy importante en el aprendizaje profundo, ya que permite introducir suficientes no linealidades en el modelo y así aprender patrones más complejos en los datos. En definitiva, es toda aquella función umbral que establece una determinada salida al valor de la entrada.

Una formulación matemática que explica lo que sucede dentro de una neurona puede considerarse a partir de la ecuación 2.28.

$$\hat{y} = \sigma(i_1 \cdot w_1 + i_2 \cdot w_2 + \cdots + i_{n-1} \cdot w_{n-1} + i_n \cdot w_n) \quad (2.28)$$

Donde \hat{y} es la predicción que se obtiene a la salida, i_x son los datos de entrada, w_x son los pesos de las entradas y $\sigma(\cdot)$ es la función de activación usada. Una representación visual de este hecho se muestra en la Figura 2.16.

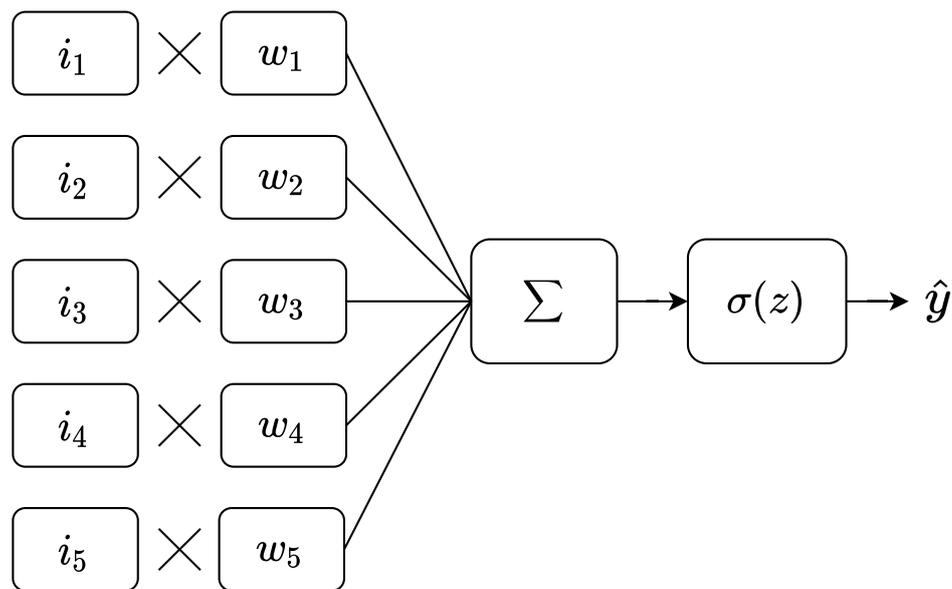


Figura 2.16: Funcionamiento simplificado de una neurona (I). Adaptada de [65]

2.5.1.1. Aprendizaje de una neurona

El objetivo de una neurona es ajustar los pesos basándose en muchos ejemplos de entradas y salidas. En este sentido, la salida de una neurona se estima como una suma ponderada de las entradas i con sus respectivos pesos w , teniendo en cuenta que se deben procesar con la función de activación pertinente. El modelo simplificado de una neurona, para que esta se ajuste lo más posible a un entorno realista, necesita de un factor de sesgo. Este factor de sesgo, denotado como b , es un parámetro adicional que se añade a la ecuación y es usado principalmente con la intención de ajustar la salida para unos datos concretos, tal como se observa en la Figura 2.17. De hecho, al tratarse de una constante, el factor de sesgo permite desplazar la función de activación hacia un lado del eje u otro. Una de las formas más sencillas de entender una red neuronal

es a través del modelo de regresión lineal. Si toda operación que realiza una neurona se simplifica a este modelo, se puede considerar que el factor de sesgo b es el encargado de modificar el eje de ordenadas de una recta entre dos puntos.

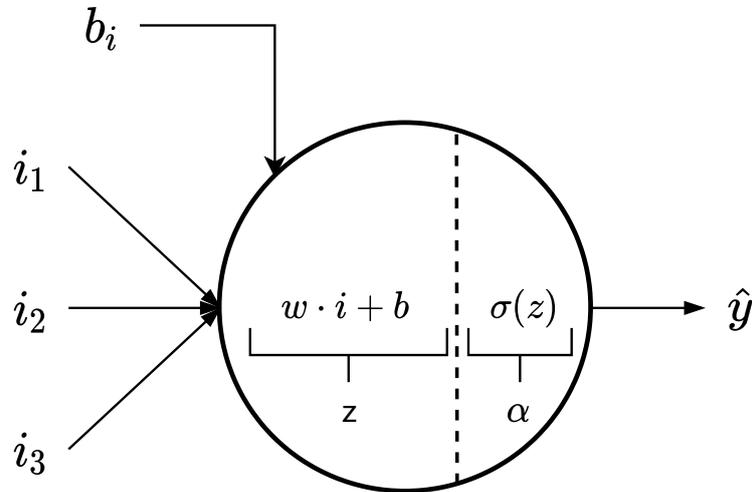


Figura 2.17: Funcionamiento simplificado de una neurona (II). Adaptada de [66]

Una vez explicado el concepto básico de una neurona, se puede extrapolar su funcionamiento a n neuronas distintas e interconectadas, permitiendo tomar la salida de una y utilizarla como entrada para la siguiente. De esta manera, quedaría conformada una red neuronal. Desde este punto se comienza con el concepto de capas, existiendo la capa de entrada, capa intermedia u oculta y la capa de salida, donde cada neurona de estas capas tiene un peso y un umbral asociados. El flujo de datos es enviado a la siguiente capa cuando la salida de cualquier neurona individual está por encima del valor umbral especificado. Estas redes son también llamadas redes *feedforward* ya que las conexiones entre estas no forman un ciclo. Las redes neuronales también se pueden clasificar en redes superficiales o redes profundas. La diferencia entre una y otra radica en el número de capas ocultas, si una red neuronal consta de una sola capa oculta se la clasifica como red superficial, mientras que si cuenta con $n > 1$ capas ocultas se la clasifica como una red neuronal profunda. En la Figura 2.18 se muestra el arquetipo de una red neuronal con n capas ocultas.

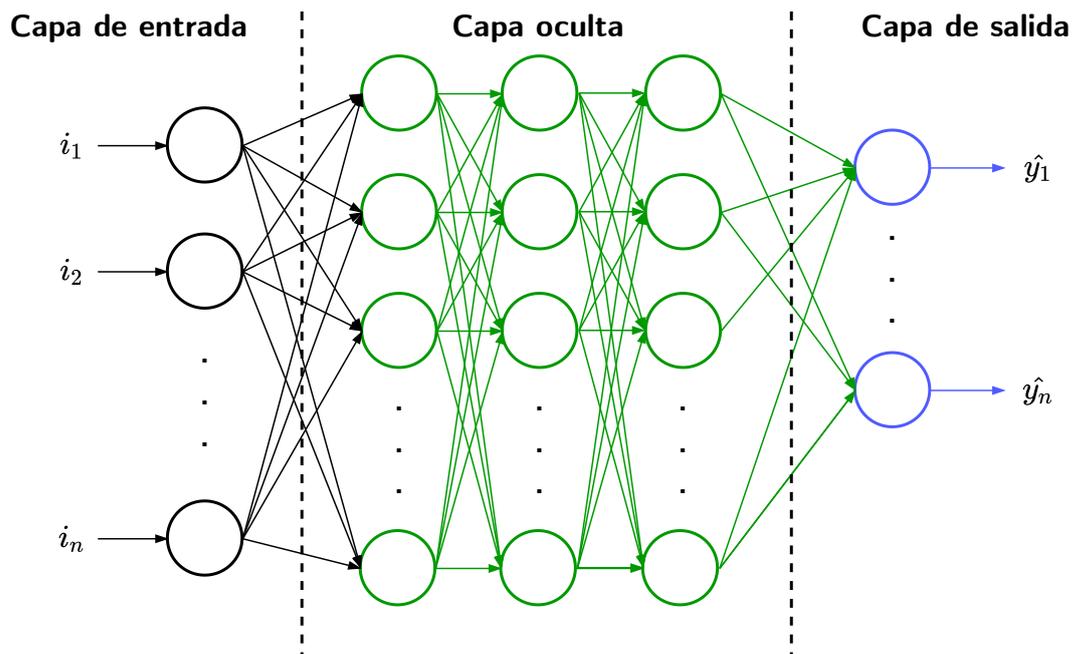


Figura 2.18: Generalización de una red neuronal profunda. Adaptada de [67]

2.5.1.2. Funciones de activación

Una función de activación toma la señal de salida de la célula anterior y la convierte en un factor que pueda utilizarse como entrada para la siguiente célula. Las funciones de activación son útiles porque añaden no linealidades a las redes neuronales, esto les permite aprender operaciones y patrones complejos. Una de las funciones de activación que se han usado de forma clásica en el campo del aprendizaje automático es la función sigmoide. No obstante, en la actualidad, una de las funciones de activación más usadas es la ReLU (*Rectified Linear Unit*). La Figura 2.19 representa gráficamente la forma de ambas funciones.

- Función sigmoide.** La función sigmoide es una función logística. Es decir, sea cual sea la entrada, se obtiene una salida que oscila entre 0 y 1 para todos los valores de z . Cuanto mayor sea la entrada (más positiva), más se acercará el valor de salida a un 1, mientras que cuanto menor sea la entrada (más negativa), más cerca estará el valor de salida de un 0. La ecuación 2.29 representa una función sigmoide.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.29)$$

- **Función ReLU.** La función ReLU se considera una función lineal a trozos. Es decir, actúa de forma lineal para valores positivos y devuelve un 0 para valores negativos. La ecuación 2.30 representa una función ReLU.

$$\sigma(z) = \max(0, z) \quad (2.30)$$

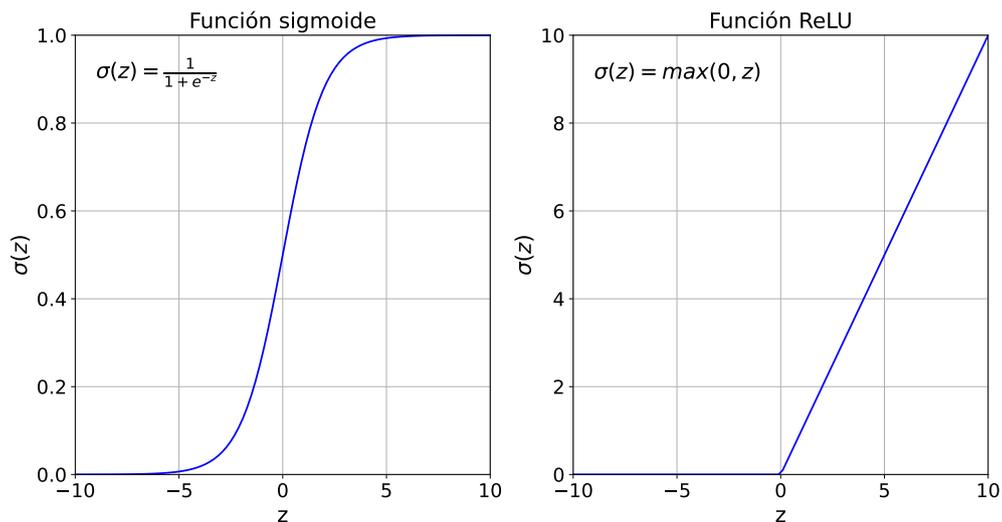


Figura 2.19: Representación de la función sigmoide y ReLU

2.5.1.3. Cálculo del error en la predicción

Para poder evaluar el desempeño y precisión de una red neuronal es necesario comprobar cuál ha sido su error al predecir una salida a partir de una serie de datos de entrada. Este paso se realiza utilizando una función de pérdida. Existen muchos tipos, cada una utilizada según el contexto de la situación.

Por ejemplo, algunas muy usadas en campo de las redes neuronales son:

- **Error absoluto medio o MAE (*Mean Absolute Error*).** Es una de las funciones de pérdida más sencillas de entender. Se define como la diferencia media absoluta entre los datos reales y las predicciones del modelo. Al realizar el operador de valor absoluto, el MAE indica como contribuye cada residuo proporcionalmente a la cantidad de error. Es decir, los errores más grandes contribuyen

linealmente al error global, por ello un MAE pequeño implica buena predicción del modelo. El MAE se describe matemáticamente según la ecuación 2.31.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y - \hat{y}| \quad (2.31)$$

- **Error porcentual absoluto medio o MAPE (*Mean Absolute Percentage Error*)**. El MAPE se define como el error promediado en formato porcentual y se describe matemáticamente según la ecuación 2.32.

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y - \hat{y}}{y} \right| \quad (2.32)$$

- **El error cuadrático medio o MSE (*Mean Square Error*)**. El MSE indica lo cerca que está una línea de regresión de los puntos predichos y se calcula de forma similar al MAE, solo que realizando el cuadrado de la diferencia entre el valor real y predicho antes de sumarlo. El efecto del término cuadrado en la ecuación del MSE es más evidente con la presencia de valores atípicos en los datos. Es decir, mientras que cada residuo en el MAE contribuye proporcionalmente al error total, el error crece de forma cuadrática en MSE. Esto significa, en última instancia, que los valores atípicos en los datos contribuirán a un error total mucho mayor en el MSE que en el MAE, penalizando al modelo cuando realiza predicciones que difieran mucho del valor real correspondiente. El MSE se describe matemáticamente según la ecuación 2.33.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y - \hat{y})^2 \quad (2.33)$$

- **La pérdida de Huber**. Esta función de pérdida es muy usada en operaciones de regresión, siendo menos sensible a valores atípicos en los datos que el MSE. De hecho, es considerada una combinación del MSE y el MAE. El MSE es una función de pérdida muy usada cuando se quiere aprender de los valores atípicos en los conjuntos de datos, mientras que con MAE se quieren evitar estos valores. La pérdida de Huber se considera cuadrática para valores pequeños del residuo ($y - \hat{y}$) y lineal para valores mayores. La pérdida de Huber se define matemáticamente según la ecuación 2.34.

$$L_{\delta}(y, f(x)) = \begin{cases} \frac{1}{2}(y - \hat{y})^2, & \text{para } |y - \hat{y}| \leq \delta \\ \delta (|y - \hat{y}| - \frac{1}{2}\delta), & \text{de lo contrario} \end{cases} \quad (2.34)$$

Donde δ es el umbral donde se usará una estrategia MAE o MSE.

2.5.1.4. Optimización del error

El siguiente paso tras elegir una función de pérdida es minimizar el error de esta. Es decir, encontrar un mínimo absoluto que represente el error en un momento dado. El uso de optimizadores facilita esta tarea, ya que estos son algoritmos o métodos utilizados para cambiar los atributos de una red neuronal de forma automática, como los pesos y la tasa de aprendizaje, con el fin de reducir las pérdidas. De esta manera, la forma que se tiene de cambiar estos parámetros viene definida por el optimizador usado. En la actualidad, de la misma forma que existen distintas funciones de pérdida, también se encuentran en muchos marcos de trabajo distintos tipos de optimizadores. Entre estos se puede nombrar el método del descenso de gradiente y el optimizador Adam. El primero representa un algoritmo clásico para optimizar una red neuronal, mientras que el segundo es uno de los optimizadores actuales más usados en el campo del aprendizaje profundo.

- **Descenso de gradiente.** El descenso de gradiente es uno de los algoritmos más presentes y básicos utilizado para optimizar una red neuronal. La función básica que realiza es el cálculo de la derivada de primer orden para una función de pérdida determinada. El cálculo de esta derivada sobre un punto permite encontrar la dirección que maximiza la función de pérdida, y, al añadir un signo negativo, el algoritmo de descenso por gradiente toma una dirección contraria. Es decir, permite determinar durante el entrenamiento donde se encuentra un mínimo. Se debe recalcar que en la actualidad se utiliza una versión concreta del descenso de gradiente, llamado descenso de gradiente estocástico o SGD (*Stochastic Gradient Descent*), que no es más que usar el algoritmo anterior de forma iterativa con mayor frecuencia. El algoritmo SGD queda explicado matemáticamente a partir de la ecuación 2.35.

$$\theta_n = \theta_{n-1} - \alpha \cdot \nabla J(\theta) \quad (2.35)$$

Donde α es la tasa de aprendizaje, ∇ es el operador de gradiente y $J(\theta)$ el valor de la función de pérdida.

■ **Optimizador Adam.** El optimizador Adam (*Adaptive Moment Estimation*) es un algoritmo capaz de regular por sí mismo su propia tasa de aprendizaje [68]. Es decir, es un método de tasa de aprendizaje adaptativo. Para poder realizar esta tarea se hace uso del primer y segundo momento, también llamados impulso de un gradiente.

- El momento N de una variable aleatoria se puede definir como el valor esperado de esa variable a la potencia n . La ecuación 2.36 muestra el cálculo del momento n .

$$m_n = E[X^n] \quad (2.36)$$

El primer momento se define como la media y el segundo momento se define como la varianza no centrada (durante el cálculo de la varianza se le resta la media). En primer lugar, los vectores del primer momento m y segundo momento v se inicializan a 0. El algoritmo se ejecuta iterativamente desde un tiempo $t = 1$, lo que implica actualizar los parámetros deseados. Dado un valor w que representa los pesos de las entradas, se necesita estudiar el caso $w_{(t-1)}$ y w_t . A partir de este punto se procesan tres operaciones diferentes.

- Se calculan las derivadas parciales para el tiempo actual a través de la ecuación 2.37.

$$g_t = f'(w_{t-1}) \quad (2.37)$$

- Se actualiza los valores del primer y segundo momento a través de la ecuación 2.38.

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{aligned} \quad (2.38)$$

- En este punto los momentos están sesgados ya que son inicializados en primera instancia a un valor 0. Para corregir los momentos sesgados se realiza un nuevo paso, descrito en la ecuación 2.39.

$$\begin{aligned}\hat{m}_t &= \frac{m_t}{1 - \beta_1} \\ \hat{v}_t &= \frac{v_t}{1 - \beta_2}\end{aligned}\tag{2.39}$$

- Finalmente, para actualizar un parámetro dado, por ejemplo el peso de una entrada w , se utiliza la ecuación 2.40.

$$w_t = w_{t-1} - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}\tag{2.40}$$

En las ecuaciones anteriores, β_1 es el factor de decaimiento para el primer momento con un valor típico de 0.9, β_2 es el factor de decaimiento para el segundo momento con un valor típico de 0.999 y ϵ suele tener un valor típico de $10^{(-8)}$. Estos parámetros pueden ser modificados según las necesidades del entrenamiento.

2.5.1.5. Propagación del error calculado

Una vez calculado el error de una red neuronal es necesario transmitir este error a través de todas las capas anteriores. El algoritmo usado actualmente para realizar esta acción se llama algoritmo de propagación hacia atrás o *Back-Propagation* y fue introducido en la década de 1960 pero popularizado en 1989 gracias al artículo publicado por Rumelhart, Hinton y Williams [69]. El concepto detrás de este algoritmo es muy similar al *feedforward* comentado anteriormente, la diferencia fundamental es como viaja la información del error. Es decir, para calcular el error de una capa determinada, es necesario conocer el error de la capa siguiente. El objetivo final del algoritmo es minimizar la función de pérdida ajustando los pesos y los sesgos de la red.

De forma resumida, el error puede quedar definido según [70] por la ecuación 2.41.

$$E_{n-1} = W_n^T \cdot E_n\tag{2.41}$$

Donde E_{n-1} sería el error que se desea calcular, W_n^T los pesos que intervienen en las capas posteriores y E_n el error de la capa actual. Entendiendo que la actualización de estos pesos dependerá del optimizador usado finalmente. No obstante, una forma genérica de explicar la actualización de los pesos es a partir de la ecuación 2.42.

$$W_n = W_n - \frac{\delta E_{n+1}}{\delta W_n} \quad (2.42)$$

El término $\frac{\delta E_{n+1}}{\delta W_n}$ representa la derivada parcial del error de la función de pérdida respecto al peso. La ecuación completa ofrece un concepto que intuitivamente se relaciona con la búsqueda del mínimo absoluto para una función de pérdida. A este término es necesario añadirle la tasa de aprendizaje α con la que se actualiza el valor de los pesos, obteniéndose la ecuación 2.43.

$$W_n = W_n - \alpha \frac{\delta E_{n+1}}{\delta W_n} \quad (2.43)$$

La tasa de aprendizaje α es un valor entre 0 y 1. Cuanto menor sea esta, menor será el cambio en los pesos.

2.5.2. Redes neuronales convolucionales

Las redes neuronales convolucionales o CNN (*Convolutional Neural Networks*) son un tipo especializado de red neuronal que se aplica fundamentalmente al procesamiento de imágenes digitales [64]. Se considera que una red neuronal es convolucional cuando aplica, en al menos una de sus capas, la operación matemática de convolución en lugar de la multiplicación general de matrices.

2.5.2.1. La operación de convolución

La convolución en el campo de la imagen consiste en una operación lineal entre una imagen y un filtro o *kernel*, obteniendo a la salida de esta operación una nueva imagen. El valor de los píxeles de esta imagen I se calcula como una suma ponderada de los píxeles de la imagen original con el núcleo de convolución K [64], [71]. Esta operación se puede definir a través de la ecuación 2.44.

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n) \quad (2.44)$$

La fórmula anterior denota que la convolución se realiza desplazando el filtro sobre distintas regiones de la imagen, teniendo en cuenta que esta región debe ser del mismo tamaño que el filtro. El resultado es la suma del producto elemento a elemento entre ambas estructuras [72]. Finalmente, se obtiene un valor numérico que indica el valor

de la probabilidad de que la característica que representa el filtro usado se muestre en la región de la imagen analizada. Una representación de la operación de convolución se muestra en la Figura 2.20.

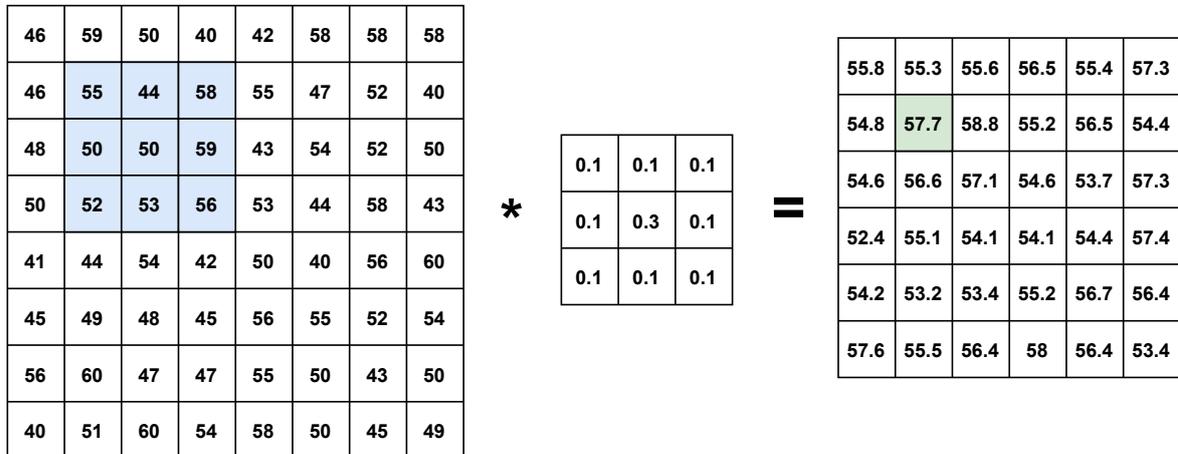


Figura 2.20: Ejemplo de una convolución de matrices

Normalmente existen cuatro parámetros del filtro que conviene tener en cuenta: su tamaño, el número de filtros, el relleno o *padding* y el parámetro de paso o *stride*.

- **Tamaño del filtro.** Denota el alto y ancho del filtro usado para recorrer la imagen y generar una abstracción de esta.
- **El número de filtros.** Este parámetro denota el número total de convoluciones que se realizan en paralelo, una por cada filtro. Cada filtro aumenta el número de características que son extraídas de una imagen determinada. Es decir, la profundidad de la imagen resultante depende de los filtros aplicados.
- **El relleno.** Durante el entrenamiento de una red convolucional existen ocasiones donde el filtro no se ajusta perfectamente a la imagen de entrada. Para estos casos se utilizan las estrategias de relleno válido o relleno equivalente.
 - Se entiende por relleno válido al método que restringe la salida de la operación de convolución a las regiones donde el filtro se superpone completamente, esto termina provocando que la salida sea más pequeña que la región original, puesto que pierde los bordes.
 - La estrategia equivalente permite que la salida sea de tamaño idéntico a la entrada. Esto se realiza si, por el contexto de la aplicación, se necesita

aplicar una convolución pero no se desea reducir el tamaño de la imagen. Para evitar la pérdida de los bordes es necesario añadir bordes artificiales a la estructura de entrada, el objetivo de este proceso es que el filtro pueda superponerse completamente en todas las posiciones de la entrada original. Esto se realiza rellenando la imagen con ceros.

- El parámetro de paso.** Se define como el número de desplazamientos de píxeles sobre toda la matriz de entrada. Si este parámetro es 1, entonces el filtro se mueve 1 píxel hacia una dirección. En caso de usar un paso más grande, el filtro no pasaría por todos los elementos, por lo que el tamaño de salida se vería reducido. Este método es muy usado para ajustar el tamaño de las imágenes siempre que pasen por una capa convolucional.

Un ejemplo de distintos formatos de relleno y paso se muestra en la Figura 2.21.

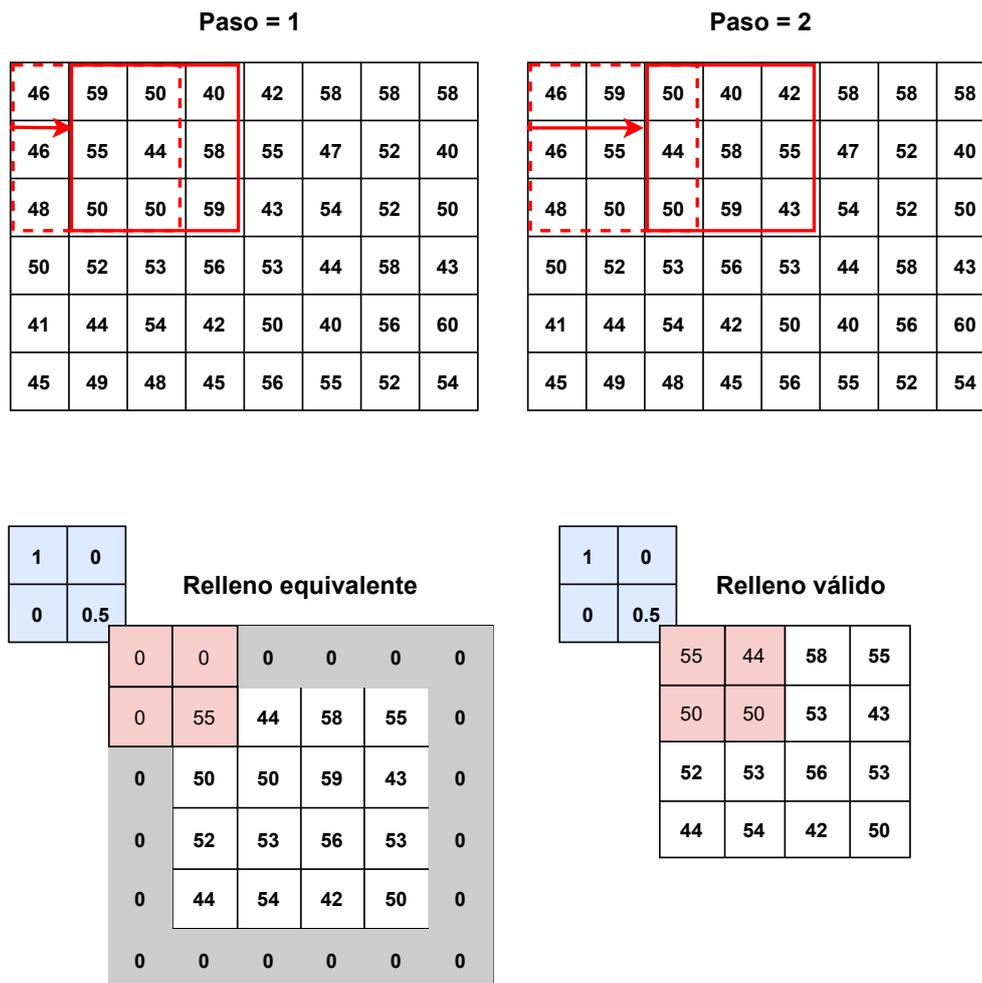


Figura 2.21: Proceso de convolución con distintos rellenos y parámetros de paso

Después del proceso de convolución se genera un conjunto de mapas de características que representa información de la imagen. Para mejorar el aprendizaje de una red neuronal es necesario aplicar no linealidades a través de una función de activación. Una bastante usada después de las capas convolucionales es la ReLU.

2.6. Contexto de las redes neuronales en la súper-resolución

En la actualidad, centrando el estudio del aprendizaje profundo en el campo de la súper-resolución, existe una gran variedad de modelos propuestos. Estos modelos se centran en un aprendizaje supervisado. Es decir, se entrena un algoritmo tanto con imágenes LR como con las correspondientes imágenes HR. Uno de los primeros estudios que se realizó en el campo del aprendizaje profundo con la intención de mejorar la resolución de una imagen está registrado en el artículo “Image Super-Resolution Using Deep Convolutional Networks” [73].

2.6.1. Método SRCNN

El método SRCNN (*Super-Resolution Convolutional Neural Network*) es propuesto por Chao Dong en 2015, siendo considerado uno de los trabajos precursores en el ámbito del aprendizaje profundo aplicable a la tarea de súper-resolución. Este método para la súper-resolución de una sola imagen se basa en aprender un mapeo extremo a extremo entre imágenes LR y HR. Este mapeo es representado como una red neuronal convolucional profunda capaz de procesar una imagen LR a su entrada, generando una imagen SR de un tamaño idéntico [73].

2.6.1.1. Formulación del método

En primera instancia, se considera una imagen original X que ha sido escalada al tamaño deseado mediante una interpolación bicúbica. Si la imagen interpolada se le denomina como Y , entonces el objetivo final es la recuperación de una imagen $F(Y)$ a partir de Y , siendo $F(Y)$ lo más parecida a la imagen original X . El objetivo es que la red aprenda un mapeo F , que consta fundamentalmente de tres etapas [73]. La

arquitectura que se propone en SRCNN se puede observar en la Figura 2.22.

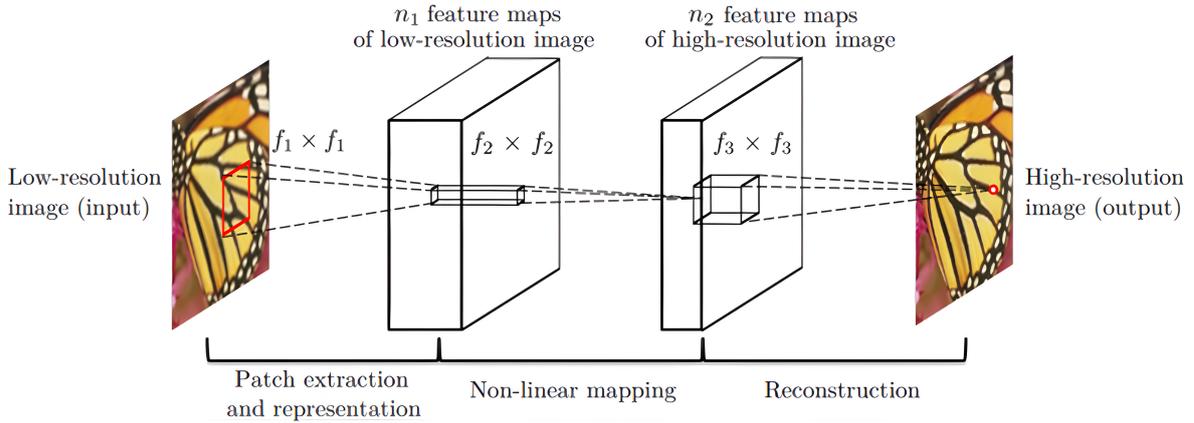


Figura 2.22: Arquitectura del método SRCNN. Extraída de [73]

La primera etapa de la red, la extracción y representación de parches, puede ser expresada como la operación F_1 , esta operación se modela con la ecuación 2.45.

$$F_1(Y) = \max(0, W_1 * Y + B_1) \quad (2.45)$$

Donde W_1 representa los filtros aplicados, B_1 representa los sesgos y la operación $*$ es la convolución de matrices. El factor W_1 se corresponde con un número n_1 de filtros de un tamaño $c \times f_1 \times f_1$, donde c es el número de canales de la imagen. El parámetro f_1 es el tamaño espacial del filtro. De forma intuitiva se puede decir que W_1 aplica n_1 convoluciones a la imagen y cada convolución tiene un filtro de tamaño $c \times f_1 \times f_1$. Por lo tanto, la salida se compone de n_1 mapas de características. En este caso, el factor B_1 es un vector con n_1 dimensiones, por lo que cada elemento es asociado a un filtro. Finalmente, $\max(0, x)$ representa la función de activación ReLU para rectificar los resultados de los filtros.

La segunda capa, el mapeado no lineal, convierte cada uno de los vectores dimensionales n_1 en un vector n_2 . Esta operación es equivalente a la aplicación de n_2 filtros de 1×1 , pudiendo generalizar la operación a filtros mayores como 3×3 o 5×5 . La transformación no lineal no se realiza sobre la imagen de entrada, sino sobre los parches del mapa de características generado por una capa anterior. La segunda capa de la red puede ser expresada como una operación F_2 según la ecuación 2.46.

$$F_2(Y) = \max(0, W_2 * F_1(Y) + B_2) \quad (2.46)$$

Donde W_2 contiene n_2 filtros de tamaño $n_1 \times f_2 \times f_2$ y B_2 es un vector dimensional n_2 . Cada uno de los vectores n_2 de salida de la transformación no lineal se consideran conceptualmente como una representación de un parche HR que se usará para la reconstrucción. De forma general, en los métodos tradicionales de súper-resolución, las predicciones de parches HR superpuestos a menudo se suelen promediar para producir la imagen final completa. Este promediado se puede considerar como un filtro predefinido en un conjunto de mapas de características. La fortaleza de utilizar redes neuronales profundas es la capacidad de aprendizaje para este filtro, ya que pueden adaptar este a distintos datos de entrada, en lugar de usar uno genérico.

La capa convolucional final sigue esta filosofía y se centra en producir la imagen deseada $F(Y)$, esto queda reflejado en la ecuación 2.47.

$$F(Y) = W_3 * F_2(Y) + B_3 \quad (2.47)$$

Donde W_3 corresponde a c filtros de tamaño $n_2 \times f_3 \times f_3$ y B_3 es un vector con c dimensiones. El parámetro c corresponde al número de canales de la imagen de entrada. De esta manera, el modelo es capaz de reconstruir una imagen de salida con la estructura esperada. Al juntar las tres operaciones anteriores se consigue formar una red neuronal convolucional específica, donde todos los pesos y sesgos de filtrado necesitan ser optimizados. Esta optimización se realiza a través del SGD, pero puede ser modificado con otros optimizadores como el Adam. Por otro lado, la función de pérdida usada en el artículo original era la MSE. La implementación oficial en MATLAB de SRCNN puede ser obtenida a partir de [74].

A partir de la publicación del método SRCNN se presentaron una gran serie de artículos mejorando la idea básica que se planteaba. Es decir, el uso de redes neuronales convolucionales a través de distintas arquitecturas y metodologías para aplicar súper-resolución en imágenes digitales. Esta nueva corriente sigue avanzando en la actualidad y cuenta con una gran cantidad de innovaciones en el campo del aprendizaje profundo. Por un lado, conviene resaltar los tipos más conocidos de redes neuronales en la actualidad, donde se destaca el uso de redes neuronales lineales, redes neuronales recursivas, redes neuronales residuales y redes generativas adversarias. Por otro lado, es importante nombrar el tipo de muestreo que se le realiza a la imagen para poder

implementar el factor de escala, si es un muestreo clásico por interpolación o muestreo basado en aprendizaje, si el muestreo se realiza antes, en medio o al final de la red neuronal, además de otros factores. De la misma manera, el tipo de aprendizaje de estas arquitecturas interviene con un gran peso en el resultado final del proceso de súper-resolución, algunas estrategias de aprendizaje son la PSNR, las pérdidas de píxel (L1 y L2), pérdidas de contenido y las pérdidas adversarias, entre otras [75], [76].

2.6.2. Tipos de redes neuronales

En esta sección se presentan distintos tipos de redes neuronales usadas actualmente en el campo de la súper-resolución.

2.6.2.1. Redes neuronales lineales

Las redes lineales en el campo de la súper-resolución pueden ser vistas como una red *feedforward*, ya que tienen una estructura sencilla que consiste en un único camino que recorre el flujo de la señal, sin conexiones de salto ni ramificaciones múltiples. De forma general, para este tipo de redes, se apilan varias capas convolucionales y la imagen de entrada LR fluye secuencialmente desde las capas iniciales hasta las finales. Algunas redes lineales, como la propuesta por Chao Dong, aprenden a reproducir la imagen residual, siendo esta la diferencia entre las imágenes LR y HR.

2.6.2.2. Redes neuronales recursivas

Este tipo de red neuronal está especializada en procesar una secuencia de valores de longitud variable, utilizando la información de la salida de la red en un tiempo anterior. En el contexto de la súper-resolución, las redes neuronales recursivas implementan capas convolucionales conectadas recursivamente o unidades conectadas recursivamente. Es decir, la recursividad se encuentra en aplicar los mismos módulos varias veces de forma iterativa. La principal motivación de estos diseños consiste en descomponer progresivamente el problema de la súper-resolución, a priori complejo, en un conjunto de problemas más sencillos, que son más fáciles de resolver.

2.6.2.3. Redes neuronales residuales

Este tipo de redes parten de una premisa opuesta a las redes lineales. Las redes neuronales residuales utilizan un aprendizaje residual a través de conexiones de salto en el diseño de la red, de esta forma consiguen evitar el desvanecimiento de gradientes y hace posible el diseño de redes muy profundas.

- **Desvanecimiento de gradientes.** Si el resultado de las derivadas parciales es muy pequeño, es posible que durante la propagación de este error ocurra un desvanecimiento, desapareciendo el dato que posibilita el aprendizaje de la red.

En este enfoque, los algoritmos aprenden el residuo, de las frecuencias altas entre la imagen LR y la imagen HR. La popularidad de este tipo de redes neuronales se incrementó con la presentación de ResNet. Esta red fue propuesta en 2015 por Kaiming He en el artículo “Deep Residual Learning for Image Recognition” [77]. Las conexiones de salto que caracterizan a las redes residuales son representadas con una transición entre capas. La arquitectura propuesta en ResNet se observa en la Figura 2.23.

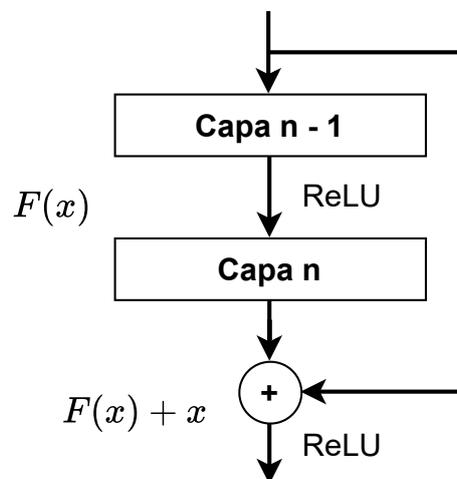


Figura 2.23: Conexión de salto en una ResNet. Adaptada de [77]

La incorporación de este puente entre capas no añade parámetros extras al entrenamiento, ya que su finalidad es sumar la salida de la capa $n - 1$ a la capa $n + 1$. En muchos casos los vectores que fluyen de la capa $n - 1$ a $n + 1$ no tienen la misma dimensión, ya que operaciones como la convolución reducen la resolución espacial de una imagen. En estos casos se deben ajustar las dimensiones de la capa $n - 1$ mediante operaciones como la convolución.

2.6.2.4. Redes generativas adversarias

Uno de los cambios del paradigma actual en el campo de la súper-resolución se debe a la concepción e implementación de un nuevo tipo de redes neuronales llamadas GAN (*Generative Adversarial Networks*) por parte de Ian Goodfellow, entre otros [78]. La estructura de una GAN viene definida por dos agentes que compiten entre sí, un generador y un discriminador. De este modo, cuando se habla de una red GAN en realidad se hace referencia a dos tipos de redes distintas y antagónicas. Tanto el generador como el discriminador pueden diseñarse usando diferentes redes, como las comentadas anteriormente. Dado que las redes GAN fueron fundamentalmente pensadas para problemas relacionados con el campo del procesamiento de la imagen, se entiende que las CNN sean las más adecuadas. El diseño de una red GAN se muestra en la Figura 2.24.

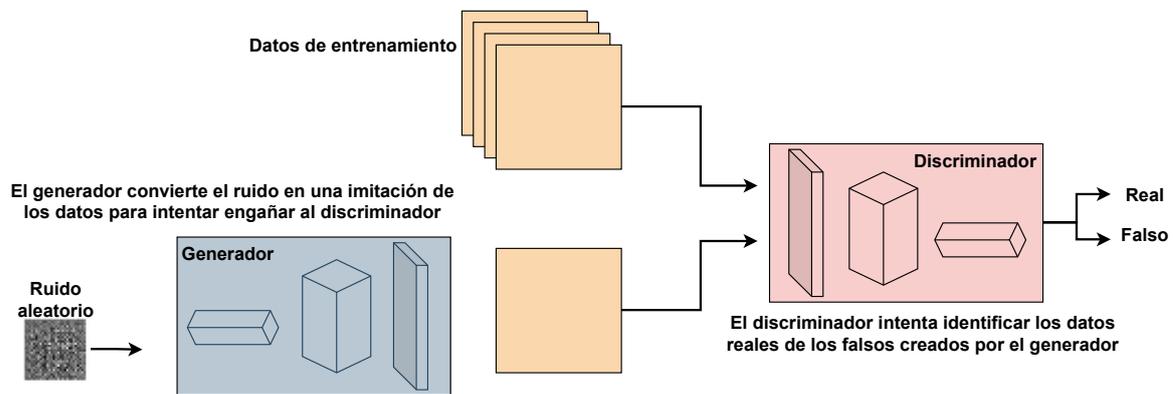


Figura 2.24: Diseño de una red GAN. Adaptada de [79]

En primera instancia, para entender el funcionamiento de una red GAN es necesario definir que es un modelo generativo y discriminativo dentro del aprendizaje profundo. Por un lado, se entiende como modelo generativo a un tipo concreto de red capaz de generar nuevos datos que se ajusten a una determinada distribución de valores con los que ha sido entrenado. Por otro lado, el modelo discriminativo es el encargado de clasificar los datos que genera el modelo generativo, denotando si son reales o falsos.

Los primeros generadores implementados en el estado de la técnica son entrenados a partir de una cierta distribución de probabilidad (vector numérico), teniendo como objetivo final la transformación de este ruido en datos similares a los del conjunto de entrenamiento. Es decir, el generador no tiene constancia de la clase de datos que intenta generar. Este dato generado se transmite al discriminador, decidiendo este si

la entrada proviene de un verdadero conjunto de datos de entrenamiento o de datos generados de forma artificial. Esta decisión del discriminador se transforma en un residuo o pérdida que se transmite al generador, intentando en la siguiente iteración del entrenamiento optimizar la generación de los datos para que coincidan con el conjunto de entrenamiento real. De la misma forma, el discriminador también cuenta con una serie de pérdidas que ocurren cuando el generador es capaz de engañarle. Después de un cierto número de iteraciones y de recibir suficiente información del discriminador, el generador aprenderá a engañar al discriminador, obteniendo un modelo generativo capaz de crear nuevas muestras que comparten propiedades muy similares a la distribución de muestras del entrenamiento.

2.6.3. Metodologías de muestreo aplicadas a la súper-resolución

En esta sección se estudiarán las distintas soluciones para asegurar un determinado factor de escala en la imagen. Es decir, que operaciones se efectúan sobre la imagen LR para mejorar su resolución espacial.

2.6.3.1. Muestreo de la imagen

Este paso es de gran importancia para implementar el factor de escala en los métodos de súper-resolución. Normalmente se estudian dos factores para realizar el muestreo, en qué momento realizarlo y con qué técnica.

Por un lado, es importante denotar en qué momentos se puede realizar el muestreo de una imagen [76], [80]:

- **Muestreo temprano.** Los primeros diseños de muestreo presentados en el estado de la técnica consisten en redes lineales que interpolan la imagen LR para que coincida con el tamaño deseado de la salida HR. Esta solución se basa en obtener una imagen de mayor resolución para luego refinarla usando distintos tipos de redes neuronales, puesto que en muchas ocasiones aprender directamente el mapeo del espacio entre una imagen LR y HR es complicado.
- **Muestreo tardío.** Estos diseños nacen de uno de los problemas fundamentales del muestreo temprano, la operación de interpolación al inicio de la red neuronal

puede llegar a ser costosa desde un punto de vista computacional, sobre todo cuando la imagen de entrada es de un gran tamaño. Este tipo de redes realizan primero el mapeo entre la imagen LR y HR, para luego efectuar la operación de muestreo.

- **Muestreo progresivo.** Si bien el muestreo tardío es capaz de reducir el coste computacional de una red neuronal, sigue teniendo algunos inconvenientes. Por ejemplo, el muestreo se realiza en un solo paso, lo que aumenta la dificultad de aprendizaje cuando se trabaja con factores de escala grandes como x4 o x8. De la misma manera, para cada factor de escala se requiere de un modelo individual, por lo que no se puede suplir la necesidad de un modelo de súper-resolución multiescala. Los diseños basados en muestreo ascendente progresivo adoptan un enfoque distinto, ya que son basados en cascadas de distintas CNN que reconstruyen progresivamente imágenes de mayor resolución. Es decir, en cada capa las imágenes se muestrean a una mayor resolución, descomponiendo una tarea de relativa dificultad en tareas más sencillas. De esta forma, con un solo modelo, se puede realizar súper-resolución multiescala, ya que es posible obtener directamente la salida de cada una de las CNN que componen la cascada.
- **Método iterativo de muestreo.** Este tipo de diseños para aplicar factor de escala permite capturar mejor las características y dependencias mutuas que existe entre las imágenes LR y HR, ya que se basa fundamentalmente en el proceso de retroproyección. Este proceso se basa en calcular el error de reconstrucción actual y luego fusionarlo con los anteriores para ajustar la intensidad de la imagen HR. De esta forma, es posible extraer mejor las relaciones profundas entre los pares de imágenes LR-HR y por lo tanto proporcionar resultados de reconstrucción de mayor calidad.

La representación visual de los distintos momentos donde aplicar el muestreo a una imagen durante el proceso de súper-resolución se muestra en la Figura 2.25.

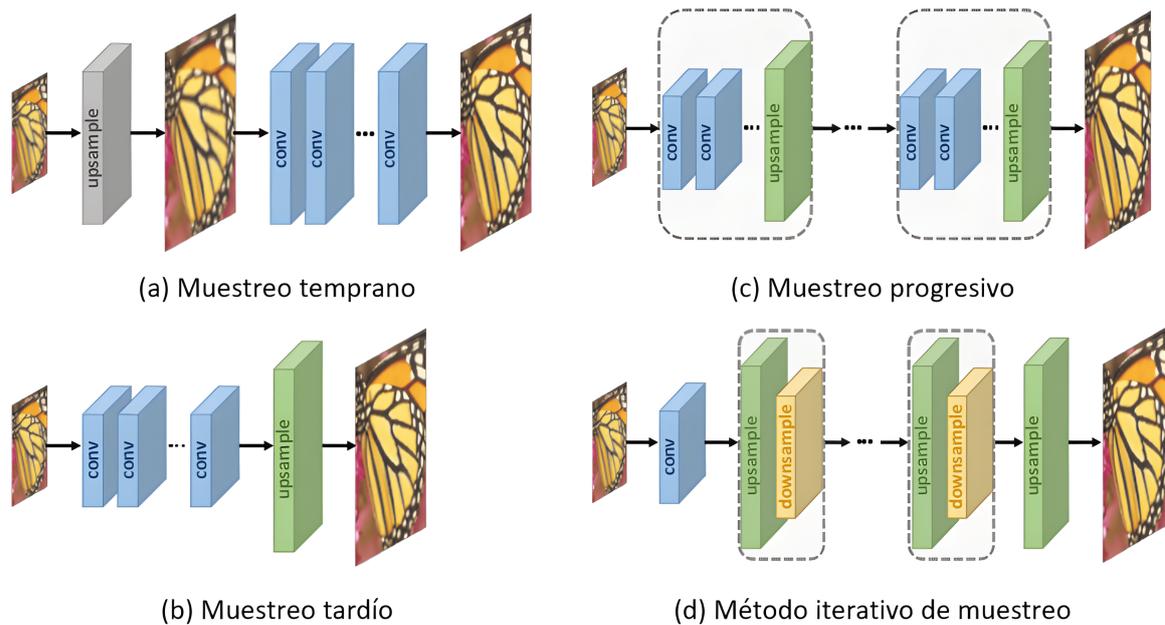


Figura 2.25: Muestreo en el proceso de súper-resolución. Adaptada de [80]

Por otro lado, es necesario entender que tipos de métodos se pueden usar para realizar el proceso de muestreo [76], [80]:

- **Muestreo basado en la interpolación.** Los muestreos basados en interpolación aplican un cambio de tamaño o escalado en las imágenes digitales. Los métodos tradicionales de interpolación incluyen la interpolación del vecino más cercano, bilineal y bicúbica, entre otros. Dado que estos métodos son interpretables y fáciles de implementar, algunos de ellos se siguen utilizando ampliamente en los modelos de súper-resolución basados en CNN.
- **Muestreo basado en el aprendizaje.** La tendencia actual ha sustituido los métodos de interpolación clásicos por un tipo de muestreo basado en el aprendizaje. Es decir, aprender el factor de escala de forma automatizada mediante técnicas como la convolución traspuesta o la capa de subpíxeles.
 - **Capa de convolución traspuesta.** La capa de convolución traspuesta o deconvolución intenta realizar una transformación opuesta a una convolución normal. Es decir, predecir la posible entrada basándose en mapas de características de tamaño similar a la salida de la convolución. Este tipo de técnica aumenta la resolución de la imagen expandiéndola mediante la inserción de ceros y realizando la operación de convolución.

- Capa de subpíxeles.** La capa de subpíxeles fue introducida por primera vez por Wenzhe Shi [81], siendo otro ejemplo de capa de muestreo basada en el aprendizaje de extremo a extremo. Uno de los conceptos más importantes propuestos por los autores del artículo es la convolución de subpíxeles. La capa de convolución de subpíxeles hace uso del proceso de convolución de una forma distinta a la convolución traspuesta. La idea fundamental es realizar dos procesos distintos: una operación convolucional ordinaria y una reordenación de los píxeles. La primera es necesaria para extraer distintas características de la imagen LR, mientras que la segunda consiste en la combinación de cada píxel de los mapas de características de múltiples canales en un área cuadrada para la imagen de salida. De esta forma, cada píxel de los mapas de características equivale al subpíxel de la imagen generada.

Un ejemplo que ilustra el muestreo a través de las capas de convolución traspuesta y capa de subpíxeles se observa en la Figura 2.26.

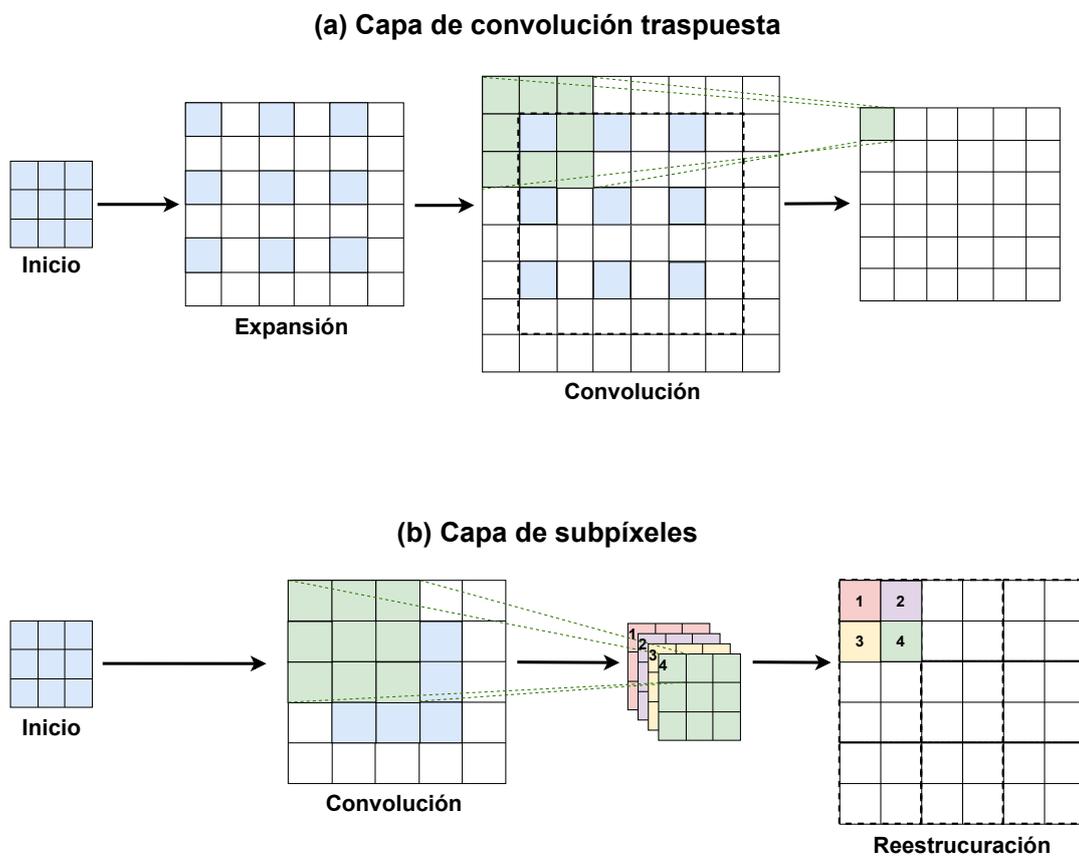


Figura 2.26: Ejemplos de muestreo basado en aprendizaje. Adaptada de [80]

2.6.4. Funciones de pérdida

Las funciones de pérdida se utilizan para medir la diferencia entre la imagen SR generada y la imagen HR real. Esta diferencia o error se utiliza para optimizar el modelo de aprendizaje supervisado [80].

2.6.4.1. Pérdida de píxeles

La pérdida de píxel es la función de pérdida más simple usada para entrenar un modelo de súper-resolución. En esta, cada píxel de la imagen SR se compara directamente con cada píxel de la imagen HR. Las funciones de pérdida más populares en este aspecto son las pérdidas L1 y L2. Por un lado, la pérdida L1 mide la diferencia de píxeles entre dos imágenes mediante el error absoluto medio o MAE. Por otro lado, la pérdida L2 realiza la misma función con el error cuadrático medio o MSE [76], [80]. Las pérdidas L1 y L2 se presentan en la ecuación 2.48.

$$\begin{aligned}\mathcal{L}_{pixel_1}(\hat{I}, I) &= \frac{1}{hwc} \sum_{i,j,k} |\hat{I}_{i,j,k} - I_{i,j,k}| \\ \mathcal{L}_{pixel_2}(\hat{I}, I) &= \frac{1}{hwc} \sum_{i,j,k} (\hat{I}_{i,j,k} - I_{i,j,k})^2\end{aligned}\tag{2.48}$$

Donde h , w y c son la altura, el ancho y el número de canales de las imágenes evaluadas respectivamente. El factor \hat{I} es la imagen SR e I es la imagen HR.

2.6.4.2. Pérdida de contenido

Esta pérdida evalúa la calidad de la imagen de forma perceptiva. Una forma generalizada de hacerlo es comparando las características de alto nivel de la imagen SR y la imagen HR. Estas características de alto nivel pueden extraerse al introducir ambas imágenes por una red de clasificación de imágenes previamente entrenada. Si se denomina esta red como ϕ y las representaciones de alto nivel extraídas en la capa l como $\phi^l(I)$, la pérdida de contenido se puede representar como la ecuación 2.49.

$$\mathcal{L}_{contenido}(\hat{I}, I; \phi, l) = \frac{1}{h_l w_l c_l} \sqrt{\sum_{i,j,k} (\phi_{i,j,k}^{(l)}(\hat{I}) - \phi_{i,j,k}^{(l)}(I))^2}\tag{2.49}$$

Donde h_l , w_l y c_l son la altura, la anchura y el número de canales de las representaciones en la capa l respectivamente.

A diferencia de la pérdida de píxeles, esta pérdida fomenta que la imagen SR sea similar a la imagen HR de forma perceptiva. Por esta razón también se la conoce como pérdida perceptiva.

2.6.4.3. Pérdida adversaria

Esta pérdida se utiliza en toda arquitectura que incluya una red GAN, dividiéndose en dos pérdidas distintas:

- La pérdida del generador al no poder producir una imagen que engañe al discriminador.
- La pérdida del discriminador al no poder distinguir entre los datos generados y los datos reales.

La nomenclatura matemática para las pérdidas adversarias es menos genérica que las anteriores, ya que depende de cómo los creadores de una arquitectura determinada hayan implementado el generador y el discriminador.

2.6.5. Métricas de calidad

En esta sección se nombran las métricas de calidad más usadas a la hora de probar el rendimiento de un método de súper-resolución.

2.6.5.1. PSNR

La PSNR es una de las medidas de calidad de reconstrucción más populares en el procesamiento digital de la imagen. En el caso de la súper-resolución de imágenes, la PSNR se define mediante el valor máximo de los píxeles y el error cuadrático medio entre las imágenes. Dada una imagen real I con N píxeles y la reconstrucción \hat{I} , la PSNR entre I e \hat{I} se puede definir según la ecuación 2.50.

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{L^2}{\frac{1}{N} \sum_{i=1}^N (I - \hat{I})^2} \right) \quad (2.50)$$

Donde L es igual a 255 en los casos generales que utilizan representaciones de 8 bits y 65535 en representaciones de 16 bits.

2.6.5.2. SSIM

El SSIM es un modelo basado en la percepción, que cuantifica cuanto se degrada una imagen después de su procesado. Esta métrica está basada en tres comparaciones relativamente independientes, como son la luminancia, el contraste y la estructura. La representación comúnmente utilizada de la fórmula SSIM es la que se muestra en la ecuación 2.51. La diferencia principal respecto PSNR es que la primera estudia la diferencia absoluta entre valores, mientras que esta tiene en cuenta la interdependencia de píxeles que se encuentren muy cerca unos de otros, ya que esta relación acarrea una información adicional sobre la estructura de los objetos que se están estudiando.

$$\text{SSIM}(I, \hat{I}) = \frac{(2\mu_I\mu_{\hat{I}} + C_1)(\sigma_{I\hat{I}} + C_2)}{(\mu_I^2 + \mu_{\hat{I}}^2 + C_1)(\sigma_I^2 + \sigma_{\hat{I}}^2 + C_2)} \quad (2.51)$$

Donde μ_x representa la media de la imagen, σ_x representa la desviación estándar de la imagen, $\sigma_{I\hat{I}}$ representa la covarianza entre dos imágenes y C_1, C_2 son constantes establecidas para evitar la inestabilidad.

Capítulo 3

Estado de la técnica

En este capítulo se discuten de forma teórica algunos de los métodos de súper-resolución basados en redes neuronales más presentes en el estado de la técnica.

3.1. Métodos basados en redes neuronales

Durante el transcurso de este TFM se ha realizado una búsqueda rigurosa sobre el estado de la técnica que incumben distintos métodos de súper-resolución para imágenes digitales, encontrando y referenciando todo tipo de metodologías aplicables. Algunos de estos métodos pueden basarse en la interpolación de matrices, transformaciones en el dominio frecuencial, métodos bayesianos y métodos iterativos, entre otros. De la misma manera, se ha observado un punto de inflexión en el estado de la técnica actual, debido fundamentalmente a la implementación práctica de algoritmos de aprendizaje profundo, más concretamente aquellos que se basan en redes neuronales convolucionales. A partir la propuesta presentada por Chao Dong aparece toda una nueva corriente de métodos en el campo de la súper-resolución que conviene ser estudiados.

3.1.1. VDSR

El método VDSR (*Very Deep Super Resolution*) es uno de los enfoques clásicos del estado de la técnica de la súper-resolución a partir del aprendizaje profundo y que se presenta en 2016 por parte de Jiwon Kim [82] como una evolución directa al SRCNN. La estructura que emplea el método VDSR es similar a la del SRCNN, pero recibe su nombre al profundizar en el número de capas por las que pasa la imagen LR durante el

entrenamiento de la red, logrando mejores resultados. Al igual que el método SRCNN, el VDSR utiliza una interpolación bicúbica como método de muestreo en la etapa de entrada, tratando los mapas de características a la misma escala que la salida de la red. La arquitectura del método VDSR se ilustra en la Figura 3.1.

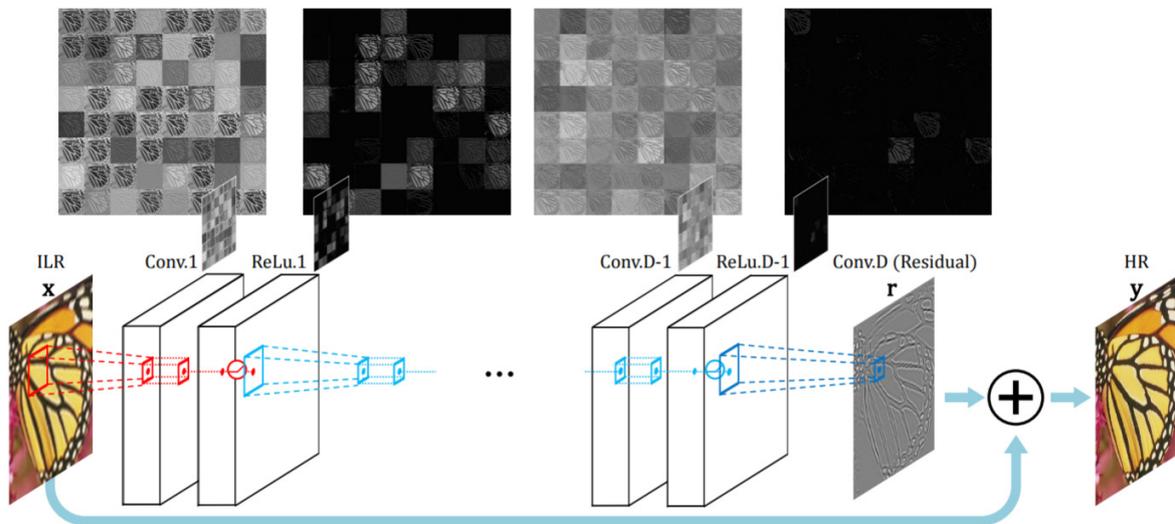


Figura 3.1: Arquitectura de VDSR. Extraída de [82]

3.1.1.1. Proceso de entrenamiento y función de pérdida

Para aplicar el proceso de súper-resolución, el método VDSR propone una cascada tanto de capas convolucionales como de capas ReLU de forma repetida. Una imagen ILR (*Interpolated Low-Resolution*) pasa por las capas que componen la arquitectura y se transforma en una imagen residual en la capa final. La red neuronal suma esta imagen residual con la imagen ILR original, obteniendo a su salida una imagen HR.

Las fases por las que pasa la imagen LR se pueden resumir en los siguientes puntos:

1. La imagen LR se interpola y pasa a llamarse ILR para ser introducida en la red.
2. La primera capa opera sobre la imagen ILR de entrada, mientras que la última capa es utilizada para la reconstrucción de la imagen.
3. La imagen ILR pasa por $d - 1$ capas de convolución y ReLU intermedias. Cada capa de convolución cuenta con 64 filtros de tamaño 3×3 . Es decir, se obtienen 64 mapas de características de la imagen por cada capa.

4. Finalmente, la salida de la última etapa de convolución, la imagen residual, se añade a la imagen ILR original y se obtiene la imagen HR.

Para poder encontrar los parámetros óptimos del modelo es necesario minimizar el error entre la imagen predicha y la imagen original. En este caso se denota x a la imagen de entrada interpolada (ILR) e y a la imagen original (HR). Por lo tanto, dado un conjunto de entrenamiento determinado, el objetivo es encontrar un modelo f que sea capaz de predecir $\hat{y} = f(x)$, siendo \hat{y} la imagen SR. Para calcular el error de la predicción los autores proponen el error cuadrático medio sobre el conjunto de entrenamiento, algo similar a lo que se propuso en SRCNN.

No obstante, VDSR utiliza un aprendizaje residual. Es decir, como las imágenes de entrada y salida se consideran similares, se define la imagen residual como $r = y - x$, donde la mayoría de los valores tienden a 0 o son muy pequeños debido a la función ReLU. De esta forma, la red es entrenada a partir de los errores residuales r y la predicción \hat{y} . La función de pérdida final propuesta por los autores se representa en la ecuación 3.1.

$$L_{vdsr} = \frac{1}{2} \|r - \hat{y}\|^2 \quad (3.1)$$

3.1.2. SRGAN

Uno de los grandes paradigmas en el campo de la súper-resolución es la introducción de las redes GAN para la generación de imágenes SR a partir del modelo del generador. En este contexto, uno de los trabajos precursores es el presentado por Christian Ledig en 2016 [83]. El método SRGAN (*Super Resolution Generative Adversarial Network*) se basa en el uso de una red generativa adversaria para la súper-resolución de imágenes. Una de las innovaciones propuestas por los autores es el uso de dos pérdidas distintas como estrategia de entrenamiento, la pérdida adversaria y la pérdida de contenido. Esto posibilita recuperar de forma eficiente los detalles de alta frecuencia de la imagen LR y obtener una imagen SR que sea perceptivamente similar a la imagen HR. Esto se consigue precisamente con el continuo enfrentamiento entre el generador y discriminador. La arquitectura propuesta en este artículo, que define tanto el generador como el discriminador, se muestra en la Figura 3.2.

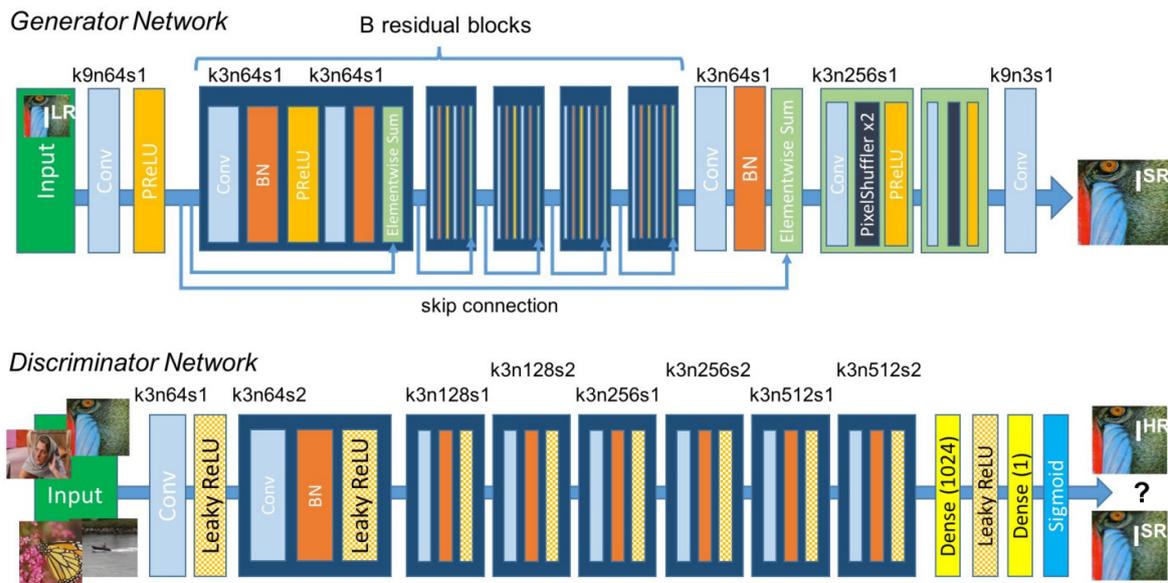


Figura 3.2: Arquitectura de SRGAN. Extraída de [83]

Donde k3n64s1 significa que se han usado 3 filtros con 64 canales y un parámetro de paso de 1. Esta nomenclatura se extiende a todos los bloques de la arquitectura.

3.1.2.1. Proceso de entrenamiento y función de pérdida

Para poder entrenar la red GAN los autores disponen de una red residual entrenada previamente a la que denominan SRResNet. En la figura anterior, SRResNet es usada para entrenar el modelo generador, implementando conexiones de salto al igual que ResNet [83]. Este fragmento de la arquitectura propuesta se compone de $B = 16$ bloques residuales apilados uno detrás de otro, compuestos de 2 capas convolucionales con 64 filtros de un tamaño de 3×3 . Como función de activación se propone una PReLU (*Parametric ReLU*), que es una variación de la ReLU clásica. Por último, el proceso de escalado de la imagen se realiza a través de 2 capas de convolución de subpíxeles.

El proceso de entrenamiento se puede resumir en tres pasos:

1. Procesado previo de las imágenes HR para transformarlas en imágenes LR.
2. Las imágenes LR se insertan al generador, que realiza un determinado muestreo generando la imagen SR.
3. El discriminador se encarga de comparar la imagen SR con la HR. El error resultante es propagado hacia atrás para entrenar tanto generador como discriminador.

Para poder diferenciar las imágenes HR reales de las imágenes SR generadas se necesita entrenar un discriminador. Con este propósito, se utiliza una función de activación LeakyReLU (*Leaky Rectified Linear Unit*), otra variante de la ReLU original. La red discriminadora contiene 8 capas convolucionales con un número creciente de filtros de tamaño 3 x 3, comenzando con 64 núcleos y llegando hasta 512. Después de las capas convolucionales se aplican 2 capas densas y una función de activación sigmoide final para obtener una probabilidad de clasificación de la muestra. De esta forma, las dos posibles respuestas permiten obtener la pérdida adversaria, ya que el discriminador ha decidido si la imagen se considera real o falsa.

Como se ha nombrado anteriormente, la función de pérdida queda definida por dos pérdidas distintas, la adversaria y la de contenido. La pérdida adversaria se aplica en el generador y es representada en la ecuación 3.2.

$$l_{\text{Gen}}^{SR} = \sum_{n=1}^N -\log D_{\theta_D} (G_{\theta_G} (I^{LR})) \quad (3.2)$$

Donde $D_{\theta_D} (G_{\theta_G} (I^{LR}))$ es la probabilidad de que la imagen reconstruida $G_{\theta_G} (I^{LR})$ sea la imagen original.

La pérdida de contenido se usa para poder mantener una cierta similitud perceptiva con respecto a la imagen original. Los autores definen la pérdida de contenido a partir del modelo VGG presentado por Karen Simonyan y Andrew Zisserman [84]. La pérdida VGG mide el MSE de las características extraídas de las imágenes y se representa a partir de la ecuación 3.3.

$$l_{\text{VGG}/i,j}^{SR} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \left(\phi_{i,j} (I^{HR})_{x,y} - \phi_{i,j} (G_{\theta_G} (I^{LR}))_{x,y} \right)^2 \quad (3.3)$$

Donde $W_{i,j}$ y $H_{i,j}$ describen las dimensiones de los respectivos mapas de características dentro de la red VGG.

3.1.3. EDSR

El método EDSR (*Enhanced Deep Super Resolution*) es propuesto en 2017 por Bee Lim y puede ser considerado como un método intermedio entre VDSR y SRGAN, debido a que fundamentalmente adapta el bloque SRResNet introducido por la segunda

pero no hace uso de redes GAN, sino de una serie de combinaciones de capas convolucionales y residuales [85]. Los autores del artículo proponen un modelo residual donde se han eliminado algunos procesos innecesarios de las redes residuales convencionales. De la misma forma, el método planteado permite la tarea de escalado único y multiescalado a través de distintas arquitecturas. En este trabajo se comenzará explicando la arquitectura que permite el escalado único, siendo esta representada en la Figura 3.3.

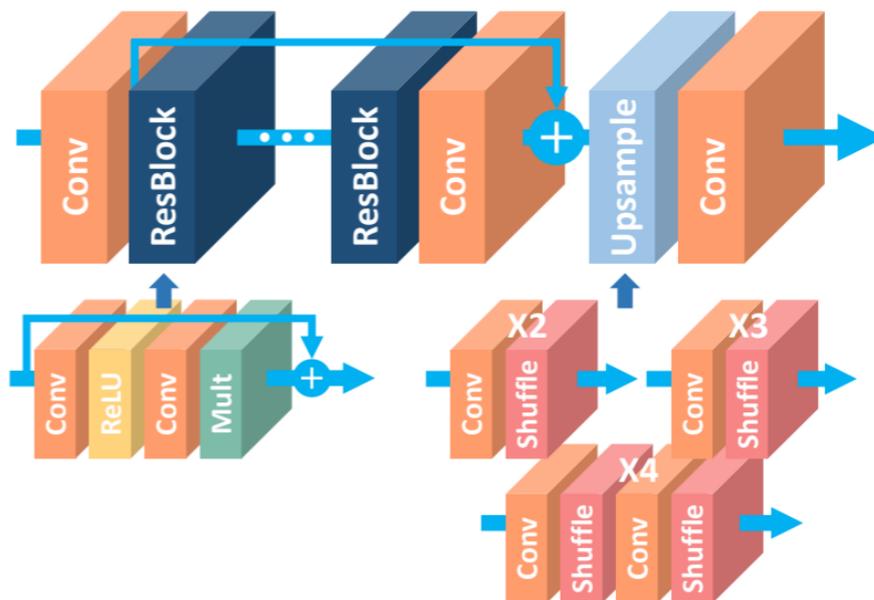


Figura 3.3: Arquitectura de EDSR. Extraída de [85]

En la figura anterior se puede observar la totalidad de bloques que conforman el modelo. En este caso, EDSR utiliza el bloque “Shuffle”, que denota una capa de convolución de subpíxeles para implementar el factor de escala y poder conseguir una imagen de súper-resolución. Más concretamente, se usa una única capa de este tipo para implementar las escalas x2 y x3 y dos capas para la escala x4.

3.1.3.1. Proceso de entrenamiento y función de pérdida

El modelo final planteado por los autores cuenta con un total de 32 capas convolucionales y residuales, extrayendo en cada una de estas un total de 256 mapas de características. Su diseño de bloques residuales difiere del planteado en ResNet ya que se han eliminado las capas de normalización de lotes, además de la función de activación final ReLU, tal como se ve en la tercera imagen de la Figura 3.4.

- La normalización de lotes es un proceso que consiste en normalizar la salida de

una capa de activación anterior, restando la media del lote y dividiéndola por la desviación estándar del lote [86].

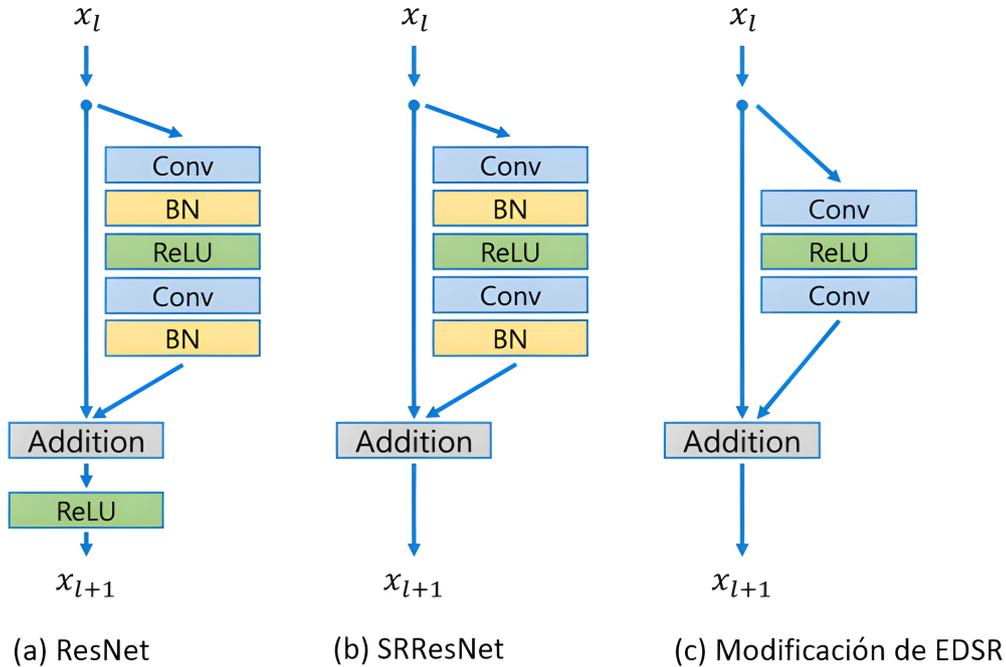


Figura 3.4: Modificación del bloque SRResNet propuesta en [85]

Para entrenar al modelo se sigue una estrategia de aprendizaje basada en la pérdida de píxeles L1. Es decir, se toma en consideración la diferencia absoluta entre el valor predicho y el valor real. Por otro lado, es necesario dividir el entrenamiento del modelo en al menos 2 pasos:

1. En primer lugar, se entrena desde cero el modelo que aplica una escala $\times 2$.
2. En segundo lugar, se utiliza este modelo pre-entrenado para entrenar los futuros modelos que aplicarán las otras escalas.

3.1.4. MDSR

El método MDSR (*Multi-scale Deep Super Resolution*) parte del mismo principio que la propuesta realizada en EDSR, ya que hace uso de un bloque residual modificado por los autores del artículo [85]. No obstante, es diseñado para poder realizar multi-escalado de imágenes. La arquitectura del modelo propuesto se presenta en la Figura 3.5.

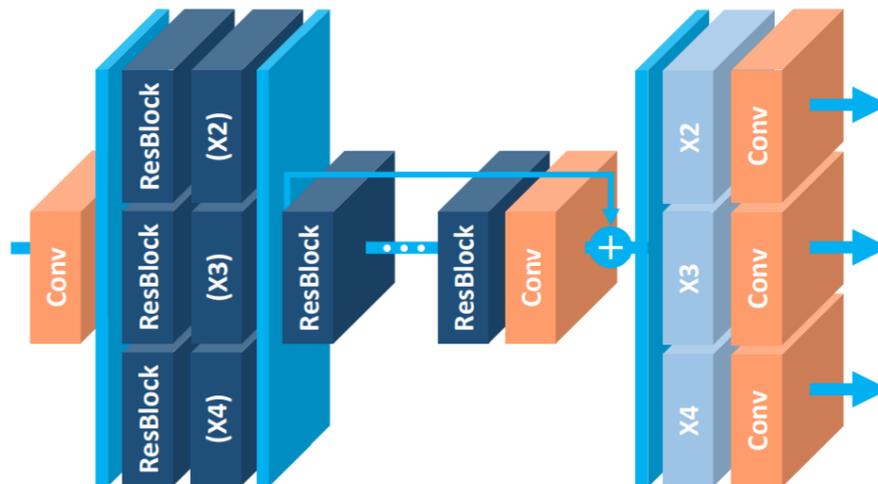


Figura 3.5: Arquitectura de MDSR. Extraída de [85]

Una de las principales diferencias respecto al EDSR es que tanto los módulos de preprocesamiento, como las capas residuales, se sitúan al principio de la red neuronal y de forma apilada, esto permite reducir la varianza de las imágenes de entrada a las que se les apliquen distintas escalas.

3.1.4.1. Proceso de entrenamiento y función de pérdida

El modelo final planteado por los autores cuenta con un total de 80 capas convolucionales y residuales, extrayendo en cada una de estas un total de 64 mapas de características. Además, el método MDSR adopta filtros más grandes para los módulos de preprocesamiento, en este caso de 5×5 .

Para entrenar al modelo se sigue una estrategia de aprendizaje basada en la pérdida de píxeles L1, al igual que se efectúa en EDSR. Desde esta perspectiva, se puede concluir que ambos modelos propuestos parten de un formato de entrenamiento similar. Sin embargo, MDSR permite que la futura evaluación de datos se realice con un sólo modelo compilado, al contrario de lo que pasa en EDSR.

3.1.5. WDSR

El método WDSR (*Wide Activation for Image Super Resolution*) es planteado por Jiahui Yu y parte de las premisas definidas por el método EDSR [87]. La arquitectura propuesta en el artículo se muestra en la Figura 3.6.

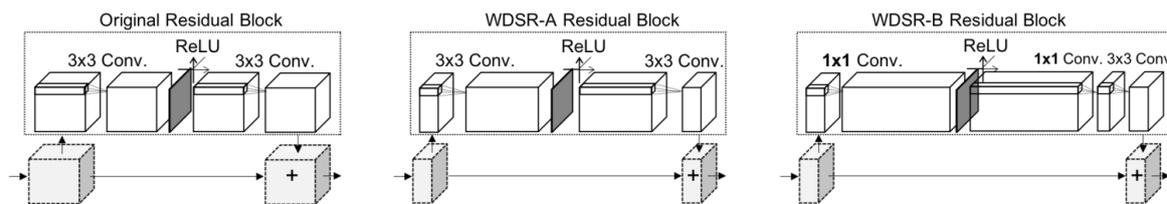


Figura 3.6: Arquitectura de WDSR. Extraída de [87]

El método WDSR realiza algunos cambios significativos en el modelo original planteado por el método EDSR. Por ejemplo, se reduce el número de canales en la ruta de mapeo de identidad y se aumenta el número de canales en cada bloque residual, esto se realiza sin aumentar la cantidad total de parámetros en la red neuronal.

- La ruta de mapeo de identidad es una operación que garantiza que la salida de una determinada red neuronal multicapa sea igual a su entrada.

De forma resumida, los autores establecen que sin parámetros ni cálculos adicionales, la simple expansión de características antes de la activación de las funciones ReLU, conduce a mejoras significativas para la súper-resolución de una sola imagen. La intuición del trabajo propuesto es que esta expansión de características permite que pase más información mientras se mantiene la alta no linealidad de las redes neuronales profundas. El método WDSR ofrece dos variantes distintas, WDSR-A y WDSR-B. La única diferencia entre estas variantes es la cantidad y la disposición de los bloques residuales que conforman la red neuronal.

3.1.5.1. Proceso de entrenamiento y función de pérdida

Para entrenar al modelo se sigue una estrategia de aprendizaje basada en la pérdida de píxeles L1, al igual que se efectúa en EDSR. No obstante, en ambos modelos se plantea la eliminación de capas convolucionales redundantes para agilizar el proceso de entrenamiento. Por un lado, cabe destacar que el modelo WDSR-A propone un aumento de las características antes de la función ReLU de cada bloque residual en un factor de x2 y x4. Por otro lado, el modelo WDSR-B plantea un aumento de entre x6 y x9 de los mapas de características de las imágenes, consiguiendo aún mayor eficiencia a cambio de un mayor esfuerzo computacional.

3.1.6. ESRGAN

El método ESRGAN (*Enhanced Super Resolution Generative Adversarial Network*) es presentado como una mejora directa del método SRGAN por parte de Xintao Wang [88]. En particular, este modelo introduce un nuevo tipo de bloque residual llamado RRDB (*Residual-in-Residual Dense Block*) como la unidad básica de construcción de la red. Además, se implementa un nuevo tipo de pérdida adversaria y un nuevo tipo de pérdida de contenido.

Más concretamente, su trabajo propone las siguientes mejoras respecto a SRGAN:

1. Mejorar la arquitectura del modelo base utilizando RRDB sin normalización de lotes. Esta mejora se basa en las observaciones realizadas por el método EDSR.
2. Utilizar la pérdida relativista RaGAN (*Relativistic GAN*) en lugar de la anterior pérdida adversaria de SRGAN.
3. Mejorar la pérdida de contenido de SRGAN extrayendo las características de la imagen antes de la capa de activación de la red VGG.

La arquitectura general de la red neuronal se presenta en la Figura 3.7.

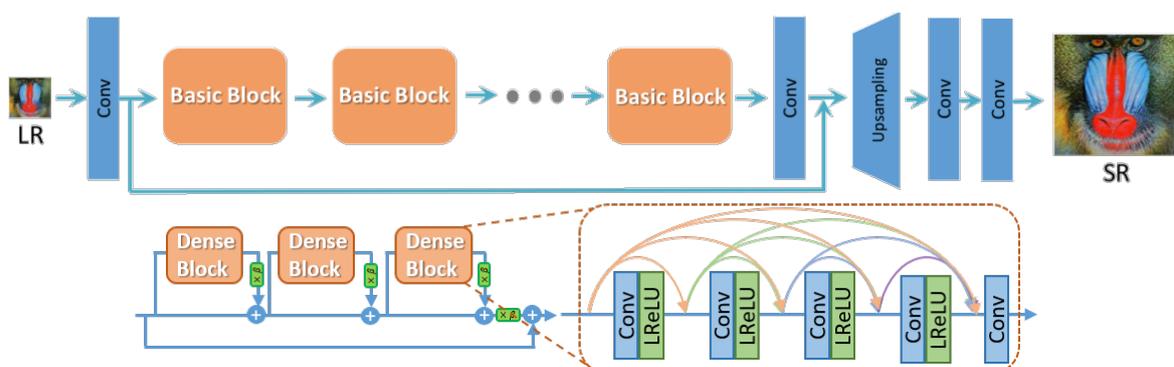


Figura 3.7: Arquitectura de ESRGAN. Extraída de [88]

Al igual que en EDSR, se opta por eliminar las capas de normalización de lotes, reduciendo la complejidad computacional y optimizando el tiempo de entrenamiento. El método ESRGAN sigue la arquitectura básica al estilo ResNet de SRGAN, pero sustituye el bloque residual SRResNet por el bloque RRDB propuesto por los autores.

3.1.6.1. Cambios en el generador y discriminador

El modelo del generador de ESRGAN es similar al generador planteado en SRGAN, sin contar la modificación del bloque residual comentada anteriormente. No obstante, el discriminador si sufre cambios significativos a la hora de predecir si una imagen es real o falsa.

Este artículo propone una “pérdida relativista” capaz de predecir, a través de un discriminador modificado, la probabilidad de que una imagen real x_r sea relativamente más realista que una imagen falsa x_f . El cambio en el discriminador puede observarse en la Figura 3.8.

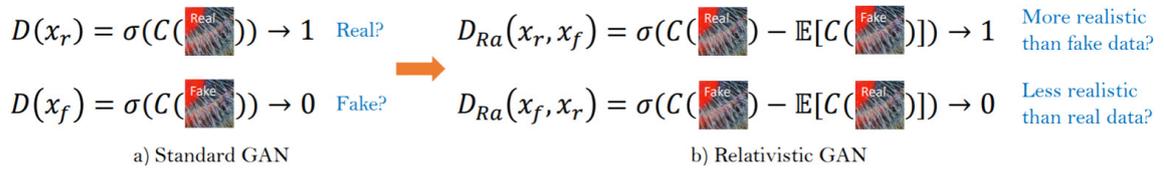


Figura 3.8: Discriminador que implementa ESRGAN. Extraída de [88]

En la imagen anterior σ representa la función sigmoide usada en SRGAN para clasificar la imagen generada como real o falsa, de esta manera $\sigma(C(x))$ representa la salida del discriminador. Por otro lado, en un discriminador relativista, $\mathbb{E}_{x_f}[\cdot]$ representa la operación de tomar la media de todos los valores falsos de un determinado conjunto de datos. Siguiendo con esta línea, las pérdidas del discriminador y del generador se formulan en la ecuación 3.4.

$$\begin{aligned}
 L_D^{Ra} &= -\mathbb{E}_{x_r} [\log (D_{Ra} (x_r, x_f))] - \mathbb{E}_{x_f} [\log (1 - D_{Ra} (x_f, x_r))] \\
 L_G^{Ra} &= -\mathbb{E}_{x_r} [\log (1 - D_{Ra} (x_r, x_f))] - \mathbb{E}_{x_f} [\log (D_{Ra} (x_f, x_r))]
 \end{aligned}
 \tag{3.4}$$

Donde x_f y x_r representan respectivamente la imagen HR y la imagen LR de entrada. Se observa que la pérdida adversaria, tanto para el generador como para el discriminador, contiene los factores x_f y x_r .

3.1.6.2. Pérdida VGG modificada

Los autores de ESRGAN, al igual que se hizo en SRGAN, proponen una pérdida de contenido para poder generar una imagen SR relativamente realista. La pérdida

de contenido se define previamente en las capas de activación de una red profunda pre-entrenada, donde se minimiza la distancia entre dos características activadas. La innovación de los autores de ESRGAN se basa en utilizar las características generadas por la red VGG antes de la capa de activación ReLU, lo que permitirá superar un inconveniente del modelo original. Este inconveniente se relaciona con la escasez de características activadas a partir de la cuarta capa convolucional.

3.1.6.3. Proceso de entrenamiento y función de pérdida total

El proceso de entrenamiento, al igual que en toda red GAN, se basa en el continuo enfrentamiento del generador y discriminador diseñado. Durante este proceso los autores del artículo original usaron un total de 23 bloques RRDB. La pérdida total que se pondera durante el entrenamiento del modelo se define según la ecuación 3.5.

$$L_G = L_{\text{percep}} + \lambda L_G^{Ra} + \eta L_1 \quad (3.5)$$

Donde L_{percep} y λL_G^{Ra} son la pérdida de contenido obtenida de la red VGG y la pérdida adversaria respectivamente. Mientras que ηL_1 es la pérdida de píxel L1 que evalúa la diferencia entre la imagen generada y la imagen original, siendo η un coeficiente propuesto por los autores para equilibrar los diferentes términos de pérdida.

3.1.7. Real-ESRGAN

El método Real-ESRGAN es la evolución directa del método ESRGAN comentado anteriormente, siendo este propuesto por Xintao Wang [89]. El objetivo de este nuevo método es intentar restaurar una imagen LR que ha sido expuesta a degradaciones desconocidas y complejas. Es decir, intentar abordar las imágenes degradadas que se encuentran en el mundo real, donde no se sabe con exactitud a que tipo de situaciones se han visto expuestas. Con esta intención, los autores entrenan una red neuronal con datos sintéticos puros, siendo estos datos una serie de imágenes sometidas a un proceso de degradación particular para así simular distintos escenarios del mundo real.

Este proceso de degradación se muestra en la Figura 3.9. Como se puede observar, las imágenes son degradadas en dos ocasiones por procesos de primer y segundo orden

que incumben operaciones como el desenfoco, la reducción de resolución, la adición de ruido y la compresión JPEG (*Joint Photographic Experts Group*).

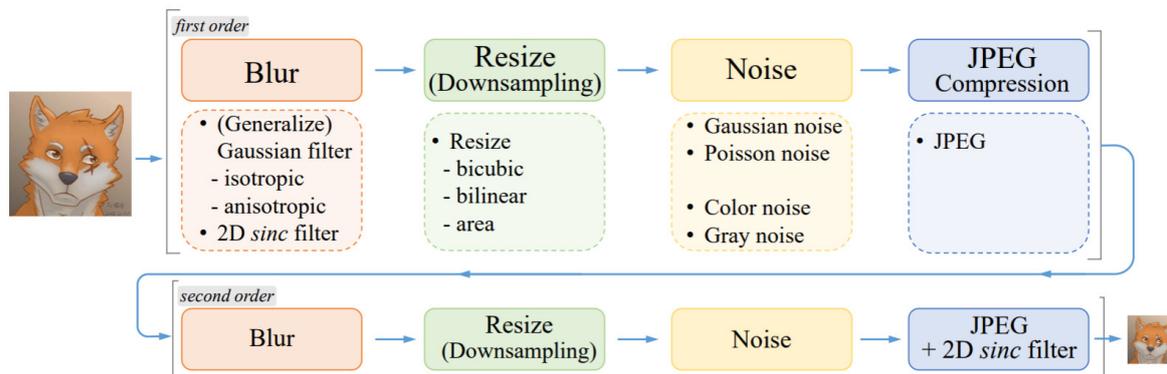


Figura 3.9: Degradación de la imagen llevada a cabo en el método Real-ESRGAN [89]

La arquitectura de Real-ESRGAN se muestra en la Figura 3.10, en esta se ilustra la implementación de la arquitectura ESRGAN propuesta en el método anterior.

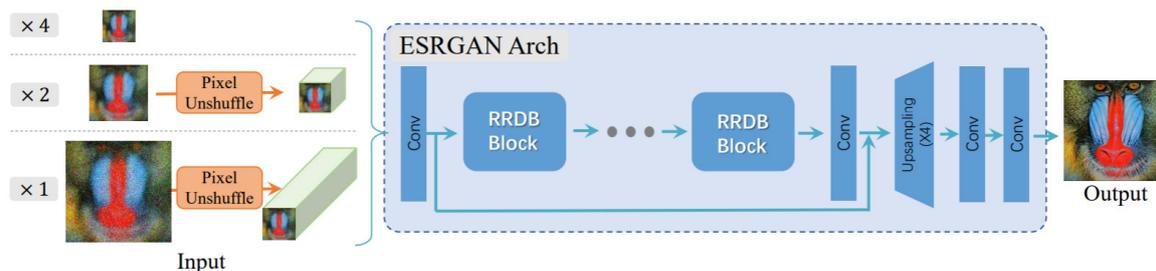


Figura 3.10: Arquitectura de Real-ESRGAN. Extraída de [89]

Como se puede observar en la imagen anterior, Real-ESRGAN implementa una arquitectura similar a ESRGAN. No obstante, existen ciertas novedades tanto en la parte del generador como en la parte del discriminador:

- El nuevo generador utiliza una capa de deconvolución de subpíxeles, una operación inversa a la capa de subpíxeles. Esto se realiza con la intención de reducir el tamaño espacial y reordenar la información antes de que esta entre en la arquitectura principal del modelo ESRGAN.
- Por otro lado, la arquitectura del discriminador sufre un cambio. En este caso se opta por una arquitectura de tipo U-NET. Una arquitectura U-NET, como su nombre indica, no es más que la disposición de capas convolucionales y deconvolucionales de forma inversa, componiendo la forma de una U. Un ejemplo de

arquitectura U-NET se muestra en la Figura 3.11. A esto se le añaden conexiones de salto para evitar posibles cuellos de botella en las zonas donde se trabaje con pocos píxeles.

- El nuevo modelo propuesto, al contrario que ESRGAN, es ampliado para efectuar súper-resolución con escala x2 y x4.

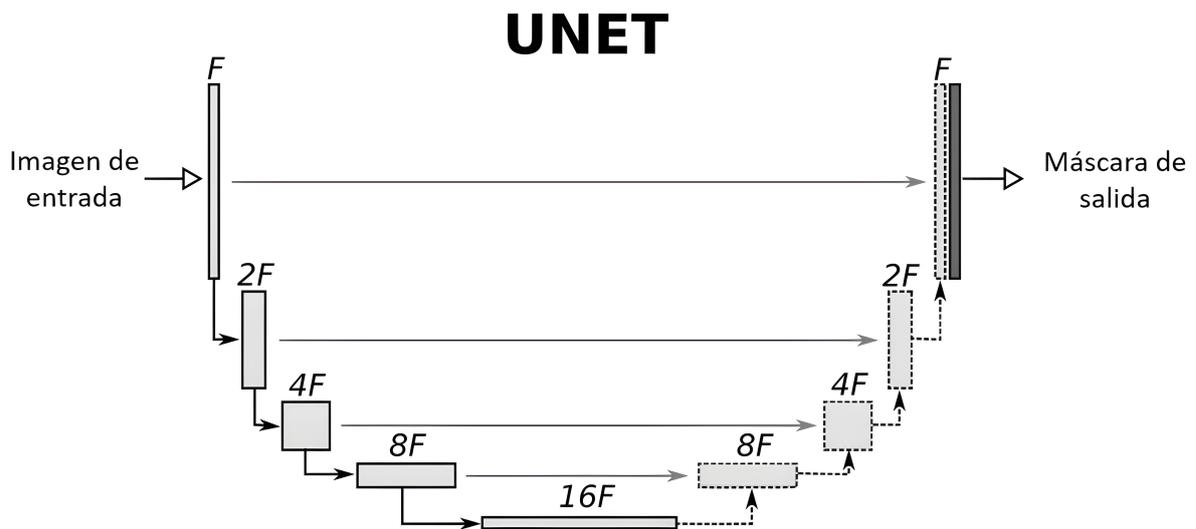


Figura 3.11: Diseño genérico de una U-NET. Adaptada de [90]

3.1.7.1. Proceso de entrenamiento y función de pérdida

Los autores proponen un proceso de entrenamiento distinto a ESRGAN y que se puede dividir en dos etapas:

- En la primera etapa se entrena un modelo con la pérdida L1. El modelo obtenido se denomina Real-ESRNet.
- En la segunda etapa se usa el modelo Real-ESRNet como inicialización del generador y se vuelve a entrenar el modelo con una combinación de pérdida L1, pérdida de contenido y pérdida adversaria, como se realiza en el modelo ESRGAN.

Finalmente, cabe destacar que al reutilizar la arquitectura anterior, se vuelven a usar un total de 23 bloques RRDB.

Capítulo 4

Bases de datos

El estado de la técnica en métodos de súper-resolución espacial ha sido un tema muy estudiado a lo largo de los años, debido a la necesidad de mejorar la calidad de una imagen que ha sido degradada por el ruido, el desenfoque o simplemente porque cuenta con una tasa de píxeles insuficiente para extraer características de esta. Sin embargo, en la termografía, al contrario de lo que puede suceder en el campo de la fotografía clásica, un valor de píxel representa una relación real con la temperatura del objeto estudiado. De esta manera, es necesario establecer una diferencia entre lo que se conoce como súper-resolución espacial, mejorar la calidad visual de la imagen, y súper-resolución a nivel del valor radiométrico que representa cada píxel, que es mejorar o aumentar la precisión de las lecturas.

Por este hecho, es imprescindible asegurar dos puntos:

1. Que los métodos de súper-resolución espacial estudiados no degraden en exceso el valor radiométrico de los píxeles de la imagen.
2. Que el valor radiométrico y la temperatura equivalente del píxel sea cercano a la realidad estudiada.

En este sentido, se considera necesario estudiar lo que en este trabajo se denomina súper-resolución radiométrica. Esto significa mejorar las lecturas obtenidas de los píxeles de la imagen para que así sean cercanas a la temperatura real del objeto estudiado, teniendo en cuenta que la incertidumbre de estos datos aumenta cuanto mayor es la distancia entre el objeto y la cámara. Para conseguir los objetivos propuestos, ya

se trate de súper-resolución espacial o radiométrica, es necesario contar con una gran cantidad de datos a la hora de implementar y probar un modelo basado en aprendizaje profundo. Por este motivo, se plantea el uso de cuatro bases de datos distintas, dos de ellas accesibles a través de internet y otras dos generadas en este trabajo.

4.1. Bases de datos para la súper-resolución espacial

Esta sección abarca un total de tres bases de datos distintas, dos de ellas dedicadas al entrenamiento de algunos métodos de súper-resolución espacial y otra dedicada completamente a la evaluación de estos métodos.

4.1.1. Bases de datos para el entrenamiento de los métodos

Las bases de datos usadas en el futuro entrenamiento de los métodos de súper-resolución se nombran en los siguientes puntos:

- Por un lado, se ha usado la base de datos DIV2K [91]. Esta cuenta con 1000 imágenes RGB de resolución 2K (2048 x 1080) divididas en 800 imágenes para el entrenamiento, 100 imágenes para la validación y 100 imágenes para la realización de distintas pruebas.
- Por otro lado, se ha usado la base de datos ADAS de FLIR [92]. Esta base de datos de imágenes ha sido recabada para tareas como la detección de vehículos y personas en las calles y carreteras, contando con un total de 10228 imágenes divididas en 8862 imágenes de entrenamiento y 1366 de validación. Todas las imágenes están disponibles en formato RGB (1800 x 1200), formato termográfico de 8 bits (640 x 512) y formato termográfico de 16 bits (640 x 512).

4.1.2. Base de datos para la evaluación de los métodos

Una pequeña base de datos de imágenes termográficas ha sido realizada con la intención de evaluar el rendimiento de los métodos de súper-resolución espacial, tanto los basados en reconstrucción como los basados en redes neuronales. Para esta base de datos propia se han usado tres modelos del fabricante FLIR, A615, A35 y Lepton. Estas cámaras son de distintas gamas de precios, siendo la primera de unas prestaciones

superiores a las otras dos, con una resolución de 640 x 480 y una óptica con enfoque regulable. En la Figura 4.1 se pueden observar las cámaras usadas para la generación de la primera base de datos creada en este trabajo. Por otro lado, las prestaciones de estas cámaras se muestran en la Tabla 4.1.



Figura 4.1: Cámaras usadas para la creación de la primera base de datos

El emplazamiento escogido para esta toma de datos ha sido el edificio Polivalente II del parque científico tecnológico de la ULPGC (Universidad de Las Palmas de Gran Canaria), más concretamente en el IDeTIC (Instituto para el Desarrollo Tecnológico y la Innovación en Comunicaciones), donde se realizaron la mayoría de los experimentos de este trabajo. La base de datos cuenta con un total de 160 imágenes termográficas de distintos escenarios, ya que se tomaron imágenes tanto del interior de los laboratorios como del exterior del edificio.

Camaras	Características físicas			
	Resolución	Distancia focal (mm)	FOV _H (°)	FOV _V (°)
FLIR A615	640 x 480	24.6	25	18.8
FLIR A35	320 x 256	19	24	19.2
FLIR Lepton	160 x 120	2	57	42

Tabla 4.1: Características de las cámaras mostradas en la Figura 4.1

La base de datos para la evaluación de los métodos de súper-resolución espacial se denotará a lo largo de este trabajo como DASSR (*Database for the Assessment of Spatial Super-Resolution*). Para tener un control básico del procesado y evaluación que se realizará en capítulos posteriores, es necesario segmentar las imágenes según la localización del experimento. Siguiendo esta misma nomenclatura, la segmentación de esta base de datos será DASSR-I para las imágenes 1-40, DASSR-II para las imágenes 41-100, DASSR-III para las imágenes 101-130 y DASSR-IV para las imágenes 131-160.

Algunas de estas imágenes se ilustran en la Figura 4.2, donde se pueden observar tanto ejemplos en formato RGB como en formato termográfico de 8 bits con escala de grises.

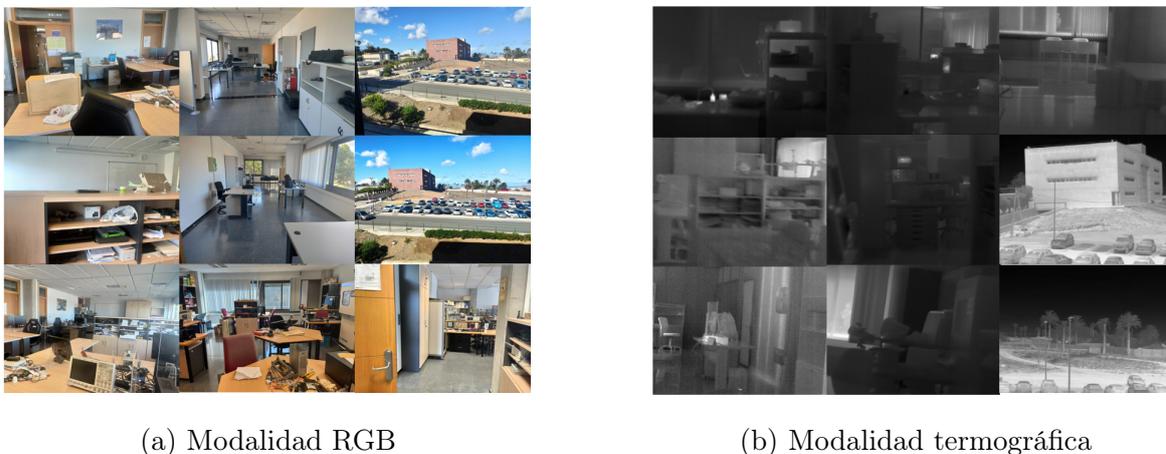


Figura 4.2: Muestras extraídas de la base de datos DASSR

4.2. Base de datos para la súper-resolución radiométrica

Para llevar a cabo la premisa propuesta sobre la súper-resolución radiométrica, una gran cantidad de datos necesitan ser recogidos por parte de varias cámaras termográficas de distintas gamas. De la misma manera, un amplio rango de temperaturas ha de ser estudiado para asegurar el correcto comportamiento del modelo diseñado y entrenado.

4.2.1. Base de datos para el entrenamiento y evaluación

El escenario de medidas propuesto es el corredor del IDeTIC, que cuenta con 32 metros de pasillo para poder obtener imágenes de entrenamiento y evaluación con los que alimentar la red neuronal. Para poder tomar muestras es necesario contar con una superficie radiante adecuada, que sea capaz de ofrecer unos amplios rangos de temperatura con los que poder modelar bien la mayor cantidad de casos posibles. En este acercamiento se ha decidido utilizar como fuente radiante una vitrocerámica portátil, el modelo PCE-5000 del fabricante Orbegozo.

4.2.1.1. Calibración de la superficie radiante

A la hora de proceder con la calibración de temperaturas que ofrece la vitrocerámica en entornos de interiores controlados, es necesario contar con toda una serie de medidas de protección, ya que esta alcanza temperaturas mayores de 400°C. El modelo de vitrocerámica escogido es regulable por un potenciómetro, pero este no marca la temperatura de la superficie, por lo que es imprescindible tomar la referencia de temperatura a través de medios externos. En este caso, los medios para contrastar la temperatura de la superficie serán las cámaras termográficas, un termómetro infrarrojo y una sonda de temperatura. La Figura 4.3 muestra las cámaras y la superficie radiante usada para recabar los datos.



Figura 4.3: Sistema de cámaras y superficie radiante propuesta

Como puede observarse en la figura anterior, estos experimentos añaden una cámara termográfica más, la Tau 2 del fabricante FLIR. Por este motivo, conviene actualizar la Tabla 4.1 mostrada anteriormente. La Tabla 4.2 muestra las características de todas las cámaras usadas en la campaña de medidas de esta sección.

Camaras	Características físicas			
	Resolución	Distancia focal (mm)	FOV _H (°)	FOV _V (°)
FLIR A615	640 x 480	24.6	25	18.8
FLIR A35	320 x 256	19	24	19.2
FLIR Tau 2	336 x 256	35	9.3	7.1
FLIR Lepton	160 x 120	2	57	42

Tabla 4.2: Características de las cámaras mostradas en la Figura 4.3

Debido a que las cámaras pueden presentar errores en la lectura de temperaturas por varias circunstancias, es de necesidad contar con instrumentos externos capaces de tomar referencias correctas. Un ejemplo que ilustra las referencias de temperaturas obtenidas se observa en la Figura 4.4.



Figura 4.4: Validación de temperaturas con la sonda Fluke 51 II

El experimento para la calibración de la superficie radiante se desarrolló en un margen temporal ajustado de 175 minutos. Durante el desarrollo de este experimento se situó la vitrocerámica a 1 metro de distancia respecto al sistema de cámaras usado. Las capturas fueron tomadas cada 3 segundos hasta llegar a un número de 500 capturas

por cámara, regulando manualmente el potenciómetro de la vitrocerámica para ajustar la temperatura. La distribución del experimento se observa en la Figura 4.5, donde además es visible el uso de un extintor para casos de emergencia y un termómetro infrarrojo como medio alternativo a la toma de referencias de la superficie radiante.



Figura 4.5: Experimento realizado para la calibración de la vitrocerámica

El protocolo de medidas seguido para el experimento se desglosa en el Protocolo 1.

Protocolo 1: Calibración de la superficie radiante

1. Situar la superficie emisora a 1 metro de distancia de las cámaras.
2. Ajustar de forma manual el trípode para enfocar bien la superficie emisora.
3. Ajustar la distancia focal de las cámaras.
4. Ajustar el potenciómetro de la vitrocerámica.
5. Esperar a que la temperatura de la superficie emisora se estabilice.
6. Ejecutar las rutinas programadas para obtener 500 capturas de cada cámara con un intervalo de 3 segundos entre capturas y almacenar los datos en carpetas separadas.
7. Comprobar que los datos se han almacenado de forma correcta.
8. Se repiten los pasos anteriores a partir del punto 4 (incluido).

Los resultados de un procesado posterior se pueden resumir en la Figura 4.6. Esta figura muestra los rangos de temperatura media que ofrece la vitrocerámica, segmentados según la posición del potenciómetro.

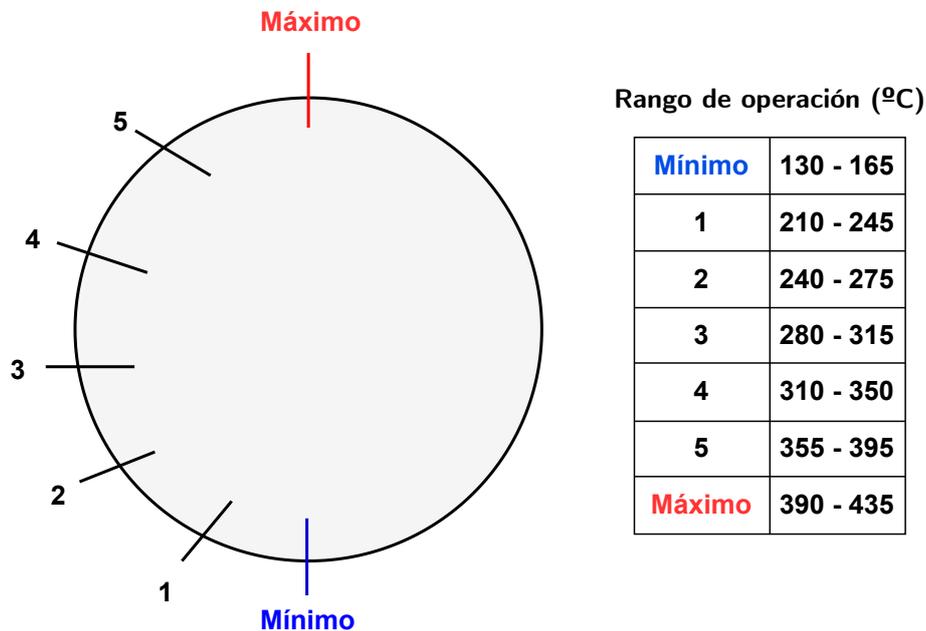


Figura 4.6: Rangos de temperatura de la vitrocerámica PCE-5000

Una vez parametrizados los rangos de temperatura que ofrece la vitrocerámica se pasa a la fase de creación de la base de datos.

4.2.1.2. Obtención de muestras para el entrenamiento y la evaluación

El experimento consiste en la toma de datos radiométricos por parte de las 4 cámaras comentadas anteriormente. En este acercamiento se necesita ocupar el corredor del IDeTIC para poder cambiar la posición de la superficie emisora, mientras que las cámaras estarán posicionadas de forma fija cerca del operario. El principal objetivo de este experimento es obtener una gran base de datos radiométricos de cámaras termográficas con distintas resoluciones y ópticas.

El experimento se desarrolló en un margen temporal de 10 días, ya que se realizaron medidas para todos los modos de temperatura de la superficie emisora y para una gran cantidad de distancias distintas. Un ejemplo de la distribución del experimento se observa en la Figura 4.7.



Figura 4.7: Procedimiento para la creación de la base de datos

Para conseguir el objetivo propuesto de crear una base de datos propia es necesario volver a realizar un protocolo de medidas específico, este queda referenciado en el Protocolo 2. Las muestras obtenidas de este experimento serán las que se destinarán al entrenamiento de la red neuronal, añadiendo también las muestras obtenidas durante la calibración de la vitrocerámica, ya que ofrecen información de interés para distancias cercanas.

Con la intención de evitar una futura sobrecarga de datos y posibles comportamientos sesgados durante el proceso de entrenamiento, se debe contar con muestras suficientes para una correcta evaluación de la red neuronal. Con estos valores se podrá comprobar su rendimiento ante situaciones no contempladas durante el entrenamiento. De igual forma, para mantener una metodología común al resto de experimentos, se decide hacer uso de la misma superficie emisora para obtener las muestras de evaluación. El objetivo final de las muestras de evaluación es conseguir los primeros resultados empíricos del prototipo de red neuronal que se ha diseñado, teniendo en cuenta unas distancias concretas y temperaturas distintas. Por este motivo, las muestras de evaluación fueron obtenidas en un marco temporal de 3 días, incluyendo distancias que van de 2 a 18 metros para temperaturas mayores de 300 °C y distancias de 3 a 31 metros para distintos rangos de temperaturas entre 50 °C y 400 °C.

Protocolo 2: Captación de muestras para el entrenamiento

1. Situar la superficie emisora a 4 metros de distancia de las cámaras.
2. Ajustar de forma manual el trípode para enfocar bien la superficie emisora.
3. Ajustar la distancia focal de las cámaras.
4. Ajustar la superficie emisora a una temperatura conocida.
5. Esperar a que la temperatura de la superficie emisora se estabilice.
6. Ejecutar las rutinas programadas para obtener 550 capturas de cada cámara con un intervalo de 3 segundos entre capturas y almacenar los datos en carpetas separadas.
7. Comprobar que los datos se han almacenado de forma correcta.
8. Incrementar en 4 metros la distancia entre la superficie emisora y las cámaras, repitiendo los pasos desde el punto 6 (incluido) hasta alcanzar los 32 metros.
9. Cuando se termine con la iteración se aumentará la temperatura a través del potenciómetro, comenzando otra nueva iteración desde el punto 1 y siguiendo los puntos anteriormente nombrados.

La base de datos conseguida tras el procedimiento de medida queda conformada por un total de 77725 imágenes por cada una de las cámaras termográficas utilizadas. A partir de este punto se denotará a esta base de datos como TID (*Thermal IDeTIC Database*). Para las muestras dedicadas al entrenamiento de la red neuronal se usará la nomenclatura TID-I, mientras que para las muestras dedicadas a su evaluación se usará la nomenclatura TID-II.

4.3. Resumen de las bases de datos propuestas

En este trabajo se proponen 4 bases de datos para el entrenamiento y la evaluación de distintos métodos de súper-resolución. Por un lado, se usarán las bases de datos DIV2K y ADAS para el entrenamiento de los métodos de súper-resolución espacial tratados en el siguiente capítulo y la base de datos DASSR para la evaluación de su desempeño, que será referenciada en el capítulo de resultados. Por otro lado, se ha creado una base de datos de imágenes termográficas denominada TID, con la intención de realizar el entrenamiento y la evaluación de una red neuronal diseñada para efectuar la súper-resolución radiométrica. Esta red neuronal se mostrará en el siguiente capítulo, mientras que su desempeño se presentará en el capítulo de resultados.

Como resumen, todas las bases de datos usadas en este trabajo se muestran en la Tabla 4.3.

	Súper-resolución espacial	Súper-resolución radiométrica
Entrenamiento	DIV2K, ADAS	TID-I
Evaluación	DASSR-I, DASSR-II, DASSR-III, DASSR-IV	TID-II

Tabla 4.3: Bases de datos usadas en el TFM

Capítulo 5

Experimentación

En este capítulo se presenta la metodología implementada para el entrenamiento de los métodos de súper-resolución espacial basados en redes neuronales, así como el diseño y entrenamiento de la red neuronal propuesta para la súper-resolución radiométrica. De la misma manera, se enunciarán los distintos experimentos llevados a cabo para referenciar el desempeño de los métodos de súper-resolución estudiados. Por un lado, se usarán métricas como la PSNR y el SSIM después de aplicar los métodos de súper-resolución espacial en la base de datos DASSR, que cuenta con 160 imágenes termográficas. Por otro lado, sobre esta misma base de datos, se comprobarán las distintas pérdidas que sufren los píxeles de una imagen termográfica al aplicar este tipo de métodos. Respecto a la súper-resolución radiométrica, se realizarán experimentos con la intención de evaluar la red neuronal a través de la base de datos TID-II, que cuenta con 14025 imágenes termográficas.

A continuación, se explicará el entrenamiento llevado a cabo para implementar la súper-resolución espacial.

5.1. Propuesta para la súper-resolución espacial

Durante el transcurso de este trabajo, una de las tareas más importantes ha sido la implementación de distintos métodos capaces de aplicar súper-resolución espacial a imágenes termográficas de 16 bits. En el caso de los métodos de reconstrucción nombrados en el capítulo 2, al no tratarse de métodos basados en aprendizaje, sólo

se ha necesitado de una serie de rutinas implementadas en el entorno MATLAB y un procesado de imagen determinado para poder conseguir el objetivo propuesto. No obstante, esto no se puede extrapolar a los métodos basados en redes neuronales, ya que se necesita de un proceso de entrenamiento previo.

Para el planteamiento de los futuros experimentos, se decide hacer uso de gran parte de los métodos de súper-resolución por reconstrucción comentados en el capítulo 2, ya que se encuentran actualmente implementados y son de libre acceso, además de todos los métodos de súper-resolución basados en redes neuronales tratados en el capítulo 3, teniendo en cuenta que su implementación se basa en bibliotecas de Python como TensorFlow, Keras y Pytorch. A la hora de elegir una implementación concreta de un método de súper-resolución espacial basado en redes neuronales, es de interés investigar si esta implementación cuenta con modelos pre-entrenados, algo que reducirá el tiempo de desarrollo de una aplicación. Sin embargo, en el caso de requerir de un entrenamiento, se han propuesto las dos bases de datos de libre acceso descritas en el capítulo anterior, una con imágenes RGB y otra con imágenes termográficas.

En los siguientes puntos se especifica la implementación elegida para cada uno de los métodos con los que se trabajará a partir de ahora.

5.1.1. Métodos basados en reconstrucción

Los métodos de interpolación, IBP, Papoulis-Gerchberg, POCS y súper-resolución robusta han sido implementados en MATLAB por otros usuarios y son accesibles a través de GitHub [93]. La implementación conjunta de todos estos métodos facilita el desarrollo de este trabajo, ya que se tiene acceso de forma ordenada a las distintas funciones de reconstrucción y registro de imágenes. Por otro lado, esta biblioteca aporta una serie de funciones que permiten, entre otras utilidades, generar imágenes LR rotadas y desplazadas a partir de una imagen HR.

5.1.2. Métodos basados en redes neuronales

La búsqueda de métodos basados en redes neuronales capaces de aplicar súper-resolución espacial a una serie de imágenes ha sido más extensa. Esto es debido fun-

damentalmente a que existen distintas bibliotecas software, la mayoría en lenguaje Python, capaces de ayudar a un usuario en el diseño y prototipado de una red neuronal de forma sencilla y con ciertos niveles de abstracción.

En este aspecto, las bibliotecas más usadas son:

- **TensorFlow.** Se trata de una biblioteca de código abierto extremo a extremo para aprendizaje automático desarrollada por Google. En la actualidad, aunque el lenguaje predominante de uso es Python, existen implementaciones en JavaScript y Android, entre otros.
- **Keras.** Se puede ver como una abstracción de más alto nivel de TensorFlow. Ha sido desarrollada por Google para facilitar la implementación cómoda y rápida de redes neuronales.
- **Pytorch.** Es una biblioteca de aprendizaje automático de código abierto para Python, desarrollada principalmente por Facebook. Junto a TensorFlow y Keras se ha convertido en una de las bibliotecas más usadas en el campo del aprendizaje profundo para el prototipado de redes neuronales.

5.1.2.1. Implementaciones elegidas

A continuación, se nombran las implementaciones usadas en este TFM, remarcando si ha sido necesario un proceso de entrenamiento y, de ser así, que bases de datos se han usado.

- **VDSR.** El método VDSR comentado en el capítulo anterior ha sido implementado en varios repositorios de la plataforma GitHub, el que se ha usado en este trabajo es accesible a través de [94]. Esta aproximación al artículo de Jiwon Kim ha sido implementada con la biblioteca Pytorch, al contrario que la implementación original en MATLAB [82]. En este repositorio se propone una serie de rutinas para realizar el proceso de entrenamiento y evaluación, utilizando una base de datos llamada TB291, que puede descargarse desde unos enlaces facilitados. De esta manera, el autor ofrece tres modelos pre-entrenados con unos factores de escala x2, x3 y x4, que serán los utilizados en este trabajo.

- **SRGAN.** El método SRGAN comentado con anterioridad ha sido implementado en la plataforma GitHub, siendo accesible a través de [95]. Esta aproximación al artículo de Christian Ledig ha sido implementada con la biblioteca Pytorch y, al contrario que en lo denotado en el artículo original, esta implementación ofrece la posibilidad de aplicar un factor de escala x2 y x4, en lugar de solamente x4 [83].

- El autor en este caso no ofrece modelos pre-entrenados de la red neuronal, pero plantea de forma guiada todo lo necesario para realizar el entrenamiento y evaluación de esta, así como las bases de datos que se han usado para entrenar el modelo. No obstante, tal como se ha comentado al principio, en esta implementación se usarán las bases de datos DIV2K y ADAS. Las características del entrenamiento llevado a cabo en este trabajo se muestran en la Tabla 5.1.

	DIV2K		ADAS	
Número de épocas totales	500		100	
Escala	x2	x4	x2	x4
PSNR (dB)	29.69	25.41	32.83	29.05
	324	477	94	95
	Época de máxima PSNR			

Tabla 5.1: Entrenamiento y evaluación del método SRGAN

- En la tabla anterior se puede observar el número de épocas realizadas durante el entrenamiento y la PSNR resultante de evaluar una serie de imágenes SR con sus correspondientes HR. El modelo obtenido del entrenamiento es elegido a partir de la época donde se ha conseguido maximizar la PSNR.

- **EDSR y WDSR.** Los métodos EDSR y WDSR tratados en el capítulo anterior han sido implementados conjuntamente y son accesibles a través de [96]. En esta aproximación a los artículos presentados por Bee Lim [85] y Jiahui Yu [87] se ha usado la biblioteca TensorFlow. Más concretamente, el autor implementa EDSR y la modalidad WDSR-B para los factores de escala x2 y x4. En este caso, se utiliza la base de datos DIV2K para el entrenamiento y la validación, ofreciendo

al usuario modelos pre-entrenados capaces de aplicar factor de escala x4. De la misma manera, se ofrecen las rutinas necesarias para entrenar y validar la red neuronal con un factor de escala x2.

- Para entrenar y evaluar la red neuronal de forma propia se utilizan las mismas bases de datos usadas en SRGAN. Las características del entrenamiento llevado a cabo en este trabajo se muestran en la Tabla 5.2.

	EDSR				WDSR-B			
	DIV2K		ADAS		DIV2K		ADAS	
Épocas	38000	---	57000	280000	41000	---	72000	257000
Escala	x2	x4	x2	x4	x2	x4	x2	x4
PSNR (dB)	33.35	28.89	33.6	28.54	34.23	28.91	32.69	28.73

Tabla 5.2: Entrenamiento y evaluación de los métodos EDSR y WDSR-B

- El proceso de entrenamiento de estas dos redes neuronales es distinto al visto en SRGAN. El autor implementa unas rutinas que almacenan automáticamente el valor de la PSNR resultante entre una imagen SR y HR sólo si esta ha mejorado respecto a la época anterior. Por ello, cada vez que el proceso de entrenamiento se reinicia, el punto de guardado corresponde con la época donde se consiguió esta PSNR.
- **MDSR.** El método MDSR, una variante propuesta a partir de EDSR para aplicar súper-resolución multiescala, ha sido implementado en GitHub y es accesible a través de [97]. Esta aproximación al artículo presentado por Bee Lim [85] ha sido implementada con la biblioteca TensorFlow, ofreciendo al usuario un modelo pre-entrenado y evaluado con la base de datos DIV2K que será usado en este trabajo.
- **ESRGAN y Real-ESRGAN.** Las evoluciones propuestas para el método SRGAN, presentadas en el capítulo anterior, han sido implementadas por los propios autores de los artículos con la biblioteca Pytorch, siendo accesibles a través de [98]. El repositorio cuenta con una implementación completa de ambos métodos, aportando distintas rutinas para que el usuario entrene y evalúe la red generativa adversaria que se propone. Sin embargo, se cuentan con modelos pre-entrenados

para ambas redes neuronales a través de bases de datos como DIV2K y Flickr2k, esta última propuesta durante el desarrollo del artículo del método EDSR [85]. Se debe tener en cuenta que ESRGAN proporciona un factor de escala $\times 4$, mientras que Real-ESRGAN proporciona factores de escala $\times 2$ y $\times 4$ [88], [89].

5.1.3. Consideraciones adicionales

Aunque todos los métodos nombrados anteriormente han sido validados para una determinada cantidad de imágenes, se debe denotar que la aplicación final buscada es la de mejorar la resolución espacial de una imagen termográfica de baja resolución. Esta es una situación que no se considera muchas veces en el momento de concepción y diseño del modelo, ya que estos suelen definirse para aplicar súper-resolución espacial a imágenes RGB. Por este motivo, además del proceso de evaluación realizado durante el entrenamiento de las redes neuronales, se considera necesario una fase propia de evaluación.

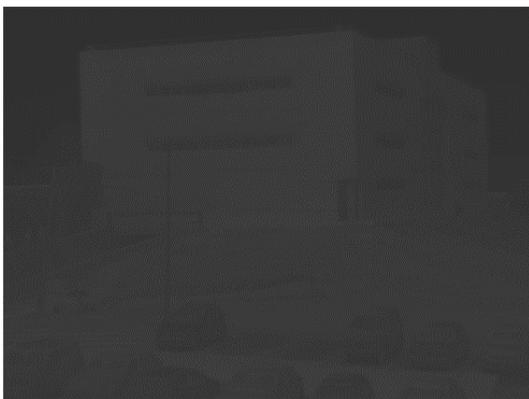
En este aspecto, los dos siguientes puntos muestran algunas consideraciones que se han de tener en cuenta:

1. Los métodos de súper-resolución espacial anteriormente nombrados han sido implementados para distintas versiones de Python y con distintas bibliotecas de aprendizaje automático. Por este motivo, es necesario alojar cada uno de los repositorios en su propio entorno virtual, activando este cuando sea necesario.
2. Es de necesidad contar con una pequeña base de datos de validación, teniendo en cuenta que se tratarán de imágenes termográficas, tanto para la prueba de los métodos basados en redes neuronales como para los métodos de reconstrucción tratados desde MATLAB. Por esta razón, se hace uso de la base de datos DASSR nombrada en el capítulo anterior.

5.2. Propuesta para la súper-resolución radiométrica

A lo largo del tiempo se ha estudiado el problema de la súper-resolución espacial, desde su concepción hasta la manera de mejorar los métodos existentes y resolver problemas tradicionales de esta técnica. Sin embargo, el uso de estos métodos en imágenes infrarrojas no ha sido tan estudiado como para imágenes RGB, aunque si existen algunos acercamientos como los vistos en [28], [30] y [31]. Dónde este último establece una comparativa interesante de distintas imágenes termográficas a las que se les ha aplicado distintos métodos de súper-resolución espacial, estudiando las pérdidas de estos y sus valores de temperatura. No obstante, sin contar este artículo, el concepto de imagen termográfica que se suele tratar en el estado de la técnica no se relaciona con el valor radiométrico binarizado que ofrece este tipo de dispositivos, más bien trata su normalización a 8 bits de datos para que sea más fácil de implementar en una red neuronal, ya que se disminuye el rango dinámico de la escena observada.

Con la intención de ofrecer un contexto adecuado, una imagen de 16 bits frente a una imagen de 8 bits se presenta en la Figura 5.1, pudiéndose observar una diferencia visual perceptible debido al amplio rango dinámico de la primera.



(a) Imagen termográfica de 16 bits



(b) Imagen termográfica de 8 bits

Figura 5.1: Comparativa entre imágenes termográficas de 16 bits y 8 bits

Debido a esto, los estudios de súper-resolución en termografía no suelen profundizar en el valor radiométrico del píxel, codificado con 16 bits de forma general. Es decir, estos métodos se encargan de mejorar la resolución espacial de una imagen termográfica, mientras que el valor de los píxeles suele mantenerse similar al original. Esto significa

que, aunque se ha mejorado el aspecto visual de una imagen, no se ha intervenido en el valor radiométrico de sus píxeles. Por este motivo, si la imagen presenta lecturas erróneas en la temperatura a causa de ciertas limitaciones de la cámara termográfica, estas lecturas seguirán sin ser corregidas o ajustadas.

5.2.1. El problema de las lecturas de una cámara

Los problemas con las lecturas de una cámara termográfica son una cuestión recurrente en el campo de la termografía, pues se sabe que los valores de una cámara de baja resolución presentan una mayor limitación que los de una cámara que, de forma nativa, incorpora una resolución del sensor mayor. Este factor aumenta cuanto mayor es el término de distancia entre blanco y cámara. Además, la capacidad de captación infrarroja de una cámara depende en gran medida de la óptica incorporada, por ello las lecturas captadas por esta no serán iguales si incorpora un gran angular o un teleobjetivo.

Las cámaras con un FOV bajo suelen ser útiles para estudiar superficies a grandes distancias, ya que la lente que incorporan permite realizar un buen enfoque y encaminar un mayor flujo radiométrico al sensor. Esto deriva en una mayor concentración de píxeles para conformar la superficie estudiada en la matriz de datos. Por otro lado, las cámaras con un FOV alto pueden ser mejores soluciones si se desea tener una vista más panorámica del entorno, aunque la superficie estudiada quede definida con un número menor de píxeles en la matriz de datos.

Debido a lo comentado anteriormente, es necesario tener herramientas que permitan corregir muchas de las lecturas incorrectas que se puedan obtener de las cámaras termográficas de bajo coste, ya que en muchos escenarios la calibración que estas incorporan de fábrica sólo es apta para distancias pequeñas. Entre las posibles correcciones que pueden implementarse se encuentra la de incorporar el término de distancia respecto el blanco de estudio, para así ajustar la temperatura ofrecida por la cámara. El uso de redes neuronales básicas podría simplificar y agilizar el desarrollo de esta corrección.

5.2.2. Solución propuesta

En este punto interviene el concepto de súper-resolución radiométrica. Este concepto, siguiendo con la línea de la súper-resolución espacial, se trata de un aprendizaje supervisado donde se tiene una entrada y una etiqueta. La entrada es una imagen termográfica tomada por una cámara de bajo coste, mientras que la etiqueta se relaciona con una imagen de la misma escena, tomada en el mismo momento, pero por una cámara termográfica de mayores prestaciones. Para este TFM se tiene acceso a una cámara termográfica de bajo coste, la cámara modelo Lepton del fabricante FLIR, cuyas limitaciones técnicas son visibles para situaciones donde la distancia supera los 3 metros respecto el blanco de estudio. Esta cámara incorpora una ecuación capaz de transformar los valores radiométricos de la imagen en temperatura, haciendo uso de los términos RBFO.

La temperatura en Kelvin de la cámara Lepton puede ser calculada a partir de la ecuación 5.1.

$$T = \frac{B}{\ln\left(\frac{R}{W-O} + F\right)} \quad (5.1)$$

Donde se pueden distinguir dos modos fundamentales de trabajo, ganancia alta y ganancia baja.

- Ganancia alta denota que la cámara puede medir de forma segura temperaturas de -10 °C hasta los 140 °C. Sus términos RBFO por defecto son: $R = 355178$, $B = 1435000$, $F = 1000$, $O = 390474$.
- Ganancia baja denota que la cámara puede medir de forma segura temperaturas de -10 °C hasta los 450 °C, a costa de una mayor imprecisión en las medidas. Sus términos RBFO por defecto son: $R = 102000$, $B = 1470000$, $F = 1000$, $O = 362643$.

Por otro lado, la cámara que se quiere usar para obtener las imágenes etiqueta y así entrenar la red neuronal es la cámara modelo A615, también del fabricante FLIR. Esta cámara contiene unas particularidades que la hacen buena candidata para estimar la temperatura de un objeto a grandes distancias. Por un lado, tiene un sistema

óptico regulable capaz de enfocar el objetivo en un rango de 1 a 1000 metros. Por otro lado, implementa una función de transferencia específica capaz de transformar el valor radiométrico binarizado en valores de temperatura, donde su mayor fortaleza es que incorpora el término de distancia d en la propia función. Esta cámara tiene tres modos de funcionamiento, el modo 0 (-20 °C a 150 °C), el modo 1 (100 °C a 650 °C) y el modo 2 (300 °C a 2000 °C). Cuánto mayor es el modo, más amplio es el rango de temperaturas que abarca, pero también aumentan las imprecisiones en las medidas. En este trabajo se ha usado el modo 1.

El objetivo de la súper-resolución radiométrica planteada es llegar a prescindir de la ecuación RBFO que ofrece el fabricante de la cámara Lepton para transformar el valor radiométrico de un píxel en temperatura. La red neuronal propuesta realizará una predicción de la temperatura equivalente de ese valor, basada en la distancia respecto el blanco de medida y el valor que ha captado la cámara A615 en un momento dado. Un boceto de la red neuronal que se pretende implementar se muestra en la Figura 5.2.

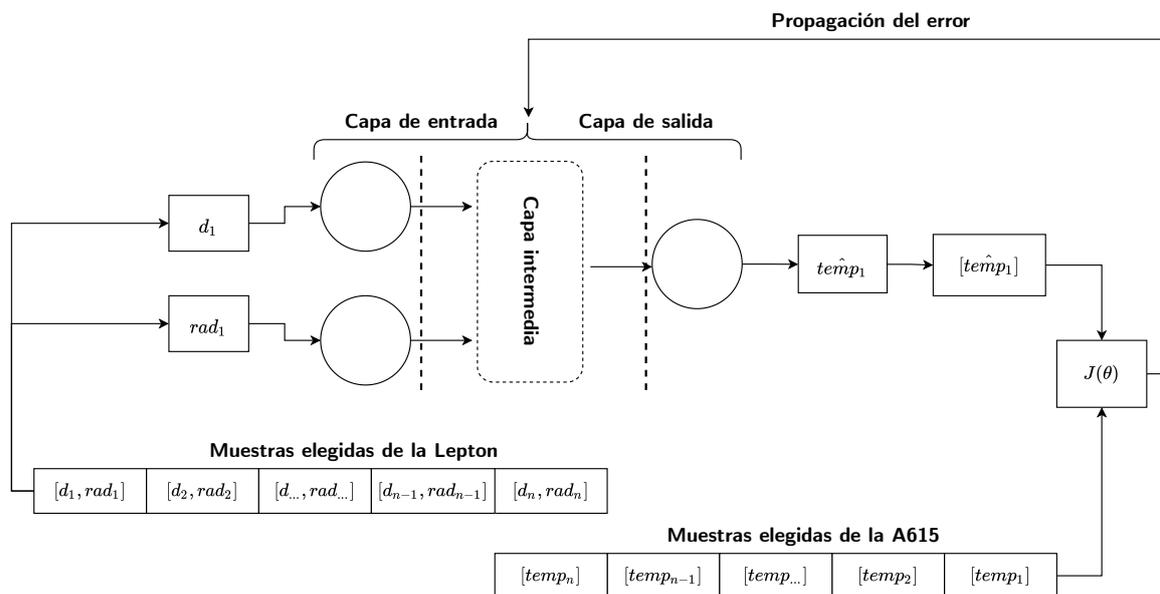


Figura 5.2: Red neuronal planteada para la súper-resolución radiométrica

La figura anterior muestra como a la entrada de la red neuronal propuesta se inserta el valor radiométrico que ofrece la cámara Lepton, junto a la distancia conocida que existe entre esta y el objeto estudiado. La etiqueta en este caso es el valor de temperatura que ofrece la cámara A615 después de aplicar su ecuación característica, por lo que los términos de distancia d , emisividad ϵ , transmisividad τ y temperatura del ambiente

t_{amb} , entre otros tantos, ya han sido tomados en cuenta a la hora de entrenar la red neuronal. La salida de esta red neuronal espera un valor, que es la temperatura predicha teniendo como entrada un valor dado y una distancia. Durante el entrenamiento, la red neuronal comparará esta salida con la etiqueta proporcionada y, a través de una función de pérdida determinada, un optimizador y un número de épocas, se intentará minimizar el error gracias al algoritmo de propagación hacia atrás.

5.2.3. Entrenamiento de la red neuronal

Para llevar a cabo el entrenamiento de la red neuronal primero es necesario implementarla en algún lenguaje de programación. En este caso se ha usado Python junto con la biblioteca de alto nivel Keras. Esta biblioteca proporciona las herramientas necesarias para prototipar una red neuronal de forma rápida y sencilla, necesitando sólo de pocas bibliotecas externas para conformar todo el circuito de entrenamiento, como podrían ser matplotlib, numpy y scikit-learn.

Para el prototipo presentado en este trabajo se plantea una red neuronal secuencial con 3 capas intermedias de 256 neuronas cada una, teniendo en cuenta que posteriormente se puede llevar a cabo un proceso de optimización para eliminar neuronas con la intención de implementar el modelo en plataformas de hardware más limitado. Siguiendo con esta línea, se contará con una sola neurona de salida, de dónde se obtendrá finalmente la temperatura predicha. Todas las capas tienen una función de activación ReLU, esta función promete buenas prestaciones puesto que por el momento no se trabajará con temperaturas negativas. Como estrategias de entrenamiento se usará la función de pérdida MAE, para así ponderar el error absoluto medio que existe entre la temperatura predicha y la etiqueta inicial introducida a la red. No obstante, otras métricas como la función de pérdida MAPE o la función de pérdida Huber serán mostradas en el proceso de entrenamiento, para así facilitar la futura parametrización de la red neuronal. La optimización de la red, que entre otras funciones tiene la de actualizar los pesos de las neuronas y realizar la propagación hacia atrás, será realizada con el optimizador Adam, muy usado en la actualidad en el campo del aprendizaje profundo. Finalmente, se ha usado un tamaño de lote de 32 para agilizar el tiempo de entrenamiento de la red neuronal.

La Tabla 5.3 muestra de forma resumida todos los hiperparámetros que conforman la red neuronal planteada.

Hiperparámetros usados	
Función de pérdida	MAE
Optimizador	Adam
Función de activación	ReLU
Número de capas intermedias	3
Número de neuronas	256
Capa final	1 neurona
Tamaño del lote	32

Tabla 5.3: Hiperparámetros de la red neuronal propuesta

De la misma forma, la biblioteca Keras permite realizar una representación de la arquitectura compilada, esta se muestra en la Figura 5.3.

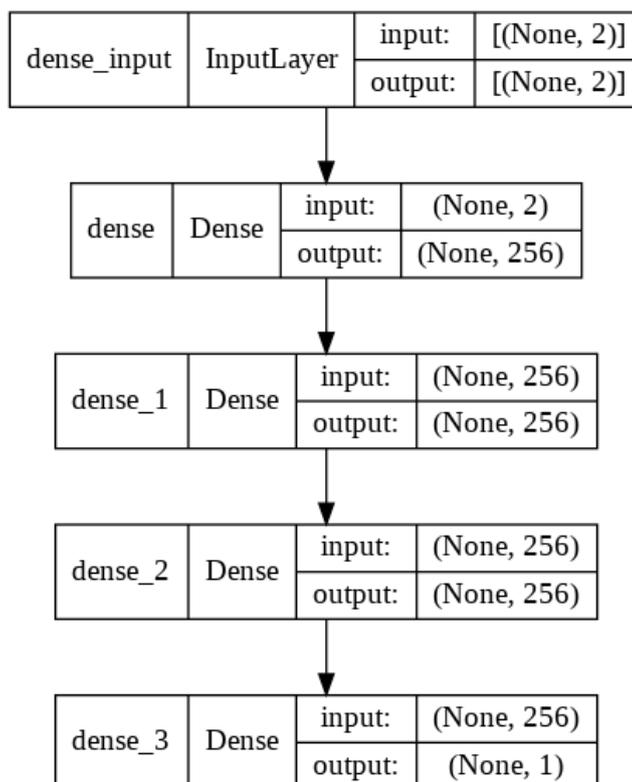


Figura 5.3: Arquitectura de la red neuronal diseñada con Keras

Las muestras usadas para el entrenamiento y evaluación de la red neuronal han sido obtenidas después de realizar un procesado concreto sobre las imágenes termográficas

de la base de datos TID, teniendo en cuenta que para esta aproximación sólo se han utilizado las imágenes que pertenecen a la cámara A615 y la cámara Lepton. Estas muestras se ilustran en la Figura 5.4, donde se puede observar un total de 64471 muestras de cada cámara dedicadas al entrenamiento y 14025 muestras de cada cámara para la evaluación de resultados.

Por consiguiente, la red neuronal será alimentada con un total de 78496 muestras, de las que el 82 % se destinan al entrenamiento y el 18 % se destinan a la evaluación de la red neuronal.

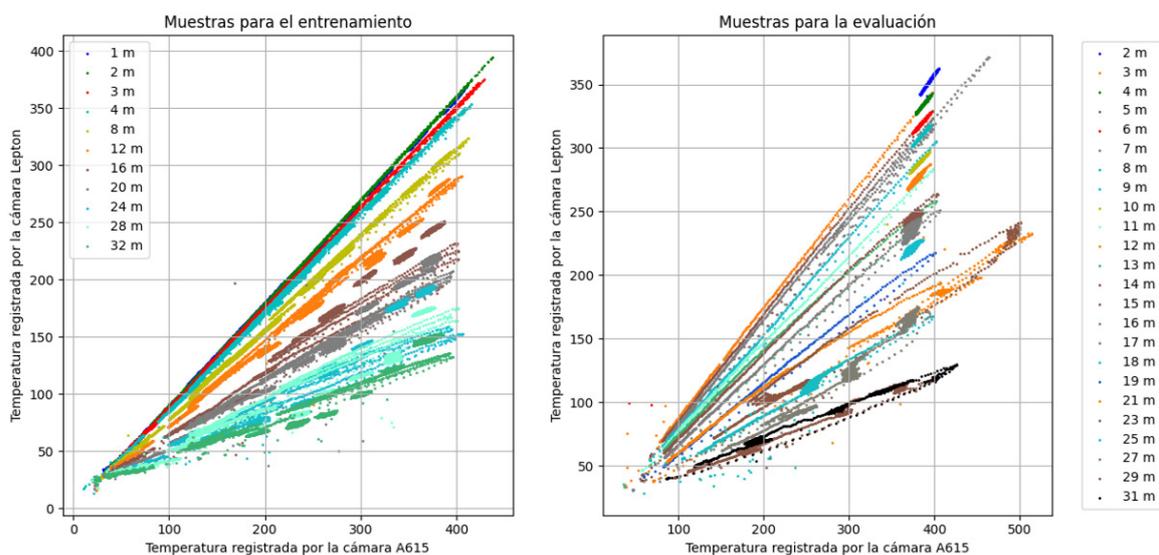


Figura 5.4: Muestras elegidas de la base de datos TID

La figura anterior representa las muestras de entrenamiento y evaluación como una nube de puntos donde cada muestra equivale al valor de temperatura medido por cada cámara. En el caso de la cámara Lepton se usó la función RBFO para pasar del valor radiométrico a temperatura, mientras que en el caso de la cámara A615 se usó una función concreta facilitada por el fabricante. Se han representado las muestras teniendo en consideración el eje de abscisas para los datos captados por la cámara A615 y el eje de ordenadas para los datos captados por la cámara Lepton. Finalmente, aunque en las gráficas se representen temperaturas, se remarca que la red neuronal tendrá como entrada el valor radiométrico de la cámara Lepton con un parámetro extra que será la distancia, mientras que la etiqueta de la red neuronal será la lectura tomada por la cámara A615 en formato de temperatura.

5.2.3.1. Estrategia para el pre-entrenamiento

Para poder evaluar las muestras destinadas al entrenamiento de la red neuronal y tener claro que no producirán ningún tipo de sobrecarga o sesgo durante este proceso, se decide hacer uso del método de validación cruzada. Esta estrategia consiste en dividir los datos de forma aleatoria en k grupos de aproximadamente el mismo tamaño. De estos grupos se emplean $k - 1$ grupos para entrenar la red neuronal, mientras que el grupo restante se emplea como validación. Este proceso se repite k veces utilizando un grupo distinto como validación en cada iteración, generando k estimaciones del error cuyo promedio se emplea como estimación final. Por este motivo, en distintas referencias consultadas se le suele denominar “Kfold” a este tipo de estrategia.

El procedimiento general se presenta en el Protocolo 3.

Protocolo 3: Proceso de Kfold

- Mezclar la base de datos de forma aleatoria.
- Dividir la base de datos en k grupos.
- Para cada grupo:
 - Tomar un grupo como base de datos para la evaluación.
 - Tomar los grupos restantes como bases de datos para el entrenamiento.
 - Entrenar la red neuronal con el grupo de entrenamiento y evaluarla con el grupo de prueba.
 - Conservar la puntuación de la evaluación y descartar la red neuronal.
- Resumir la habilidad de la red neuronal utilizando la media aritmética de las puntuaciones anteriores.

Para el proceso de validación cruzada se han decidido usar 3 grupos. Esto significa que durante 3 iteraciones distintas se procede con el entrenamiento de la red neuronal con el 80 % de las muestras y se evalúa con el 20 % restante, teniendo en cuenta que esta división de datos es distinta en cada iteración. La biblioteca usada para implementar la validación cruzada ha sido scikit-learn, ya que se tiene acceso a una función Python

que facilita el proceso.

La Figura 5.5 muestra cuál ha sido el mecanismo de Kfold implementado para evaluar la consistencia de los datos dedicados al futuro entrenamiento.

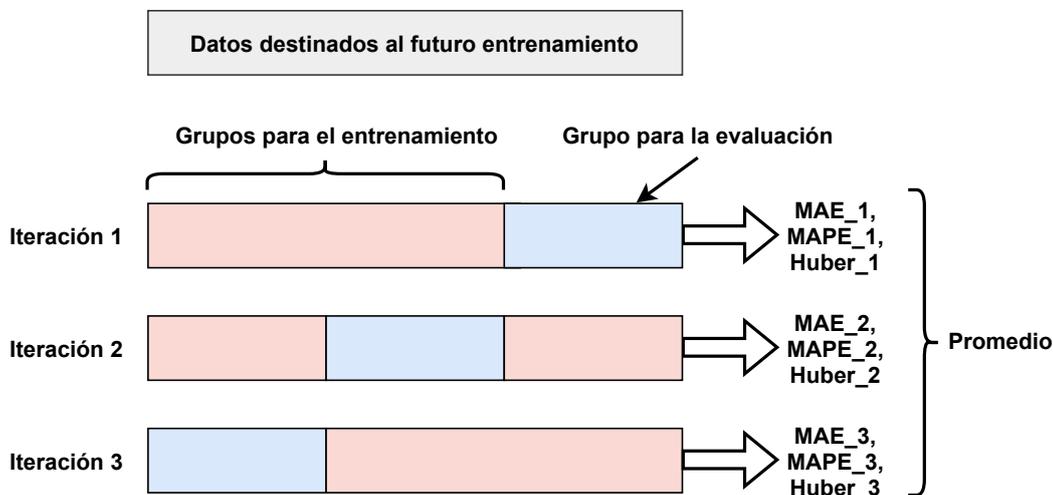


Figura 5.5: Mecanismo de Kfold implementado

Los resultados de aplicar el proceso de Kfold durante 3 iteraciones y un entrenamiento de 3000 épocas se muestran en la Tabla 5.4.

	MAE (°C)	MAPE (%)	Pérdida de Huber (°C)
Grupo 1	3.994	2.605	3.555
Grupo 2	4.417	2.846	3.963
Grupo 3	3.729	2.447	3.281
Promedio	4.047 ± 0.283	2.632 ± 0.164	3.599 ± 0.280

Tabla 5.4: Resultados del mecanismo Kfold implementado

5.2.3.2. Entrenamiento total de la red

Tras analizar los resultados obtenidos en el Kfold se estima que el margen de error dado por este, teniendo en cuenta las necesidades del trabajo, es aceptable. Por este motivo, a partir de ahora se usarán todas las muestras anteriores de forma íntegra para poder plantear un entrenamiento completo. La red neuronal es entrenada durante 3000 épocas con las mismas condiciones comentadas en la sección anterior. Los resultados

finales del entrenamiento se muestran en la Figura 5.6, donde se ofrecen valores de MAE, MAPE y pérdida de Huber a lo largo del proceso de entrenamiento.

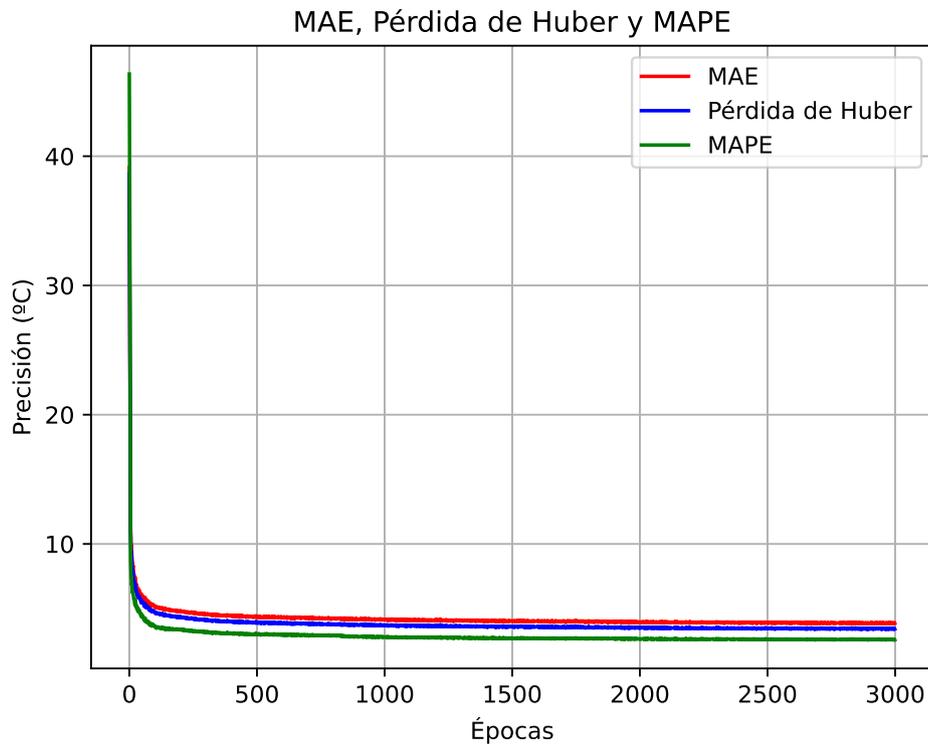


Figura 5.6: Gráfica resultante del proceso de entrenamiento

Los resultados numéricos obtenidos en el momento de la época 3000 se presentan en la Tabla 5.5.

	MAE (°C)	MAPE (%)	Pérdida de Huber (°C)
Resultado	3.852 ± 1.065	2.576 ± 1.399	3.413 ± 1.061

Tabla 5.5: Resultados del entrenamiento planteado

5.3. Experimentación con la súper-resolución espacial

En la literatura consultada, una forma muy usada de estimar el rendimiento de un método de súper-resolución espacial es a través de la PSNR y el SSIM. Para ello, normalmente se toma una imagen de referencia y se le aplica una degradación concreta, como podría ser disminuir su resolución con una interpolación clásica y aplicar a esta imagen algún tipo de ruido o desenfoque. En este punto, independientemente de que técnica se aplique, es teóricamente imposible que la imagen LR resultante pueda volver a parecerse a la imagen HR original. Por este motivo, lo que se suele hacer es aplicar las métricas nombradas al comparar la imagen SR resultante de un método de súper-resolución espacial con la imagen HR original. Se estima de esta forma que cuanto mayor sea la semejanza entre estas dos, mejor será el método propuesto.

Durante el transcurso de los experimentos propuestos en este TFM se pretenden aplicar métricas similares a las existentes en el estado de la técnica, teniendo en cuenta la peculiaridad de que se trabaja con imágenes termográficas y que la mayoría de los métodos de súper-resolución espacial analizados no han sido pensados para estas. En este sentido, también será interesante calcular cual es la diferencia real, en términos de temperatura, que tiene la imagen SR resultante con la imagen HR original. A partir de estos datos se podría parametrizar cuál es la pérdida inherente de aplicar ciertos métodos de súper-resolución espacial.

Para todo el desarrollo posterior de este trabajo se proponen tres experimentos distintos, aportando gráficas y tablas de los resultados obtenidos:

- **Experimento 1.** El primer experimento consiste en comparar la similitud de una imagen SR de la cámara Lepton con una imagen de la cámara A615, teniendo en cuenta que los factores de FOV son distintos en cada una. Es decir, se plantea una comparativa directa con métricas como la PSNR y SSIM entre una imagen con resolución original de 640 x 480 frente una imagen con resolución original de 160 x 120 pero que ha sido procesada con métodos de súper-resolución espacial. Para que se puedan comparar ambas imágenes es necesario que ofrezcan valores en formato de temperatura, ya que los valores radiométricos de ambas cámaras

presentan rangos numéricos distintos. La intuición de este experimento es, por un lado, comprobar que efectivamente existe una mejora visual al aplicar un método de súper-resolución espacial concreto a una imagen LR que ofrece una cámara de bajo coste. Una forma de comprobar esto es comparando directamente esta imagen SR con una imagen de una cámara que, de forma nativa, tenga la misma resolución. Por otro lado, en este experimento se comprobará que valores de PSNR y SSIM se obtienen al comparar dos imágenes de cámaras termográficas distintas, que incluyen diferentes ópticas y cuyas lecturas de temperatura pueden diferir por la naturaleza de su fabricación.

- **Experimento 2.** Para realizar este experimento se tomarán como entrada las imágenes de la cámara A615, aplicando sobre estas una interpolación bicúbica para obtener imágenes LR. De forma posterior, se les aplicarán los métodos de súper-resolución espacial, obteniendo de estos una serie de imágenes SR que serán comparadas con las originales. En este punto se podrá modelar como afectan estos métodos de súper-resolución espacial al resolver una imagen LR en factores de escala $\times 2$ y $\times 4$, teniendo en cuenta que se tratan de imágenes de la misma cámara termográfica. Las métricas que se utilizarán serán la PSNR y SSIM.
- **Experimento 3.** En este experimento será necesario usar como entrada las imágenes de la cámara A35 y Lepton, a las que se les aplicará directamente los métodos de súper-resolución espacial. El objetivo de este experimento consistirá en parametrizar que pérdidas experimentan los píxeles de las imágenes al aplicar estos métodos, teniendo en cuenta que estas pérdidas serán obtenidas en función de grados centígrados.

5.3.1. Pre-procesado para los métodos de MATLAB

Para que los métodos implementados en el entorno de MATLAB funcionen correctamente, las imágenes termográficas de 16 bits necesitan ser convertidas a imágenes de 8 bits de datos, con un rango de valores que oscila entre 0 y 255. Sin embargo, el objetivo final es obtener una imagen SR cuyos valores de píxeles se correspondan con los valores radiométricos originales obtenidos por las cámaras. Por este motivo, para poder realizar el proceso inverso al final del método y así pasar de 8 bits a 16 bits,

es necesario almacenar valores de referencia de la imagen original, con la intención de que la imagen SR se parezca lo más posible a esta. El flujo de trabajo que abarca el proceso de súper-resolución con este tipo de métodos se muestra en la Figura 5.7.

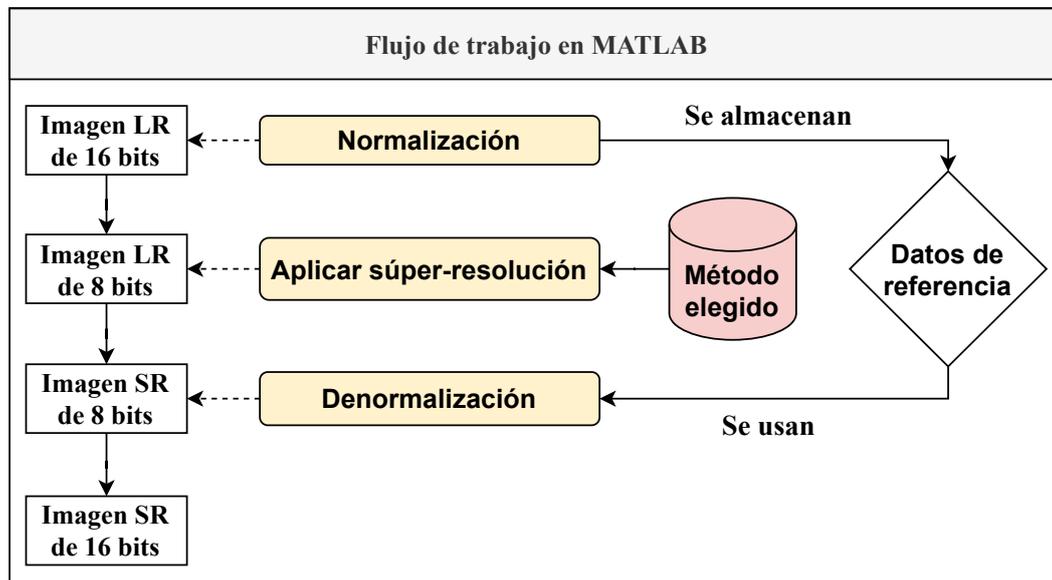


Figura 5.7: Flujo de trabajo contemplado en MATLAB para la súper-resolución espacial

5.3.2. Pre-procesado para los métodos de Python

Los métodos implementados en Python tienen una particularidad respecto los métodos anteriores. Una gran parte de las redes neuronales que se usan en estos métodos para la súper-resolución espacial han sido entrenadas con imágenes RGB, por ello esperan a su entrada un formato de 3 dimensiones.

Para poder trabajar con imágenes termográficas se debe usar una metodología similar a la anterior, aunque con ciertos ajustes. Primero, es necesario convertir la imagen de 16 bits a una imagen 8 bits de datos, almacenando los valores de referencia necesarios para aplicar el proceso inverso a la salida de la red neuronal. En este punto, se crea una imagen RGB falsa apilando tres imágenes LR de iguales características, que en este caso es la imagen LR que entra a la red neuronal. Con este pequeño procesado ya se resuelve una gran parte de los problemas derivados de trabajar con redes neuronales entrenadas para imágenes RGB y se puede aplicar la súper-resolución espacial, teniendo en cuenta que a la salida de la red neuronal se tiene que efectuar el proceso para obtener cualquiera de las tres dimensiones resultantes y volver a convertir la imagen a 16 bits.

El flujo de trabajo que abarca el proceso de súper-resolución con métodos basados en redes neuronales se muestra en la Figura 5.8.

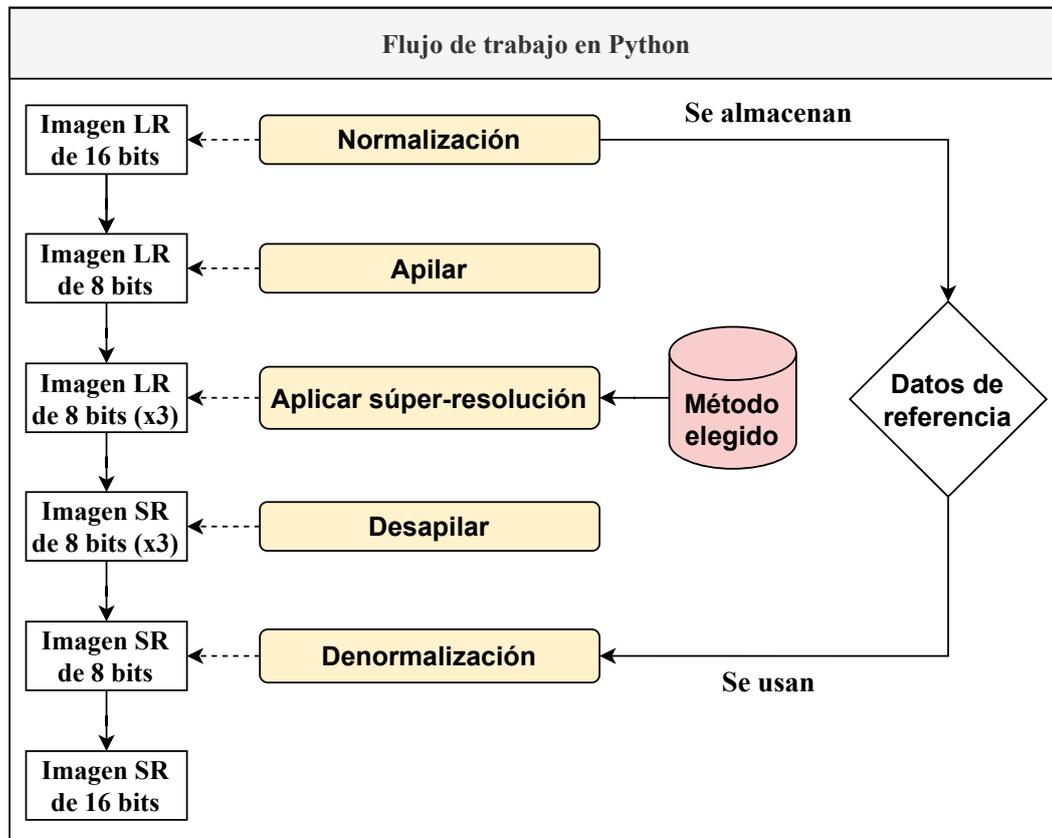


Figura 5.8: Flujo de trabajo contemplado en Python para la súper-resolución espacial

5.4. Experimentación con la súper-resolución radio-métrica

Una vez entrenada la red neuronal que permite la corrección de temperatura a través del término de distancia, es necesario comprobar su desempeño con las muestras de la base de datos TID-II. Durante el transcurso de la experimentación se contará con los datos originales proporcionados por la cámara Lepton, los datos de la cámara A615 y la corrección de temperatura que se obtiene de la red neuronal a partir de los valores de la cámara Lepton. Por este motivo, es de gran interés observar cuál es la diferencia absoluta entre estos datos. Es decir, se obtendrá la diferencia entre el valor original de la cámara Lepton y el valor de la cámara A615, además de la diferencia entre el valor predicho por la red neuronal y el valor de la cámara A615, que en teoría debería ser

mucho más pequeña que la primera. Con esto se podrá obtener una estimación simple de cómo opera la red neuronal propuesta.

Al tratarse de muchas muestras que procesar, un buen método para ver el desempeño promedio de la red neuronal es a través del MAE y el MAPE, aplicando estas métricas para cada distancia de medida. En este sentido, se realizarán una serie de rutinas que permiten representar estos datos en gráficas lineales y en formato de barras, además de almacenar en tablas los promedios generales obtenidos. Un ejemplo de la metodología propuesta se muestra de forma gráfica en la Figura 5.9.

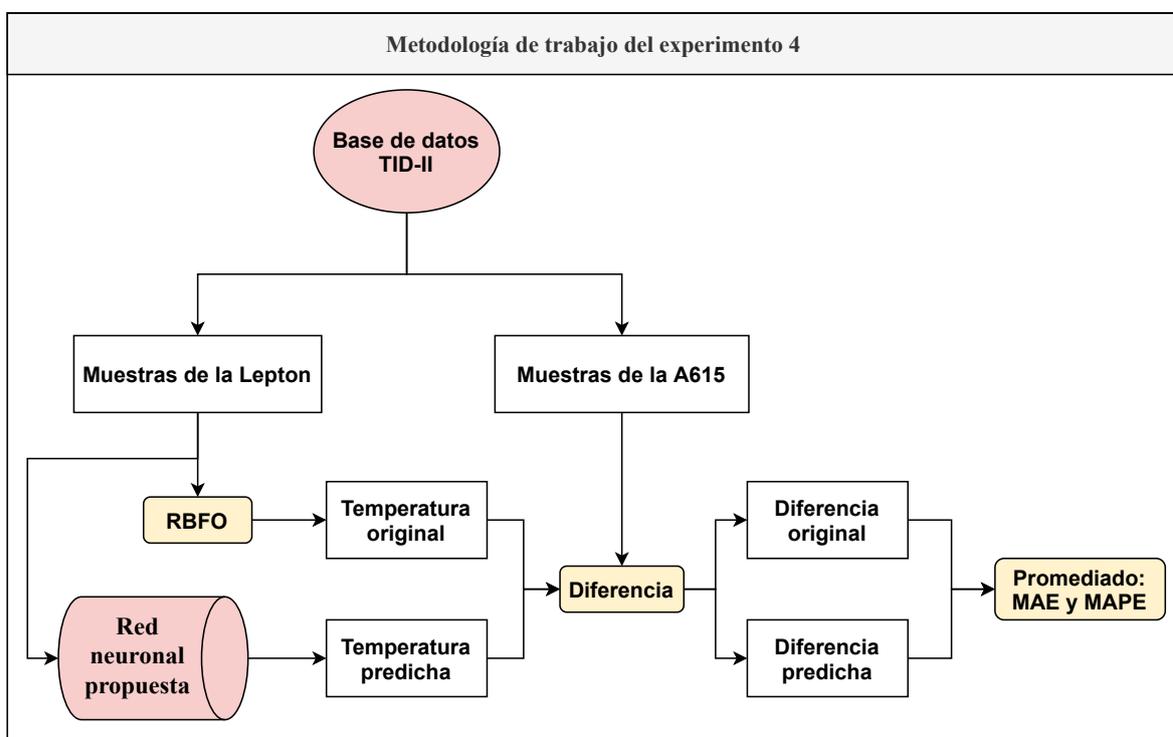


Figura 5.9: Flujo de trabajo contemplado para la súper-resolución radiométrica

5.5. Experimentación con ambas súper-resoluciones

Finalmente, una vez completadas las etapas anteriores de experimentación, se plantea el uso combinado de ambos tipos de súper-resolución, tanto espacial como radiométrica. El objetivo de esto es comprobar, para una imagen SR que ya ha sido procesada y, por lo tanto, con ciertas pérdidas, como afecta la inclusión de la red neuronal propuesta para la corrección de temperatura. En este proceso de experimentación se usará

la cámara Lepton y un método de súper-resolución espacial determinado, elegido a partir de los resultados de los experimentos anteriores.

El flujo de trabajo radicará en aplicar la súper-resolución radiométrica a píxeles concretos de la imagen original y a píxeles concretos de la imagen SR obtenida de la súper-resolución espacial, comparando los resultados de esta operación. Como se ha visto anteriormente, en la Figura 5.10 se muestra el planteamiento llevado a cabo de forma visual, teniendo en cuenta que se usará la base de datos TID-II.

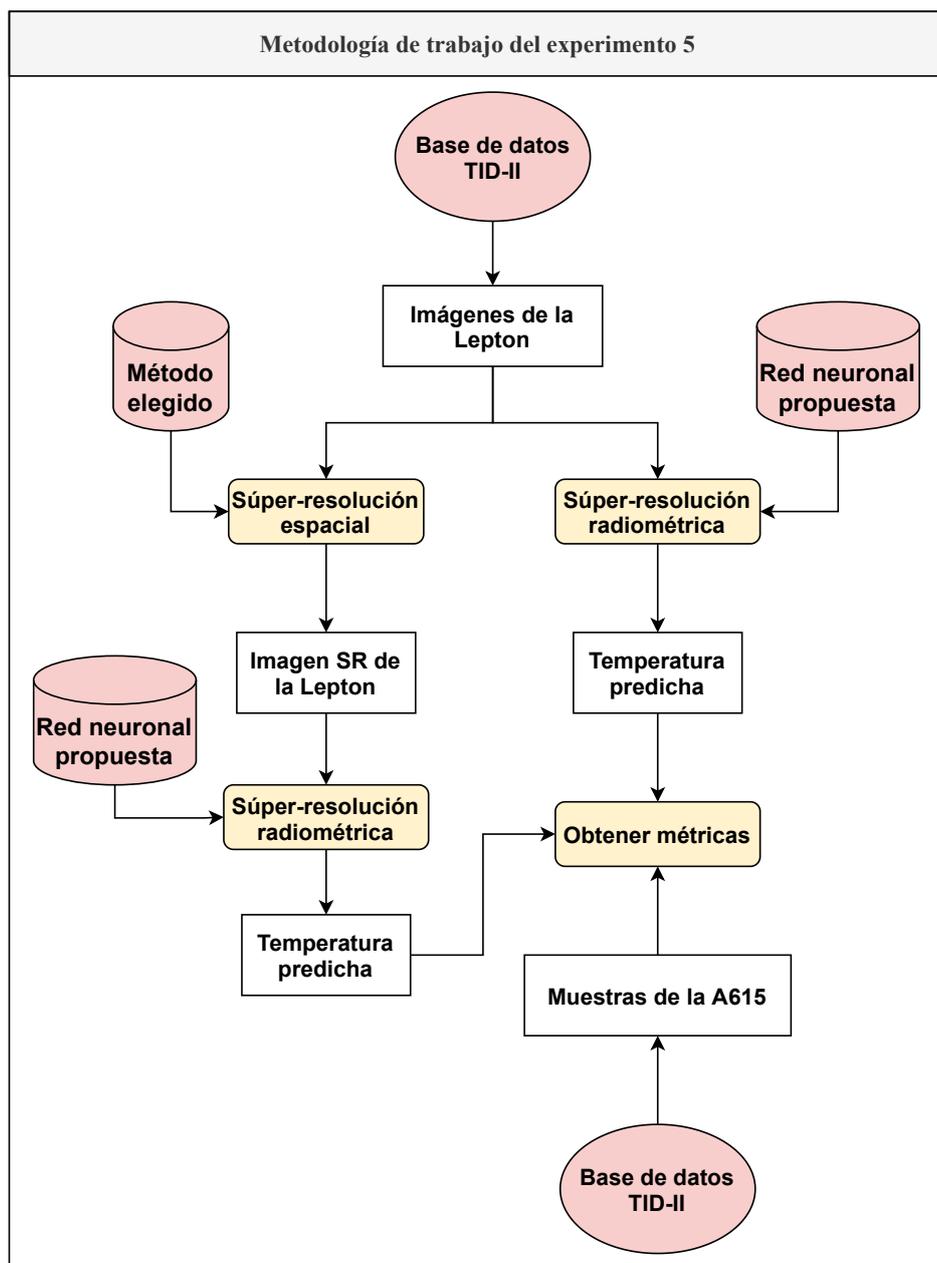


Figura 5.10: Flujo de trabajo contemplado para la conjunción de súper-resoluciones

Capítulo 6

Resultados

En las siguientes secciones se discutirán los resultados obtenidos de los experimentos enunciados en el capítulo anterior.

6.1. Resultados del experimento 1

En esta sección se presentan los resultados del experimento 1, obtenidos de aplicar los métodos de súper-resolución espacial sobre imágenes de la cámara A615 e imágenes de la cámara Lepton pertenecientes a la base de datos DASSR. De esta manera, una comparación con métricas como la PSNR y SSIM será realizada a través de distintas gráficas y tablas con los valores obtenidos al procesar las imágenes SR de cada cámara. Por otro lado, una de las claves de la súper-resolución es la mejora en la percepción visual de una imagen, por ello en los resultados de los experimentos 2 y 3 se presentarán distintas muestras obtenidas de aplicar los métodos de súper-resolución espacial a estas imágenes. Sin embargo, debido a la gran cantidad de métodos que se han estudiado en este trabajo, sólo se presentarán imágenes de ejemplos representativos. En un anexo posterior se adjuntará una cantidad mayor de imágenes para completar estos apartados.

La Tabla 6.1 muestra los resultados de comparar directamente las imágenes SR de la cámara Lepton en escala x4 con las imágenes originales de la cámara A615. Más concretamente, se realiza una comparativa de los métodos de reconstrucción implementados en MATLAB. En esta tabla se ha sombreado con color verde los métodos que consiguen un mejor desempeño general. Se debe tener en cuenta que los resultados mostrados en

esta tabla ofrecen un valor promedio de cada uno de los grupos segmentados de la base de datos, además de su desviación típica. Esto es aplicable al resto de las tablas mostradas durante las siguientes secciones.

Métricas - Escala x4				
Método	PSNR	SSIM	PSNR	SSIM
Interpolación	44.73 ± 1.79	0.9911 ± 0.0033	46.06 ± 2.93	0.9921 ± 0.0034
IBP	44.98 ± 1.88	0.9914 ± 0.0033	46.28 ± 3.38	0.9922 ± 0.0035
PG	43.49 ± 2.01	0.9873 ± 0.005	45.72 ± 3.39	0.9910 ± 0.0042
POCS	44.08 ± 1.81	0.9896 ± 0.0037	45.75 ± 2.85	0.9915 ± 0.0037
SR Robusta	44.94 ± 1.82	0.9913 ± 0.0033	46.35 ± 3.78	0.9921 ± 0.0034
	DASSR-I		DASSR-II	
	Conjunto de datos			
	DASSR-III		DASSR-IV	
Interpolación	44.14 ± 1.71	0.9875 ± 0.0149	37.96 ± 2.1	0.9572 ± 0.0038
IBP	44.38 ± 1.75	0.988 ± 0.0152	38.15 ± 2.24	0.9576 ± 0.0038
PG	43.41 ± 2.20	0.9839 ± 0.0184	37.56 ± 2.36	0.9542 ± 0.0072
POCS	43.63 ± 1.90	0.9856 ± 0.0161	37.64 ± 2.16	0.9542 ± 0.0050
SR Robusta	44.31 ± 1.7	0.9878 ± 0.0151	38.15 ± 2.24	0.9576 ± 0.0038
Método	PSNR	SSIM	PSNR	SSIM
Métricas - Escala x4				

Tabla 6.1: Resultados del experimento 1 - Métodos de MATLAB en escala x4

Como se puede observar en la tabla anterior, los resultados oscilan entre valores de PSNR de 43 y 44 dB. Según la fórmula de la PSNR para imágenes de 16 bits, el máximo al que se podría llegar es $20\log(2^{16}) - MSE$, esto equivale a una cantidad ideal aproximada de 96 dB. Los resultados de este experimento están muy por debajo de la cantidad ideal de PSNR. Esto tiene sentido porque, tal como se ha comentado anteriormente, se están comparando dos imágenes cuya representación de la escena es distinta, ya que han sido obtenidas con ópticas diferentes, como se puede observar en un ejemplo facilitado por la Figura 6.1. En esta figura se aprecia la escena captada por la cámara A615 y por la cámara Lepton, esta última interpolada con un factor x4. Se debe añadir que la cámara A615 consigue lecturas de temperatura más precisas que la cámara Lepton, algo que contribuye en la disminución de valores en la PSNR.

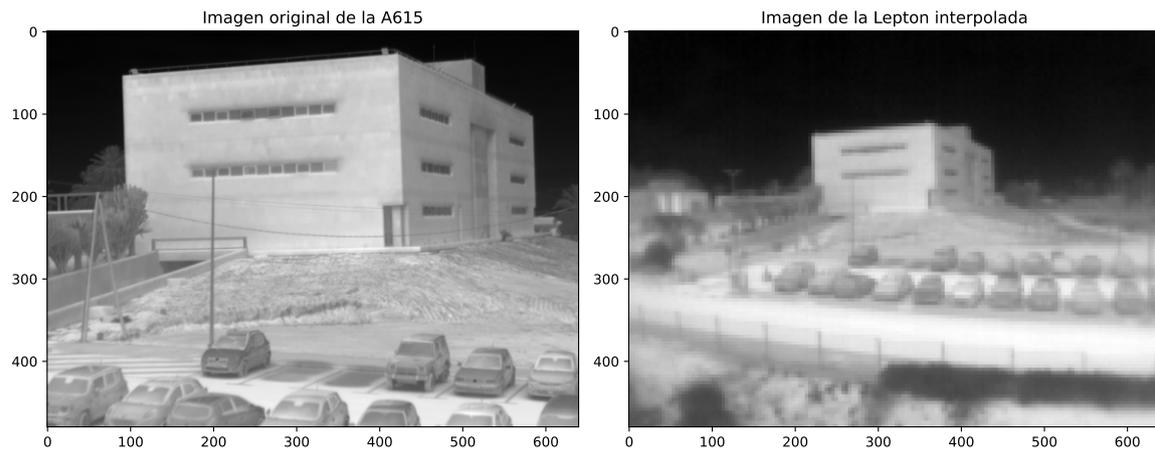


Figura 6.1: Comparación de escenas observadas entre cámaras con distinta óptica

Por otro lado, en la Tabla 6.2 se puede observar la misma metodología llevada a cabo que en la tabla anterior. No obstante, en este caso se han analizado los resultados de los métodos de súper-resolución basados en redes neuronales. En esta tabla se ha sombreado con un color verde el método con un mayor valor promedio de PSNR y SSIM, mientras que se ha usado un color azul para sombrear aquellos métodos cuya imagen SR de salida tiene una mejor calidad visual que otras estudiadas. Se debe resaltar que los métodos marcados en verde, aunque ofrecen un mejor resultado numérico tanto de PSNR como de SSIM, no son los que generalmente ofrecen un mejor resultado visual. Por este motivo, se ha necesitado clasificar los métodos a través de estos dos colores.

Tanto en la Tabla 6.2 como en todas las tablas que forman el grueso de resultados de los experimentos 1, 2 y 3, se ha usado la nomenclatura “SRGAN_TH”, “EDSR_TH” y “WDSR_TH” para denotar los métodos basados en redes neuronales que han sido entrenados con la base de datos ADAS. De la misma manera, el mismo código de colores usado en las Tablas 6.1 y 6.2 se usa en los resultados de los experimentos 2 y 3.

	Métricas - Escala x4			
Método	PSNR	SSIM	PSNR	SSIM
VDSR	44.85 ± 1.8	0.9912 ± 0.0033	46.03 ± 2.92	0.992 ± 0.0034
SRGAN	44.76 ± 1.8	0.9909 ± 0.0033	46.03 ± 2.92	0.9919 ± 0.0034
SRGAN TH	44.8 ± 1.8	0.991 ± 0.0033	46.02 ± 2.91	0.9919 ± 0.0034
EDSR	44.85 ± 1.79	0.9912 ± 0.0033	46.02 ± 2.91	0.9919 ± 0.0034
EDSR TH	44.85 ± 1.79	0.9912 ± 0.0033	46.02 ± 2.91	0.9919 ± 0.0034
MDSR	44.82 ± 1.8	0.9911 ± 0.0033	46.03 ± 2.93	0.992 ± 0.0034
WDSR	44.85 ± 1.79	0.9912 ± 0.0033	46.02 ± 2.91	0.9919 ± 0.0034
WDSR TH	44.86 ± 1.8	0.9912 ± 0.0033	46.03 ± 2.91	0.992 ± 0.0034
ESRGAN	44.86 ± 1.82	0.9908 ± 0.0034	46.06 ± 2.86	0.9919 ± 0.0034
Real-ESRGAN	44.93 ± 1.8	0.9913 ± 0.0033	45.8 ± 2.63	0.9916 ± 0.0034
	DASSR-I		DASSR-II	
	Conjunto de datos			
	DASSR-III		DASSR-IV	
VDSR	44.19 ± 1.67	0.9875 ± 0.0153	38.1 ± 2.24	0.9571 ± 0.0038
SRGAN	44.1 ± 1.72	0.9871 ± 0.0154	38.06 ± 2.24	0.9568 ± 0.0039
SRGAN TH	44.15 ± 1.68	0.9873 ± 0.0153	38.07 ± 2.23	0.9566 ± 0.0038
EDSR	44.17 ± 1.66	0.9874 ± 0.0152	38.1 ± 2.23	0.957 ± 0.0038
EDSR TH	44.17 ± 1.66	0.9874 ± 0.0152	38.1 ± 2.23	0.957 ± 0.0038
MDSR	44.17 ± 1.68	0.9874 ± 0.0154	38.07 ± 2.24	0.9569 ± 0.0038
WDSR	44.18 ± 1.66	0.9874 ± 0.0152	38.1 ± 2.23	0.957 ± 0.0038
WDSR TH	44.19 ± 1.67	0.9875 ± 0.0152	38.11 ± 2.24	0.9571 ± 0.0038
ESRGAN	44.19 ± 1.7	0.9872 ± 0.0153	38.06 ± 2.23	0.9561 ± 0.0039
Real-ESRGAN	44.15 ± 1.63	0.9873 ± 0.0153	38.13 ± 2.23	0.9559 ± 0.0038
Método	PSNR	SSIM	PSNR	SSIM
	Métricas - Escala x4			

Tabla 6.2: Resultados del experimento 1 - Métodos de Python en escala x4

En la Figura 6.2 se representa en formato de gráfica de barras los resultados de la métrica PSNR obtenidos del procesamiento final de este experimento. De forma general, los valores de PSNR son más interesantes que los de SSIM, ya que ofrecen información directa sobre los valores de píxel de cada imagen.

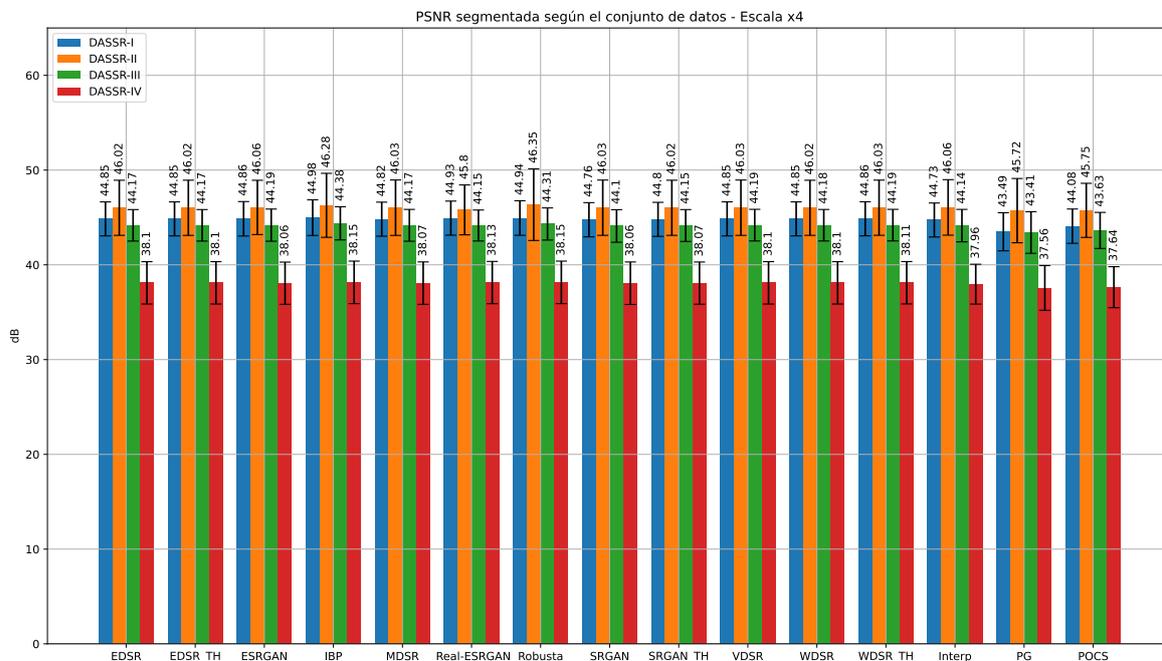


Figura 6.2: Representación gráfica de las Tablas 6.1 y 6.2 - Escala x4

6.2. Resultados del experimento 2

En esta sección se presentan los resultados obtenidos de aplicar los métodos de súper-resolución espacial discutidos en el capítulo anterior sobre las imágenes de la cámara A615 de la base de datos DASSR. En los resultados de este experimento se pretende representar algunas muestras donde quede ilustrado visualmente el efecto de estos métodos sobre imágenes que han sido previamente degradadas con una interpolación bicúbica. De la misma forma, se aplicará la PSNR y SSIM entre las imágenes SR y las originales, comprobando así el desempeño general de los métodos.

En las Tablas 6.3, 6.4, 6.5 y 6.6 se representan los resultados obtenidos de procesar las imágenes SR obtenidas. La Tabla 6.3 muestra el desempeño general de los métodos de reconstrucción para una escala x2 y la Tabla 6.4 muestra lo mismo para una escala x4.

Métricas - Escala x2				
Método	PSNR	SSIM	PSNR	SSIM
Interpolación	69.55 ± 6.83	0.9999 ± 0.0001	76.68 ± 2.77	1.0 ± 0.0
IBP	73.42 ± 2.98	1.0 ± 0.0	80.12 ± 2.44	1.0 ± 0.0
PG	69.83 ± 6.31	0.9999 ± 0.0001	76.99 ± 4.64	1.0 ± 0.0
POCS	69.77 ± 5.74	0.9999 ± 0.0001	77.24 ± 2.63	1.0 ± 0.0
SR Robusta	76.22 ± 3.12	1.0 ± 0.0	81.67 ± 1.76	1.0 ± 0.0
		DASSR-I	DASSR-II	
Conjunto de datos				
		DASSR-III	DASSR-IV	
Interpolación	78.35 ± 2.51	1.0 ± 0.0	57.05 ± 1.21	0.9994 ± 0.0001
IBP	77.81 ± 3.25	1.0 ± 0.0	61.86 ± 2.25	0.9994 ± 0.0002
PG	75.68 ± 6.89	1.0 ± 0.0001	54.55 ± 3.63	0.9982 ± 0.001
POCS	77.75 ± 3.23	1.0 ± 0.0	58.36 ± 0.76	0.9992 ± 0.0001
SR Robusta	80.44 ± 2.88	1.0 ± 0.0	63.96 ± 1.22	0.9996 ± 0.0001
Método	PSNR	SSIM	PSNR	SSIM
Métricas - Escala x2				

Tabla 6.3: Resultados del experimento 2 - Métodos de MATLAB en escala x2

Métricas - Escala x4				
Método	PSNR	SSIM	PSNR	SSIM
Interpolación	64.61 ± 6.87	0.9997 ± 0.0004	72.72 ± 3.34	1.0 ± 0.0
IBP	67.16 ± 5.13	0.9998 ± 0.0001	76.49 ± 3.23	1.0 ± 0.0
PG	60.86 ± 6.38	0.9993 ± 0.0008	68.04 ± 4.79	0.9999 ± 0.0001
POCS	62.86 ± 6.31	0.9996 ± 0.0004	71.19 ± 3.82	1.0 ± 0.0
SR Robusta	71.03 ± 4.01	0.9999 ± 0.0001	78.41 ± 2.36	1.0 ± 0.0
	DASSR-I		DASSR-II	
	Conjunto de datos			
	DASSR-III		DASSR-IV	
Interpolación	73.91 ± 3.6	1.0 ± 0.0004	51.81 ± 1.14	0.9974 ± 0.0
IBP	73.8 ± 5.14	0.9999 ± 0.0005	55.19 ± 1.95	0.9978 ± 0.0001
PG	69.84 ± 4.75	0.9999 ± 0.0006	49.0 ± 0.74	0.9954 ± 0.0001
POCS	72.18 ± 3.92	1.0 ± 0.0004	51.8 ± 0.83	0.9975 ± 0.0
SR Robusta	76.03 ± 4.4	1.0 ± 0.0003	58.15 ± 1.08	0.9984 ± 0.0
Método	PSNR	SSIM	PSNR	SSIM
Métricas - Escala x4				

Tabla 6.4: Resultados del experimento 2 - Métodos de MATLAB en escala x4

En las Tablas 6.5 y 6.6 se realizan los mismos cálculos, pero esta vez para los métodos de súper-resolución espacial basados en redes neuronales. En líneas generales, se puede observar que el desempeño de estos métodos, basado en métricas como la PSNR y el SSIM, es menor para una escala x4. Se estima que esto seguirá sucediendo para escalas mayores, debido fundamentalmente a la baja cantidad de píxeles que se tiene en una imagen LR de 160 x 120 frente una imagen LR de 320 x 240, además de la capacidad del propio método para convertirla en una imagen SR.

Una solución a este problema podría ser aplicar el factor de escala de forma separada. Es decir, en vez de aplicar directamente una escala x4 a una imagen de 160 x 120, aplicar dos veces una escala x2. No obstante, se deberían tener en cuenta las pérdidas resultantes de aplicar dos veces el método de súper-resolución espacial.

	Métricas - Escala x2			
Método	PSNR	SSIM	PSNR	SSIM
VDSR	78.02 ± 2.18	1.0 ± 0.0	82.16 ± 1.55	1.0 ± 0.0
SRGAN	75.01 ± 2.8	1.0 ± 0.0	79.24 ± 2.13	1.0 ± 0.0
SRGAN TH	74.85 ± 1.91	1.0 ± 0.0	79.77 ± 1.63	1.0 ± 0.0
EDSR	78.03 ± 1.63	1.0 ± 0.0	80.93 ± 1.03	1.0 ± 0.0
EDSR TH	78.45 ± 1.66	1.0 ± 0.0	81.12 ± 0.94	1.0 ± 0.0
MDSR	78.13 ± 2.14	1.0 ± 0.0	82.29 ± 1.57	1.0 ± 0.0
WDSR	77.94 ± 1.77	1.0 ± 0.0	80.99 ± 1.05	1.0 ± 0.0
WDSR TH	76.79 ± 2.61	1.0 ± 0.0	80.68 ± 1.22	1.0 ± 0.0
ESRGAN	-----	-----	-----	-----
Real-ESRGAN	72.67 ± 3.47	0.9999 ± 0.0	78.67 ± 3.34	1.0 ± 0.0
	DASSR-I		DASSR-II	
	Conjunto de datos			
	DASSR-III		DASSR-IV	
VDSR	81.07 ± 1.91	1.0 ± 0.0	67.78 ± 1.24	0.9998 ± 0.0
SRGAN	78.19 ± 2.27	1.0 ± 0.0	65.18 ± 0.7	0.9997 ± 0.0001
SRGAN TH	78.28 ± 2.3	1.0 ± 0.0	65.7 ± 1.03	0.9997 ± 0.0001
EDSR	80.63 ± 1.14	1.0 ± 0.0	66.79 ± 1.18	0.9998 ± 0.0001
EDSR TH	80.97 ± 1.04	1.0 ± 0.0	66.93 ± 1.25	0.9998 ± 0.0001
MDSR	81.15 ± 2.08	1.0 ± 0.0	69.3 ± 0.88	0.9999 ± 0.0
WDSR	80.69 ± 1.18	1.0 ± 0.0	66.81 ± 1.17	0.9998 ± 0.0001
WDSR TH	80.3 ± 1.52	1.0 ± 0.0	66.35 ± 1.12	0.9998 ± 0.0001
ESRGAN	-----	-----	-----	-----
Real-ESRGAN	76.68 ± 4.78	1.0 ± 0.0	58.96 ± 1.17	0.9987 ± 0.0003
Método	PSNR	SSIM	PSNR	SSIM
	Métricas - Escala x2			

Tabla 6.5: Resultados del experimento 2 - Métodos de Python en escala x2

Métricas - Escala x4				
Método	PSNR	SSIM	PSNR	SSIM
VDSR	74.14 ± 3.43	1.0 ± 0.0	80.05 ± 1.71	1.0 ± 0.0
SRGAN	70.94 ± 3.51	0.9999 ± 0.0001	77.22 ± 1.89	1.0 ± 0.0
SRGAN TH	72.43 ± 3.19	0.9999 ± 0.0	78.57 ± 1.93	1.0 ± 0.0
EDSR	72.1 ± 3.54	0.9999 ± 0.0001	78.6 ± 1.9	1.0 ± 0.0
EDSR TH	72.59 ± 3.31	0.9999 ± 0.0	78.62 ± 1.8	1.0 ± 0.0
MDSR	75.37 ± 3.0	1.0 ± 0.0	80.33 ± 1.68	1.0 ± 0.0
WDSR	72.03 ± 3.59	0.9999 ± 0.0001	78.5 ± 1.82	1.0 ± 0.0
WDSR TH	72.05 ± 3.58	0.9999 ± 0.0001	78.51 ± 1.86	1.0 ± 0.0
ESRGAN	64.68 ± 2.43	0.9996 ± 0.0002	71.35 ± 2.95	0.9999 ± 0.0001
Real-ESRGAN	70.41 ± 3.77	0.9999 ± 0.0001	77.15 ± 3.28	1.0 ± 0.0
	DASSR-I		DASSR-II	
	Conjunto de datos			
	DASSR-III		DASSR-IV	
VDSR	79.07 ± 2.22	1.0 ± 0.0002	60.8 ± 1.04	0.9992 ± 0.0
SRGAN	76.18 ± 2.57	1.0 ± 0.0003	57.99 ± 0.9	0.9985 ± 0.0
SRGAN TH	77.3 ± 2.74	1.0 ± 0.0003	59.01 ± 1.05	0.9987 ± 0.0
EDSR	77.89 ± 2.4	1.0 ± 0.0004	57.53 ± 1.1	0.9982 ± 0.0
EDSR TH	77.98 ± 2.36	1.0 ± 0.0004	58.19 ± 1.09	0.9984 ± 0.0
MDSR	79.86 ± 2.23	1.0 ± 0.0001	61.68 ± 1.02	0.9994 ± 0.0
WDSR	77.82 ± 2.33	1.0 ± 0.0004	57.7 ± 1.17	0.9982 ± 0.0
WDSR TH	77.9 ± 2.41	1.0 ± 0.0004	57.75 ± 1.13	0.9983 ± 0.0
ESRGAN	69.72 ± 2.99	0.9999 ± 0.0001	49.02 ± 0.77	0.9865 ± 0.0023
Real-ESRGAN	74.85 ± 4.73	0.9999 ± 0.0004	55.98 ± 0.64	0.9976 ± 0.0001
Método	PSNR	SSIM	PSNR	SSIM
Métricas - Escala x4				

Tabla 6.6: Resultados del experimento 2 - Métodos de Python en escala x4

A continuación, se representarán los resultados de PSNR obtenidos al procesar los datos de este experimento. En este caso se aportarán dos gráficas distintas, una para la escala x2 y otra para la escala x4. Estos datos se ilustran respectivamente en la Figura 6.3 y en la Figura 6.4.

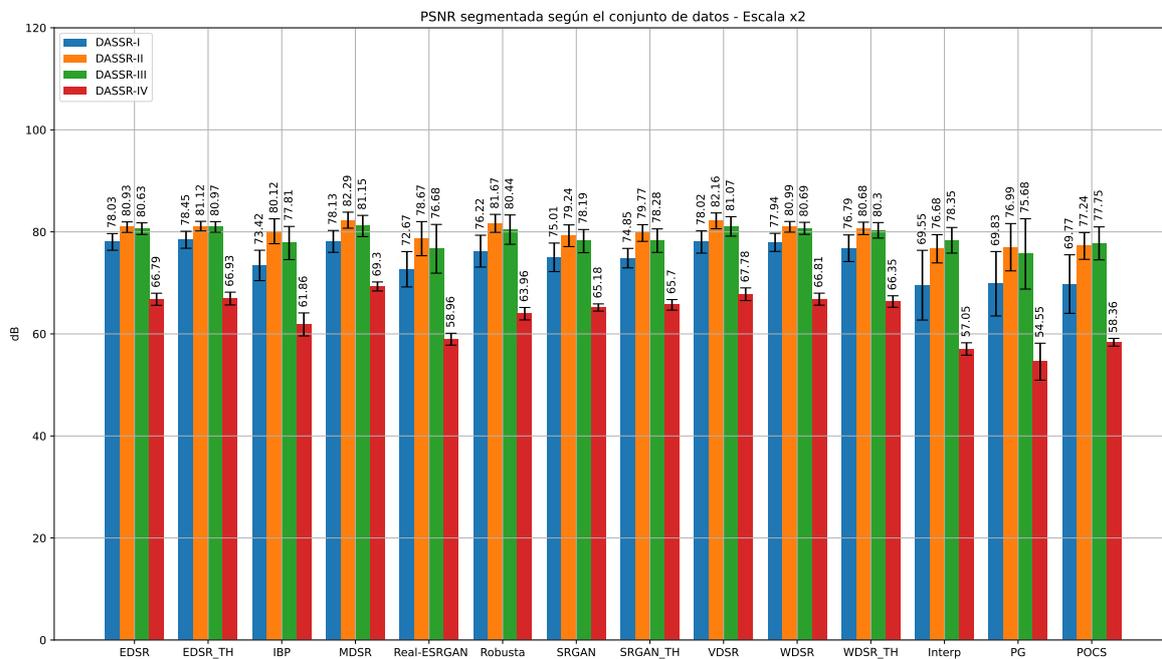


Figura 6.3: Representación gráfica de las Tablas 6.3 y 6.5 - Escala x2

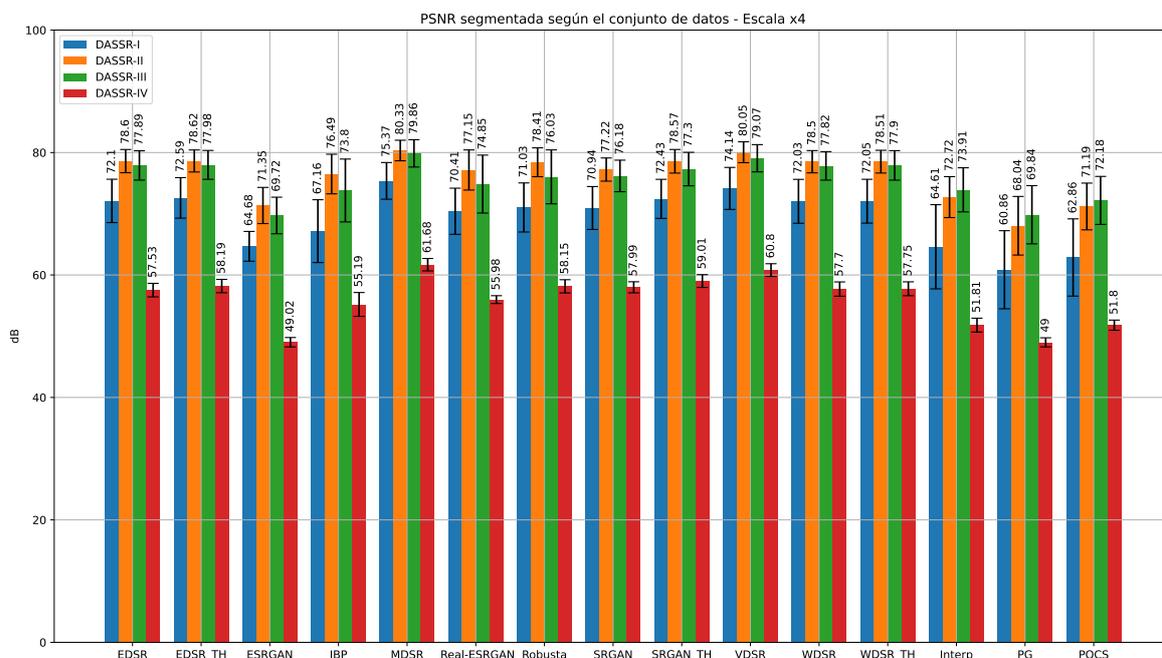


Figura 6.4: Representación gráfica de las Tablas 6.4 y 6.6 - Escala x4

A la vista de los resultados mostrados anteriormente se puede decir que los promedios obtenidos por las redes neuronales de los métodos SRGAN, EDSR y WDSR, entrenadas con la base de datos ADAS, son algo mayores de forma general que las entrenadas con la base de datos DIV2K. No obstante, esta diferencia no es muy elevada.

Por otro lado, como se ha comentado en la sección de resultados del experimento 1, los métodos sombreados en color verde ofrecen los mejores valores de PSNR y SSIM. Sin embargo, en este trabajo se estima que los métodos sombreados en azul son los que mejoran visualmente la imagen degradada. Por ejemplo, en la Figura 6.5 se puede observar los resultados visuales de aplicar el método de súper-resolución espacial basado en interpolación a una imagen de la cámara A615. Este hecho puede ser fácilmente comparado con el método EDSR o Real-ESRGAN, que ofrecen una calidad visual mayor y se corresponden con la Figura 6.6 y la Figura 6.7.

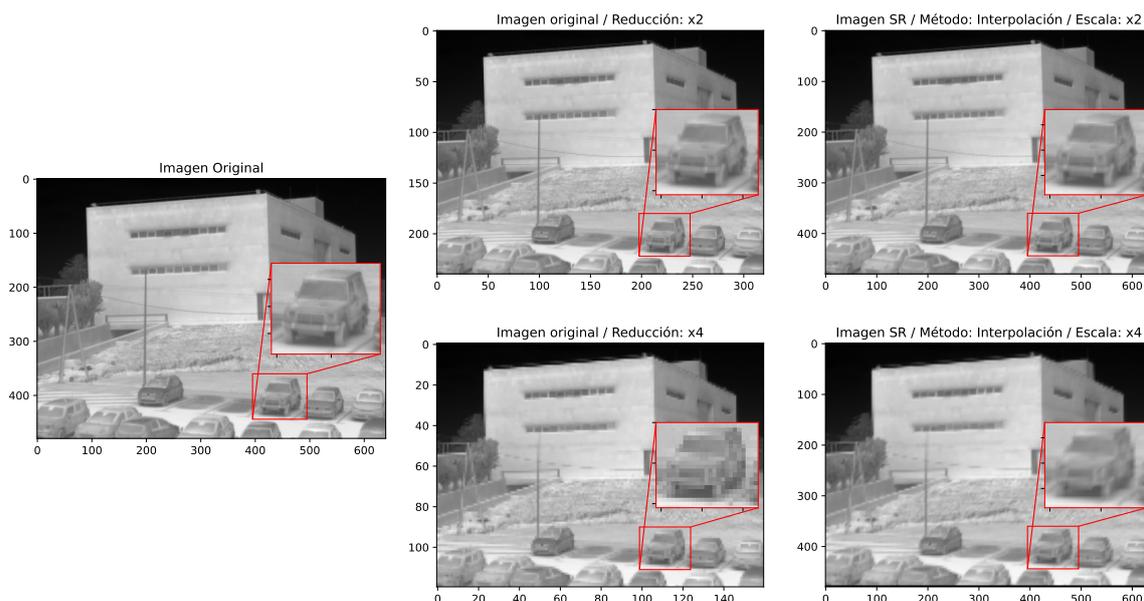


Figura 6.5: Resultados del método de interpolación en la cámara A615

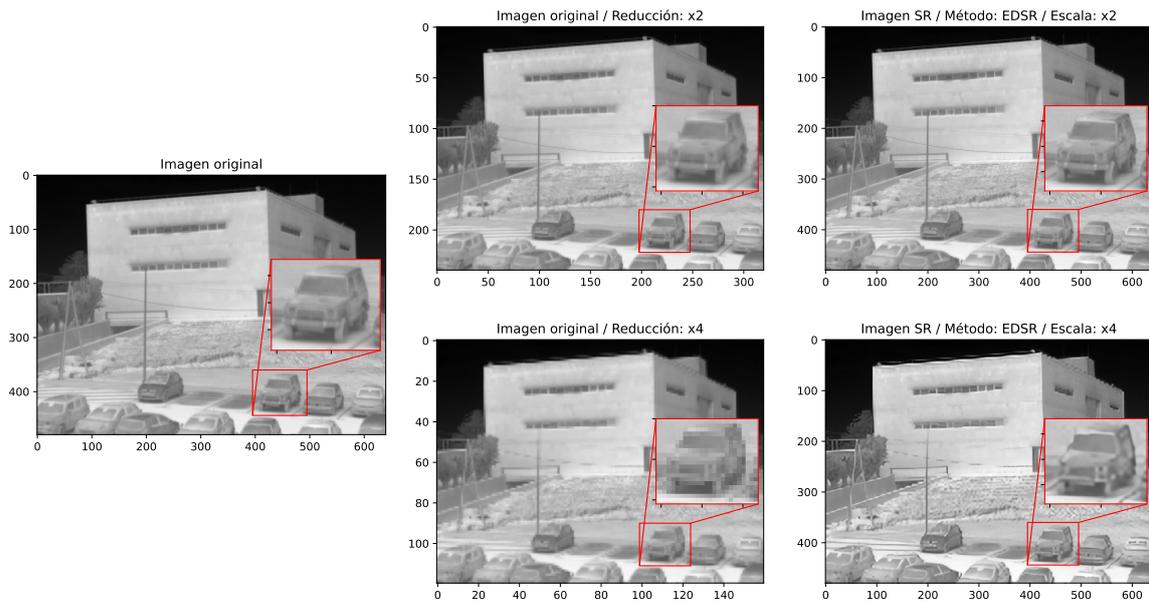


Figura 6.6: Resultados del método EDSR en la cámara A615

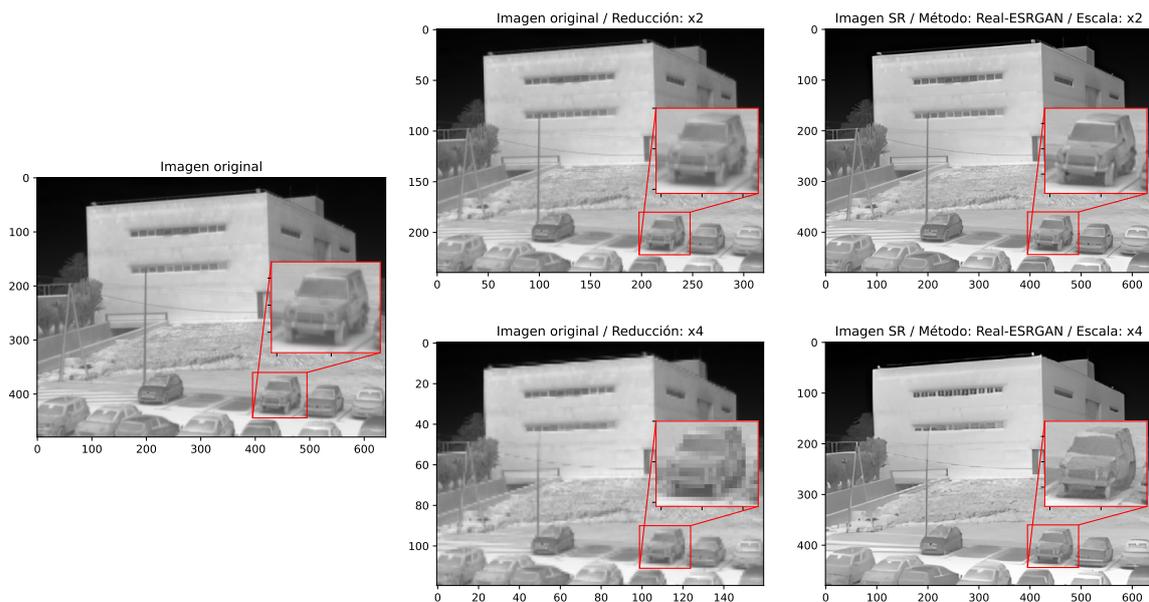


Figura 6.7: Resultados del método Real-ESRGAN en la cámara A615

De forma holgada, para todo el experimento 2, el método MDSR es el que consigue una mayor PSNR para toda la base de datos DASSR. En el anexo A se podrá observar que el resultado visual de aplicar este método a imágenes de la cámara A615 y a imágenes de la cámara Lepton se considera mejor respecto a otros métodos estudiados en este trabajo. No obstante, los métodos EDSR, WDSR y Real-ESRGAN siguen destacando por su mejora en el resultado visual final de la imagen de entrada.

6.3. Resultados del experimento 3

En esta sección se presentan los resultados obtenidos de aplicar los métodos de súper-resolución espacial sobre las imágenes de la cámara A35 y la cámara Lepton de la base de datos DASSR. Para este experimento se han recogido algunas referencias importantes de temperatura tanto antes como después de aplicar cada método, de esta forma se podrá observar que pérdida implícita existe en el propio método sin tener en cuenta ningún proceso de degradación previo.

- El primer valor de las tablas representa la diferencia entre máximos de las imágenes LR y SR. Es decir, se toma un máximo absoluto de la imagen LR, aplicando sobre esta un método de súper-resolución espacial para transformarla en una imagen SR. A continuación, se toma el valor del máximo absoluto de la imagen SR y se le resta al original. Esta medida será de especial importancia para aquellas aplicaciones donde lo que más interese sea conocer cuál es la máxima temperatura de un objeto dado, teniendo en consideración que pérdidas aproximadas se asumen al aplicar ciertos métodos de súper-resolución espacial sobre la imagen.
- El segundo valor que representan las tablas se corresponde con la diferencia promedio entre ambas imágenes. Es decir, se resta directamente todos los píxeles de la imagen SR a la imagen LR, teniendo en cuenta que la imagen LR ha sido interpolada con el mismo factor. Finalmente, se obtendrá una media ponderada de estos valores. Esta métrica es una medida que permitirá conocer la pérdida promedio del método de súper-resolución espacial, ya que se contabilizan todos los píxeles que hay en la imagen. El método de interpolación usado en la imagen LR es la interpolación del vecino más cercano.

Las Tablas 6.7, 6.8, 6.9 y 6.10 ofrecen, para cada método de súper-resolución espacial, las métricas propuestas en el caso de la cámara Lepton. De esta manera, se puede observar que no existe una pérdida alta en la columna que ofrece información sobre la diferencia de valores máximos de cada imagen. No obstante, para algunos métodos como es el caso del método de Papoulis-Gerchberg o el método POCS, la pérdida general de la matriz de datos si es más alta, sobre todo cuando se estudian mayores factores de escala.

Métricas - Escala x2				
Método	Máximo (°C)	Promedio (°C)	Máximo (°C)	Promedio (°C)
Interpolación	0.19 ± 0.17	0.1 ± 0.03	0.21 ± 0.25	0.05 ± 0.01
IBP	0.19 ± 0.17	0.13 ± 0.06	0.21 ± 0.25	0.08 ± 0.01
PG	0.19 ± 0.17	0.22 ± 0.15	0.21 ± 0.25	0.08 ± 0.04
POCS	0.19 ± 0.17	0.15 ± 0.05	0.21 ± 0.25	0.06 ± 0.02
SR Robusta	0.19 ± 0.17	0.07 ± 0.02	0.21 ± 0.25	0.05 ± 0.01
	DASSR-I		DASSR-II	
	Conjunto de datos			
	DASSR-III		DASSR-IV	
Interpolación	0.25 ± 0.21	0.07 ± 0.05	0.07 ± 0.03	0.26 ± 0.01
IBP	0.25 ± 0.21	0.12 ± 0.06	0.07 ± 0.03	0.25 ± 0.03
PG	0.25 ± 0.21	0.19 ± 0.29	0.07 ± 0.03	0.88 ± 0.31
POCS	0.25 ± 0.21	0.16 ± 0.17	0.07 ± 0.03	0.31 ± 0.01
SR Robusta	0.25 ± 0.21	0.07 ± 0.03	0.07 ± 0.03	0.19 ± 0.01
Método	Máximo (°C)	Promedio (°C)	Máximo (°C)	Promedio (°C)
Métricas - Escala x2				

Tabla 6.7: Resultados del experimento 3 (I) - Métodos de MATLAB en escala x2

Métricas - Escala x4				
Método	Máximo (°C)	Promedio (°C)	Máximo (°C)	Promedio (°C)
Interpolación	0.1 ± 0.09	0.13 ± 0.05	0.1 ± 0.12	0.06 ± 0.01
IBP	0.1 ± 0.09	0.13 ± 0.05	0.1 ± 0.12	0.07 ± 0.02
PG	0.1 ± 0.09	0.93 ± 0.56	0.1 ± 0.12	0.26 ± 0.12
POCS	0.1 ± 0.09	0.48 ± 0.25	0.1 ± 0.12	0.15 ± 0.05
SR Robusta	0.1 ± 0.09	0.07 ± 0.02	0.1 ± 0.12	0.05 ± 0.01
	DASSR-I		DASSR-II	
	Conjunto de datos			
	DASSR-III		DASSR-IV	
Interpolación	0.24 ± 0.03	0.1 ± 0.07	0.13 ± 0.11	0.38 ± 0.01
IBP	0.13 ± 0.11	0.12 ± 0.06	0.04 ± 0.02	0.25 ± 0.03
PG	0.13 ± 0.11	0.64 ± 0.68	0.04 ± 0.02	1.86 ± 0.09
POCS	0.13 ± 0.11	0.37 ± 0.39	0.04 ± 0.02	0.85 ± 0.05
SR Robusta	0.13 ± 0.11	0.06 ± 0.03	0.04 ± 0.02	0.19 ± 0.01
Método	Máximo (°C)	Promedio (°C)	Máximo (°C)	Promedio (°C)
Métricas - Escala x4				

Tabla 6.8: Resultados del experimento 3 (I) - Métodos de MATLAB en escala x4

	Métricas - Escala x2			
Método	Máximo (°C)	Promedio (°C)	Máximo (°C)	Promedio (°C)
VDSR	0.17 ± 0.15	0.06 ± 0.02	0.14 ± 0.15	0.03 ± 0.01
SRGAN	0.27 ± 0.19	0.1 ± 0.02	0.18 ± 0.19	0.08 ± 0.01
SRGAN TH	0.13 ± 0.11	0.12 ± 0.03	0.17 ± 0.22	0.07 ± 0.01
EDSR	0.17 ± 0.09	0.07 ± 0.01	0.16 ± 0.2	0.06 ± 0.0
EDSR TH	0.18 ± 0.17	0.06 ± 0.01	0.2 ± 0.25	0.05 ± 0.0
MDSR	0.09 ± 0.12	0.07 ± 0.02	0.13 ± 0.17	0.03 ± 0.01
WDSR	0.17 ± 0.09	0.07 ± 0.01	0.16 ± 0.2	0.06 ± 0.0
WDSR TH	0.18 ± 0.18	0.08 ± 0.02	0.2 ± 0.25	0.06 ± 0.0
ESRGAN	-----	-----	-----	-----
Real-ESRGAN	0.15 ± 0.15	0.12 ± 0.03	0.2 ± 0.24	0.06 ± 0.03
	DASSR-I		DASSR-II	
	Conjunto de datos			
	DASSR-III		DASSR-IV	
VDSR	0.21 ± 0.2	0.05 ± 0.02	0.05 ± 0.03	0.11 ± 0.0
SRGAN	0.18 ± 0.19	0.1 ± 0.03	0.39 ± 0.05	0.21 ± 0.01
SRGAN TH	0.15 ± 0.12	0.1 ± 0.04	0.1 ± 0.04	0.19 ± 0.0
EDSR	0.14 ± 0.13	0.07 ± 0.02	0.31 ± 0.03	0.14 ± 0.01
EDSR TH	0.22 ± 0.16	0.06 ± 0.01	0.07 ± 0.03	0.13 ± 0.01
MDSR	0.16 ± 0.15	0.06 ± 0.03	0.02 ± 0.02	0.11 ± 0.0
WDSR	0.14 ± 0.13	0.07 ± 0.02	0.31 ± 0.03	0.14 ± 0.01
WDSR TH	0.21 ± 0.17	0.07 ± 0.03	0.07 ± 0.03	0.17 ± 0.01
ESRGAN	-----	-----	-----	-----
Real-ESRGAN	0.24 ± 0.21	0.1 ± 0.11	0.07 ± 0.03	0.68 ± 0.11
Método	Máximo (°C)	Promedio (°C)	Máximo (°C)	Promedio (°C)
	Métricas - Escala x2			

Tabla 6.9: Resultados del experimento 3 (I) - Métodos de Python en escala x2

	Métricas - Escala x4			
Método	Máximo (°C)	Promedio (°C)	Máximo (°C)	Promedio (°C)
VDSR	0.11 ± 0.08	0.05 ± 0.01	0.09 ± 0.1	0.03 ± 0.01
SRGAN	0.24 ± 0.17	0.15 ± 0.05	0.12 ± 0.11	0.08 ± 0.01
SRGAN TH	0.1 ± 0.08	0.1 ± 0.02	0.08 ± 0.1	0.06 ± 0.01
EDSR	0.17 ± 0.11	0.07 ± 0.01	0.09 ± 0.07	0.06 ± 0.0
EDSR TH	0.1 ± 0.09	0.07 ± 0.01	0.1 ± 0.12	0.06 ± 0.0
MDSR	0.04 ± 0.05	0.07 ± 0.02	0.08 ± 0.09	0.03 ± 0.01
WDSR	0.17 ± 0.11	0.07 ± 0.01	0.09 ± 0.07	0.06 ± 0.0
WDSR TH	0.1 ± 0.09	0.07 ± 0.01	0.1 ± 0.12	0.06 ± 0.0
ESRGAN	0.1 ± 0.09	0.32 ± 0.06	0.11 ± 0.12	0.21 ± 0.04
Real-ESRGAN	0.1 ± 0.09	0.12 ± 0.03	0.1 ± 0.13	0.15 ± 0.09
	DASSR-I		DASSR-II	
	Conjunto de datos			
	DASSR-III		DASSR-IV	
VDSR	0.12 ± 0.11	0.04 ± 0.02	0.04 ± 0.02	0.09 ± 0.0
SRGAN	0.22 ± 0.22	0.14 ± 0.09	0.22 ± 0.05	0.23 ± 0.01
SRGAN TH	0.08 ± 0.07	0.09 ± 0.04	0.15 ± 0.04	0.18 ± 0.0
EDSR	0.1 ± 0.11	0.07 ± 0.02	0.34 ± 0.02	0.14 ± 0.01
EDSR TH	0.12 ± 0.1	0.07 ± 0.02	0.04 ± 0.02	0.13 ± 0.0
MDSR	0.07 ± 0.06	0.05 ± 0.04	0.06 ± 0.03	0.15 ± 0.0
WDSR	0.1 ± 0.11	0.07 ± 0.02	0.34 ± 0.02	0.14 ± 0.0
WDSR TH	0.13 ± 0.1	0.07 ± 0.02	0.04 ± 0.02	0.14 ± 0.01
ESRGAN	0.13 ± 0.11	0.27 ± 0.09	0.04 ± 0.02	0.44 ± 0.03
Real-ESRGAN	0.13 ± 0.11	0.13 ± 0.09	0.04 ± 0.02	0.51 ± 0.04
Método	Máximo (°C)	Promedio (°C)	Máximo (°C)	Promedio (°C)
	Métricas - Escala x4			

Tabla 6.10: Resultados del experimento 3 (I) - Métodos de Python en escala x4

Para la presentación de resultados de este experimento se visualizará en 4 gráficas distintas los valores mencionados en las tablas anteriores, así como las respectivas escalas de donde fueron obtenidos. Los resultados que ofrecen una lectura de las columnas “Máximo” de las tablas anteriores se muestran en las Figuras 6.8 y 6.9.

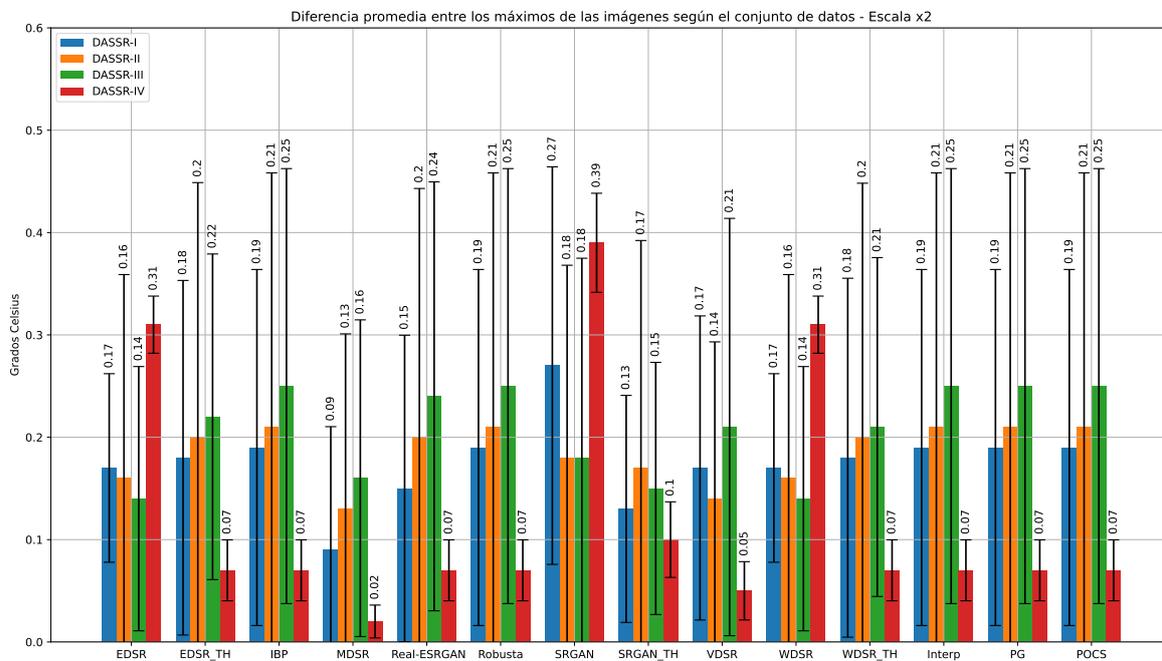


Figura 6.8: Representación gráfica de las Tablas 6.7 y 6.9 - Escala x2 (I)

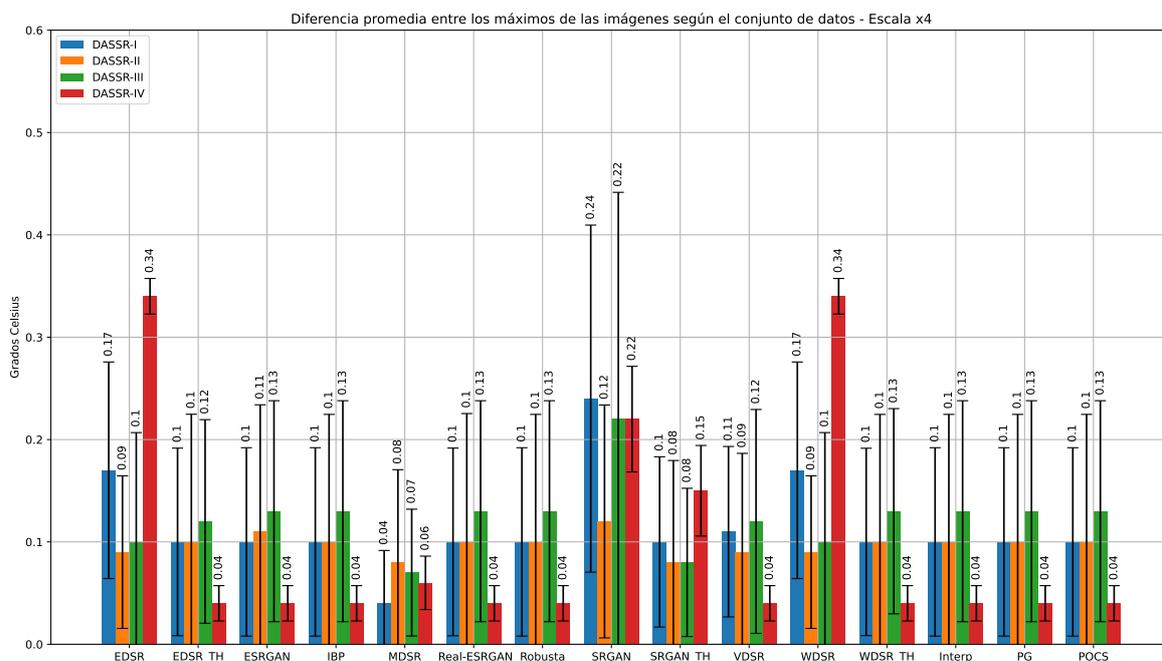


Figura 6.9: Representación gráfica de las Tablas 6.8 y 6.10 - Escala x4 (I)

Los resultados que ofrecen una lectura de las columnas “Promedio” de las tablas anteriores se muestran en las Figuras 6.10 y 6.11.

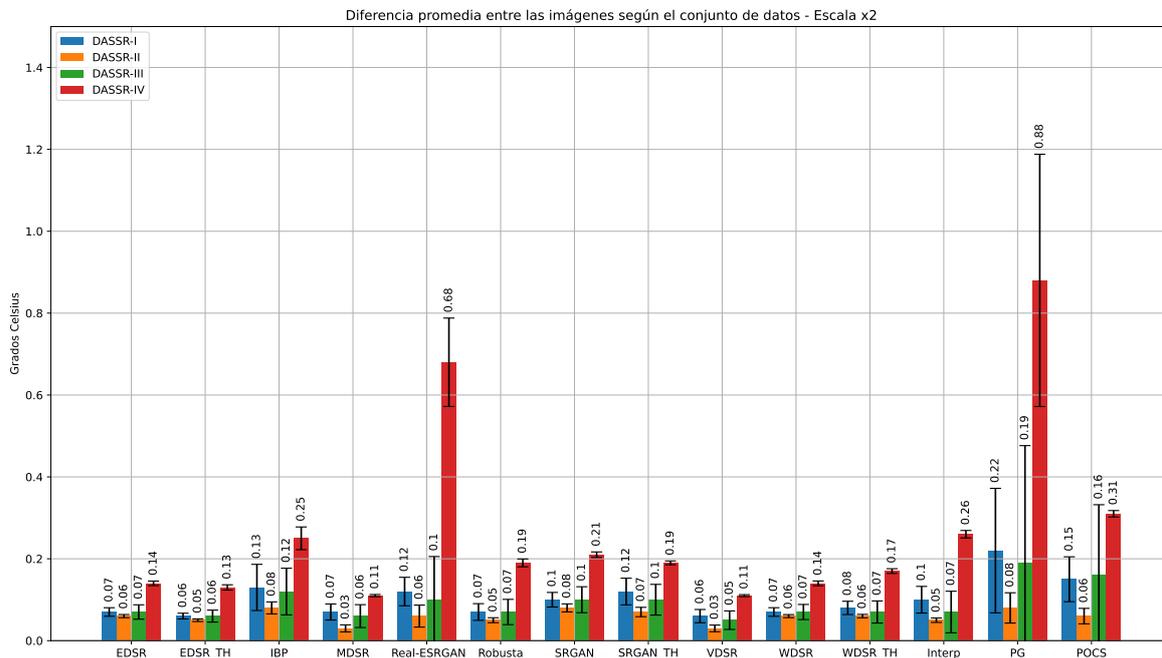


Figura 6.10: Representación gráfica de las Tablas 6.7 y 6.9 - Escala x2 (II)

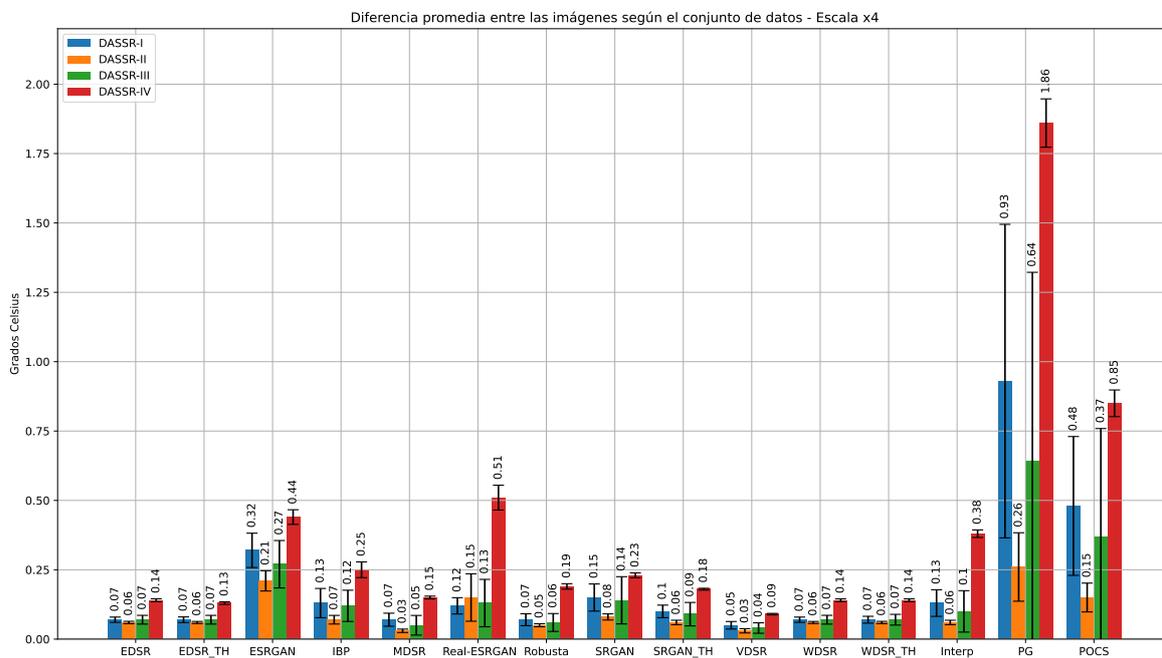


Figura 6.11: Representación gráfica de las Tablas 6.8 y 6.10 - Escala x4 (II)

A raíz de los resultados anteriores se debe resaltar que las redes neuronales de los métodos SRGAN, EDSR y WDSR, entrenadas con la base de datos ADAS, ofrecen unos valores similares a sus variantes entrenadas con la base de datos DIV2K.

Por otro lado, al igual que en el experimento anterior, es posible comparar la calidad visual resultante de aplicar distintos métodos de súper-resolución espacial, esta vez a imágenes de la cámara Lepton. La Figura 6.12 muestra el resultado de aplicar el método de súper-resolución espacial basado en interpolación, mientras que la Figura 6.13 y la Figura 6.14 muestran respectivamente el método EDSR y Real-ESRGAN.

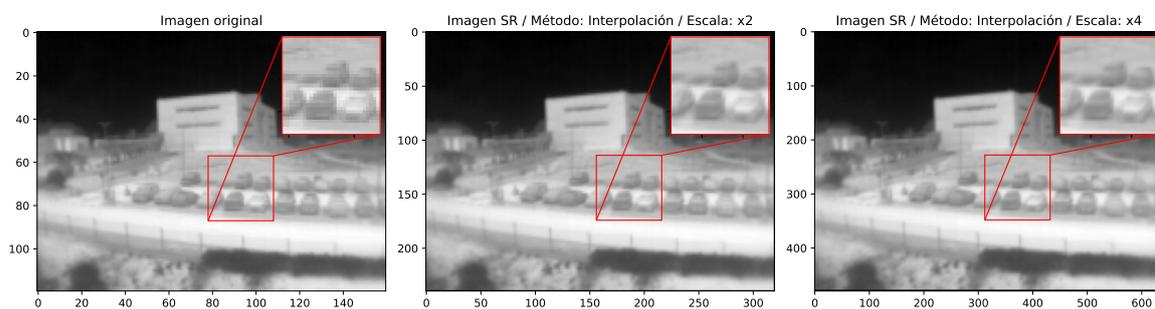


Figura 6.12: Resultados del método de interpolación en la cámara Lepton

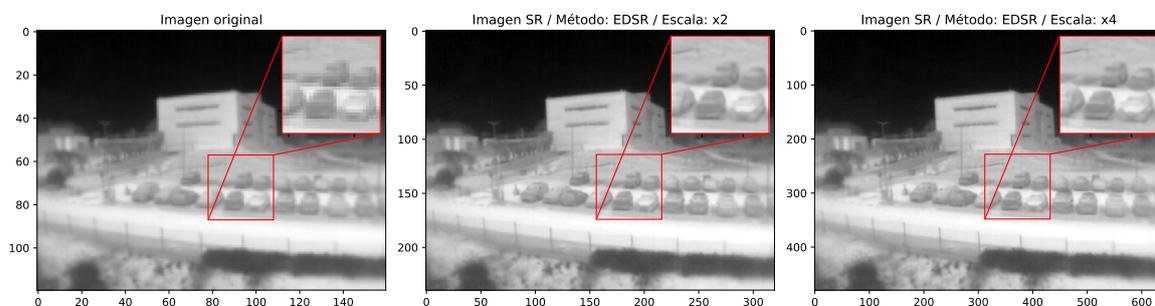


Figura 6.13: Resultados del método EDSR en la cámara Lepton

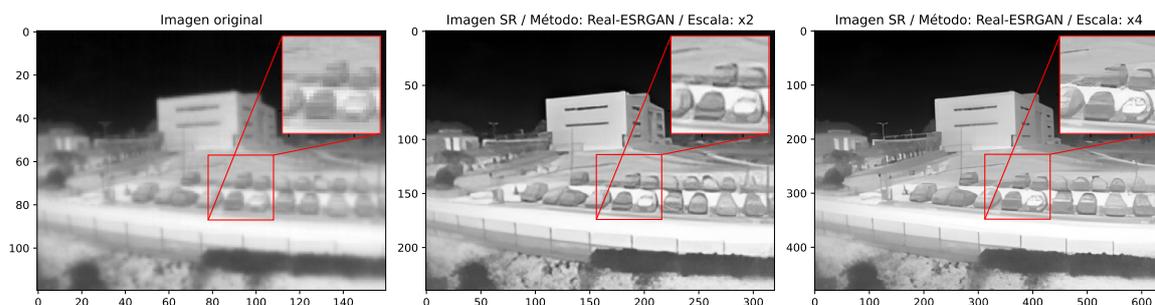


Figura 6.14: Resultados del método Real-ESRGAN en la cámara Lepton

Una de las limitaciones actuales de los métodos de súper-resolución espacial basados en redes GAN se puede ver en la Figura 6.14, específicamente en la escala x4. El generador intenta predecir la imagen SR a partir de la cantidad limitada de píxeles que forma la imagen LR original, por este motivo el resultado final puede no identificarse completamente con el objeto real que se está estudiando. Este método presenta un mejor desempeño para imágenes con un mayor enfoque de la escena y, por lo tanto, una mayor distribución de píxeles para conformar ciertos objetos de esta. Un ejemplo de este hecho se puede observar comparando la figura anterior con la Figura 6.7.

Las Tablas 6.11 y 6.12 ofrecen, para cada método de súper-resolución espacial, las métricas propuestas en el caso de la cámara A35. Al contrario que en casos anteriores, en esta parte del experimento sólo se realizaron pruebas con el factor de escala x2.

Métricas - Escala x2				
Método	Máximo (°C)	Promedio (°C)	Máximo (°C)	Promedio (°C)
Interpolación	0.12 ± 0.07	0.06 ± 0.02	0.11 ± 0.08	0.03 ± 0.0
IBP	0.12 ± 0.07	0.1 ± 0.04	0.11 ± 0.08	0.07 ± 0.04
PG	0.12 ± 0.07	0.2 ± 0.18	0.11 ± 0.08	0.08 ± 0.11
POCS	0.12 ± 0.07	0.1 ± 0.04	0.11 ± 0.08	0.05 ± 0.01
SR Robusta	0.12 ± 0.07	0.06 ± 0.02	0.11 ± 0.08	0.04 ± 0.01
	DASSR-I		DASSR-II	
	Conjunto de datos			
	DASSR-III		DASSR-IV	
Interpolación	0.14 ± 0.1	0.04 ± 0.01	0.13 ± 0.1	0.17 ± 0.02
IBP	0.14 ± 0.1	0.08 ± 0.04	0.13 ± 0.1	0.16 ± 0.02
PG	0.14 ± 0.1	0.09 ± 0.07	0.13 ± 0.1	1.13 ± 0.35
POCS	0.14 ± 0.1	0.07 ± 0.04	0.13 ± 0.1	0.21 ± 0.02
SR Robusta	0.14 ± 0.1	0.05 ± 0.01	0.13 ± 0.1	0.13 ± 0.02
Método	Máximo (°C)	Promedio (°C)	Máximo (°C)	Promedio (°C)
Métricas - Escala x2				

Tabla 6.11: Resultados del experimento 3 (II) - Métodos de MATLAB en escala x2

	Métricas - Escala x2			
Método	Máximo (°C)	Promedio (°C)	Máximo (°C)	Promedio (°C)
VDSR	0.1 ± 0.08	0.05 ± 0.02	0.07 ± 0.05	0.03 ± 0.01
SRGAN	0.24 ± 0.16	0.1 ± 0.02	0.04 ± 0.05	0.06 ± 0.01
SRGAN TH	0.05 ± 0.04	0.1 ± 0.02	0.07 ± 0.05	0.06 ± 0.01
EDSR	0.14 ± 0.09	0.07 ± 0.01	0.04 ± 0.05	0.05 ± 0.01
EDSR TH	0.12 ± 0.07	0.06 ± 0.01	0.11 ± 0.07	0.05 ± 0.0
MDSR	0.05 ± 0.06	0.06 ± 0.02	0.05 ± 0.05	0.03 ± 0.01
WDSR	0.14 ± 0.09	0.07 ± 0.01	0.04 ± 0.05	0.05 ± 0.01
WDSR TH	0.12 ± 0.07	0.08 ± 0.02	0.11 ± 0.06	0.05 ± 0.01
ESRGAN	-----	-----	-----	-----
Real-ESRGAN	0.11 ± 0.07	0.09 ± 0.03	0.09 ± 0.08	0.06 ± 0.03
	DASSR-I		DASSR-II	
	Conjunto de datos			
	DASSR-III		DASSR-IV	
VDSR	0.1 ± 0.08	0.04 ± 0.02	0.11 ± 0.08	0.1 ± 0.01
SRGAN	0.15 ± 0.12	0.08 ± 0.02	0.42 ± 0.07	0.19 ± 0.01
SRGAN TH	0.07 ± 0.08	0.08 ± 0.03	0.09 ± 0.06	0.17 ± 0.01
EDSR	0.1 ± 0.09	0.06 ± 0.01	0.28 ± 0.09	0.11 ± 0.01
EDSR TH	0.14 ± 0.1	0.05 ± 0.0	0.13 ± 0.1	0.1 ± 0.01
MDSR	0.07 ± 0.06	0.04 ± 0.02	0.07 ± 0.07	0.11 ± 0.01
WDSR	0.1 ± 0.09	0.06 ± 0.01	0.28 ± 0.09	0.12 ± 0.01
WDSR TH	0.13 ± 0.1	0.06 ± 0.01	0.13 ± 0.1	0.15 ± 0.01
ESRGAN	-----	-----	-----	-----
Real-ESRGAN	0.12 ± 0.08	0.06 ± 0.02	0.13 ± 0.1	0.39 ± 0.11
Método	Máximo (°C)	Promedio (°C)	Máximo (°C)	Promedio (°C)
	Métricas - Escala x2			

Tabla 6.12: Resultados del experimento 3 (II) - Métodos de Python en escala x2

Los resultados de procesar las imágenes de la cámara A35 en este experimento se pueden observar a través de las Figuras 6.15 y 6.16.

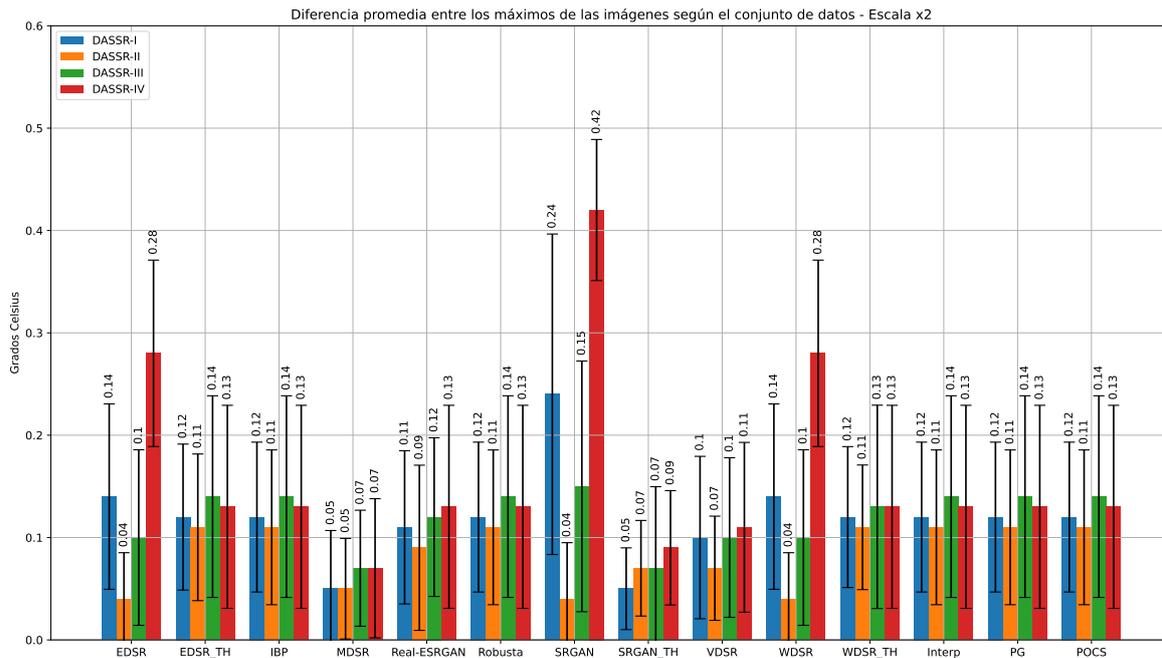


Figura 6.15: Representación gráfica de las Tablas 6.11 y 6.12 - Escala x2 (I)

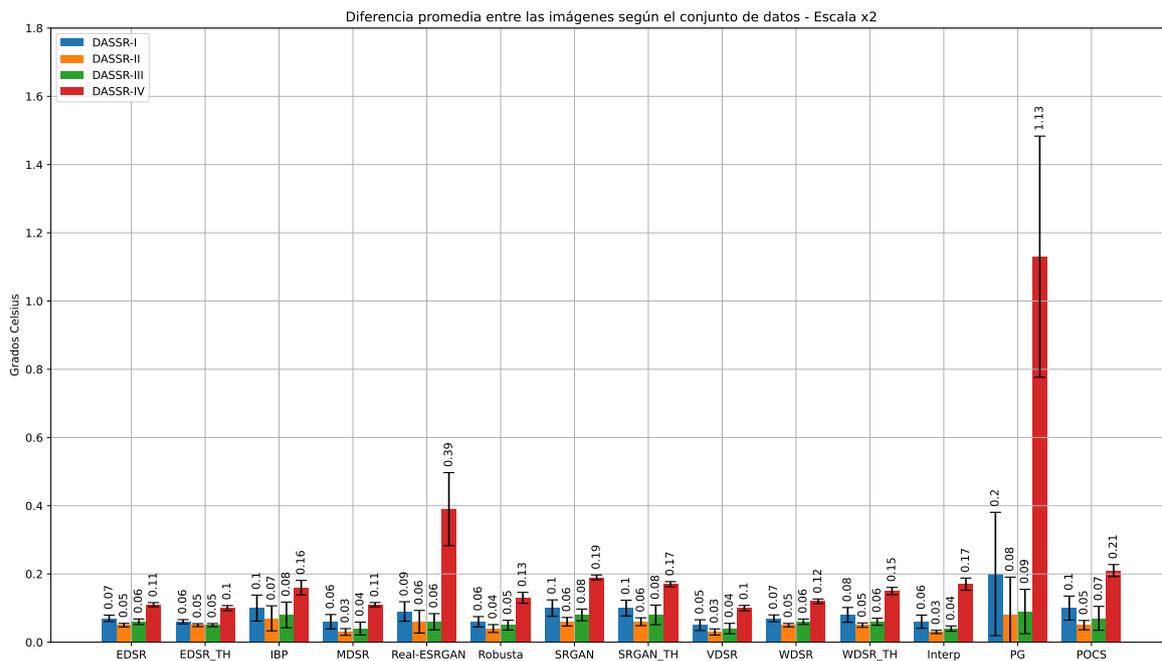


Figura 6.16: Representación gráfica de las Tablas 6.11 y 6.12 - Escala x2 (II)

Al igual que en el experimento anterior, el método MDSR consigue mantener unas pérdidas menores durante el proceso de súper-resolución espacial. En este caso, las pérdidas han sido contabilizadas a través de las métricas propuestas en formato de temperatura.

6.4. Resultados del experimento 4

En esta sección se presentan los resultados obtenidos de aplicar el método propuesto de súper-resolución radiométrica a la base de datos TID-II. Durante la presentación de resultados se mostrará una serie de gráficas y tablas donde se hará referencia a valores de las cámaras A615, A35 y Lepton.

En la Tabla 6.13 se representan dos tipos de métrica. La primera muestra la diferencia absoluta promediada que existe entre una muestra de la cámara Lepton y una muestra de la cámara A615, mientras que la segunda se define como el error promediado en porcentaje. Como se puede observar en esta tabla, la diferencia aumenta de forma general cuánto mayor es la distancia entre las cámaras y la superficie estudiada, debido fundamentalmente a que la cámara Lepton se encuentra mucho más limitada que la cámara A615. Por otro lado, se han añadido a esta misma tabla los resultados obtenidos al procesar las muestras de la cámara Lepton con el modelo de red neuronal planteado. El MAE obtenido originalmente se representa sombreado de naranja, mientras que el MAE obtenido después de aplicar el modelo se ha sombreado con un color azul.

Distancia	Original		Predicción	
	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)
2 metros	41.45 ± 0.45	11.76 ± 0.15	2.53 ± 0.71	0.65 ± 0.19
4 metros	52.77 ± 0.96	15.77 ± 0.15	3.09 ± 0.52	0.8 ± 0.13
6 metros	65.51 ± 1.3	20.44 ± 0.21	1.68 ± 0.78	0.43 ± 0.2
8 metros	75.15 ± 1.35	24.19 ± 0.17	4.12 ± 0.42	1.06 ± 0.11
10 metros	93.76 ± 1.64	32.46 ± 0.22	7.01 ± 0.68	1.87 ± 0.17
12 metros	103.45 ± 1.74	37.52 ± 0.52	5.97 ± 1.71	1.6 ± 0.45
14 metros	130.1 ± 3.3	53.13 ± 1.39	17.98 ± 3.38	5.04 ± 0.97
16 metros	139.48 ± 3.21	60.0 ± 1.66	16.3 ± 3.79	4.59 ± 1.11
18 metros	152.33 ± 2.97	68.6 ± 1.23	15.91 ± 2.62	4.44 ± 0.74

Tabla 6.13: Resultados del experimento 4 - Distancias: 2 a 18 metros

Tanto en el caso de la métrica MAE como en el caso de la métrica MAPE se observa una mejora, no sólo en la diferencia absoluta sino también en su desviación típica.

CAPÍTULO 6. RESULTADOS

En la Figura 6.17 se muestra una gráfica de barras que representa visualmente los valores de la tabla anterior, pero que ayuda a contemplar de forma intuitiva como aumenta el MAE a medida que aumenta también la distancia.

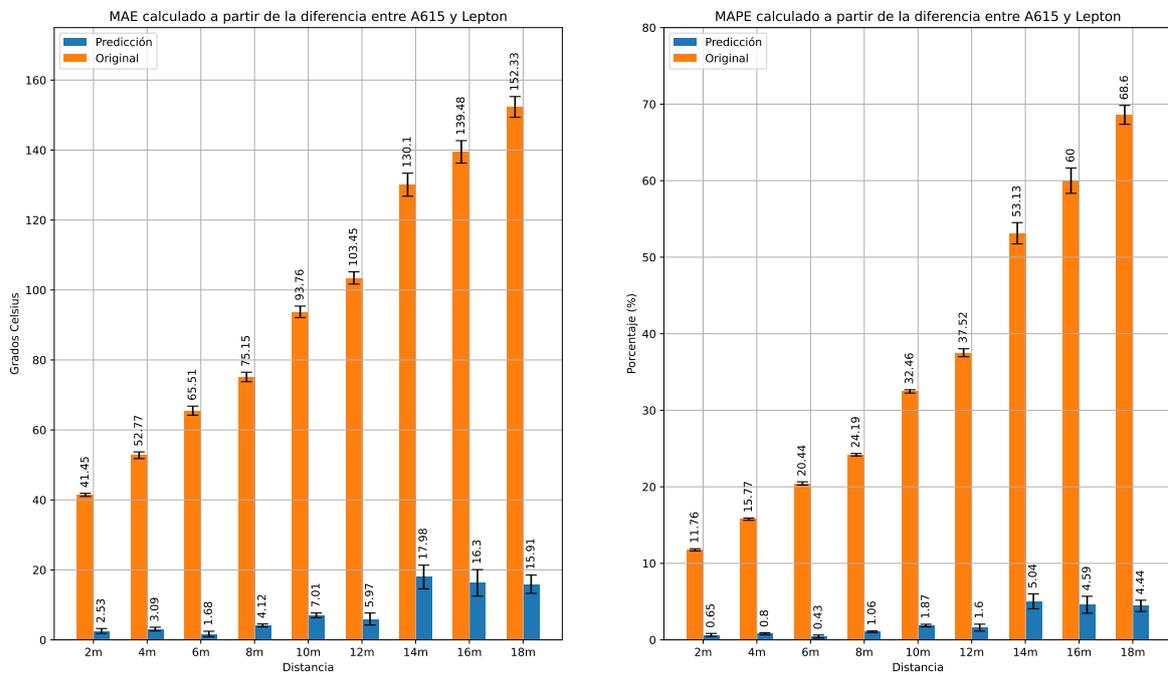


Figura 6.17: Representación gráfica de la Tabla 6.13

Por otro lado, en la Figura 6.18 se presenta una gráfica lineal con los datos utilizados a la hora de generar los valores promediados de la tabla y la gráfica anterior.

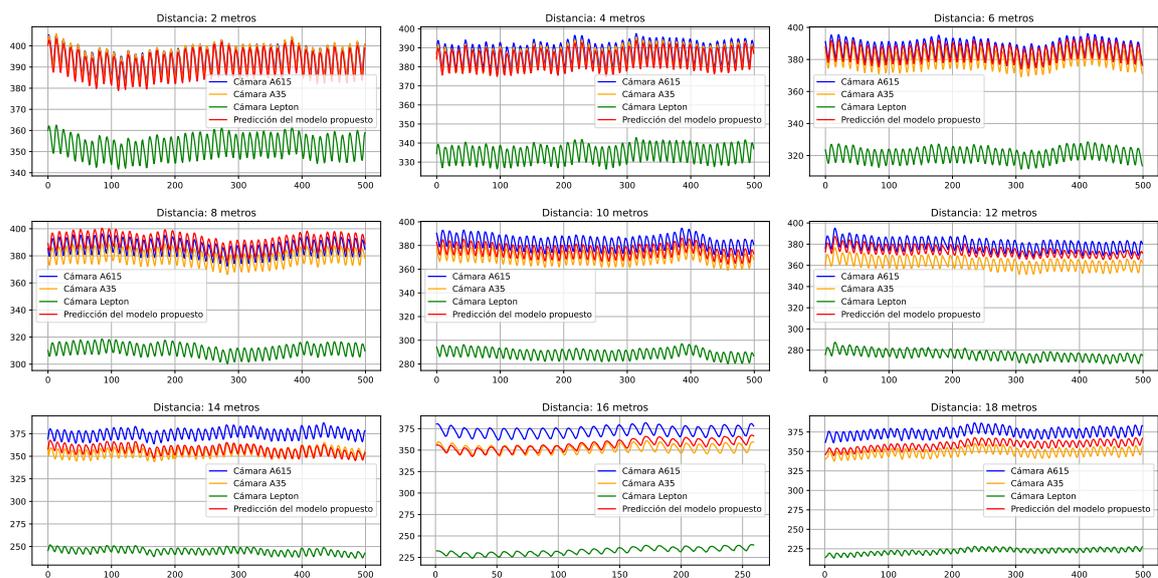


Figura 6.18: Muestras evaluadas del experimento 4 - Distancias: 2 a 18 metros

La Tabla 6.14 muestra las métricas anteriores para distancias impares que van de 3 a 19 metros. Debajo de esta tabla, en la Figura 6.19, se muestra una representación gráfica de los datos promediados. Por otro lado, la Figura 6.20 se corresponde con los valores estudiados según la distancia de medida.

Distancia	Original		Predicción	
	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)
3 metros	22.06 ± 12.22	14.45 ± 0.99	1.78 ± 2.36	0.91 ± 0.89
5 metros	27.49 ± 15.15	18.11 ± 1.38	2.48 ± 2.53	1.46 ± 1.04
7 metros	40.74 ± 22.53	22.76 ± 2.29	4.61 ± 3.21	2.39 ± 1.38
9 metros	40.06 ± 22.24	28.63 ± 2.68	3.07 ± 3.03	1.91 ± 1.33
11 metros	46.11 ± 27.53	33.54 ± 4.88	5.18 ± 4.47	3.11 ± 2.02
13 metros	53.49 ± 33.44	40.38 ± 7.16	9.37 ± 6.67	5.46 ± 2.57
15 metros	55.71 ± 31.66	43.38 ± 4.61	12.78 ± 7.08	7.85 ± 4.64
17 metros	64.3 ± 35.28	53.05 ± 4.58	12.08 ± 9.84	6.78 ± 5.18
19 metros	75.49 ± 41.23	72.14 ± 6.86	11.37 ± 8.11	6.75 ± 3.94

Tabla 6.14: Resultados del experimento 4 - Distancias: 3 a 19 metros

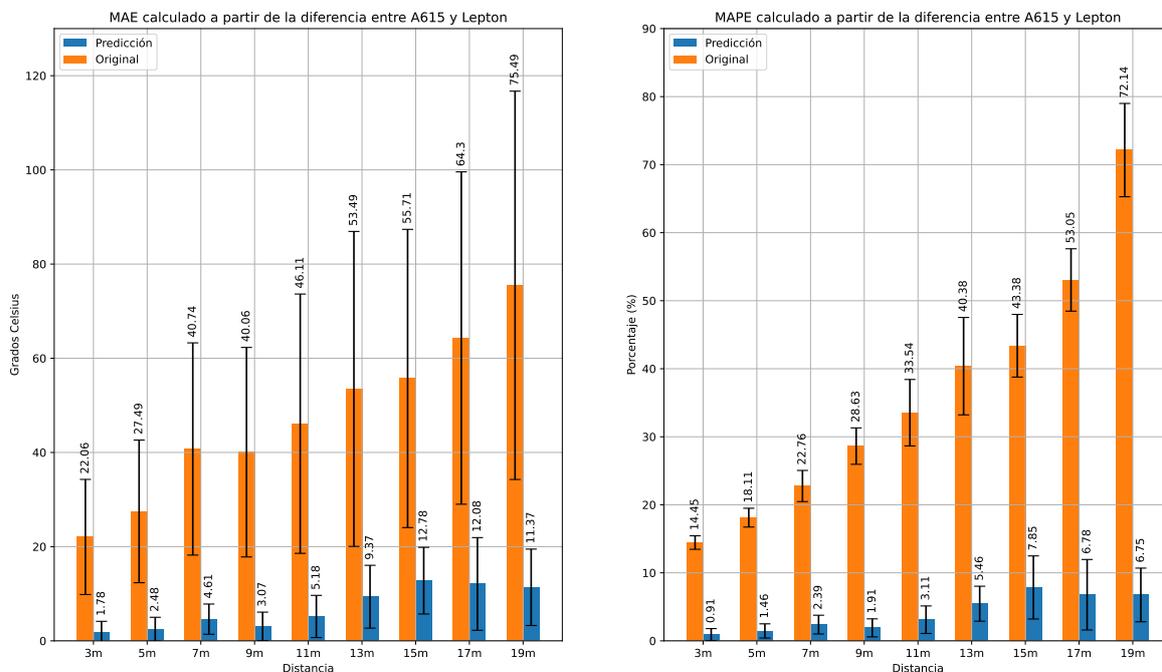


Figura 6.19: Representación gráfica de la Tabla 6.14

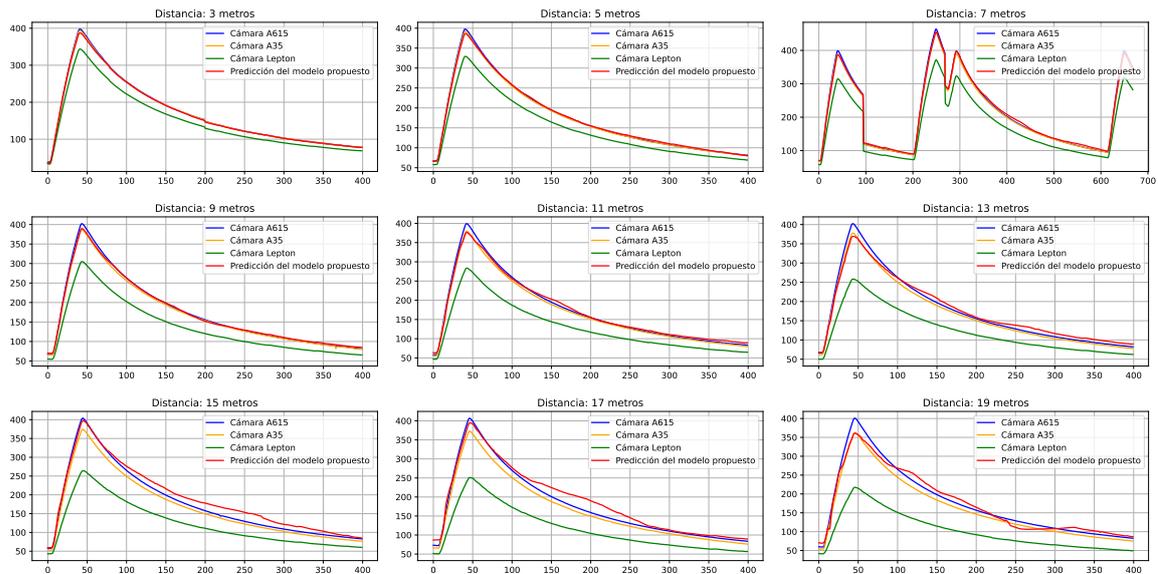


Figura 6.20: Muestras evaluadas del experimento 4 - Distancias: 3 a 19 metros

La Tabla 6.15 muestra las métricas anteriores para distancias impares que van de 21 a 31 metros. Debajo de esta tabla, en la Figura 6.21, se muestra una representación gráfica de los datos promediados. Por otro lado, la Figura 6.22 se corresponde con los valores estudiados según la distancia de medida.

Distancia	Original		Predicción	
	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)
21 metros	135.74 ± 81.45	93.61 ± 22.81	41.02 ± 30.8	15.38 ± 6.94
23 metros	134.47 ± 52.32	106.02 ± 12.59	19.37 ± 22.83	6.94 ± 7.28
25 metros	137.89 ± 27.45	124.25 ± 7.39	14.56 ± 10.75	5.75 ± 4.37
27 metros	168.27 ± 32.81	143.8 ± 15.46	12.03 ± 8.25	4.2 ± 2.88
29 metros	180.93 ± 56.44	205.44 ± 20.93	34.87 ± 19.82	15.34 ± 7.97
31 metros	179.63 ± 58.58	196.39 ± 23.94	22.43 ± 19.56	8.59 ± 6.35

Tabla 6.15: Resultados del experimento 4 - Distancias: 21 a 23 metros

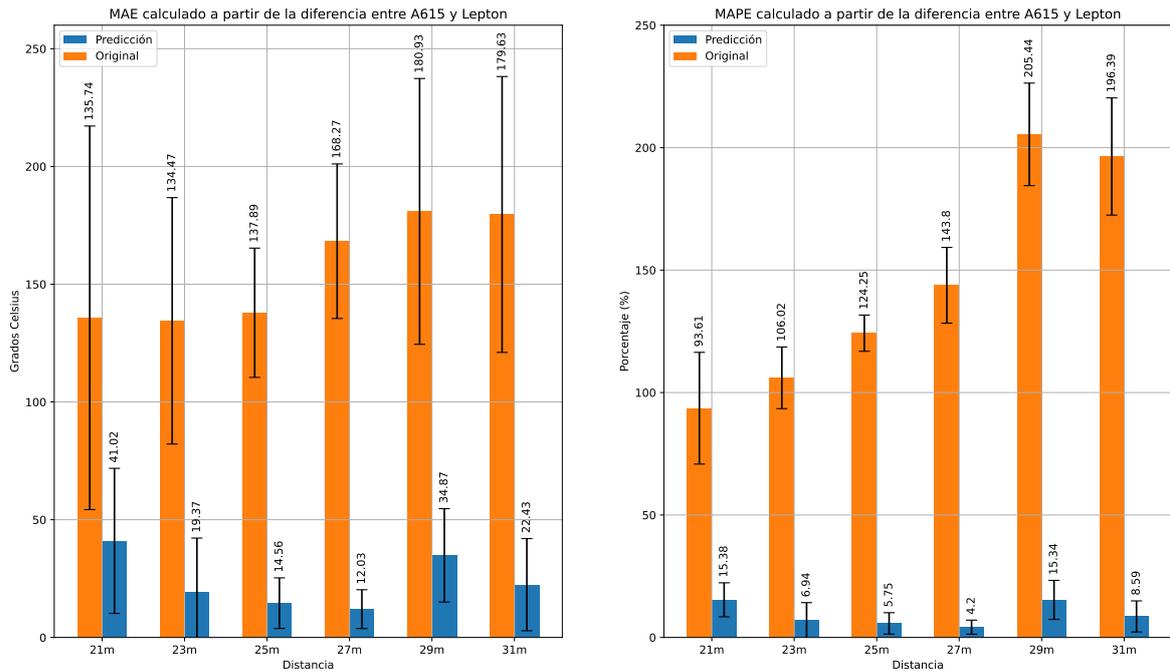


Figura 6.21: Representación gráfica de la Tabla 6.15

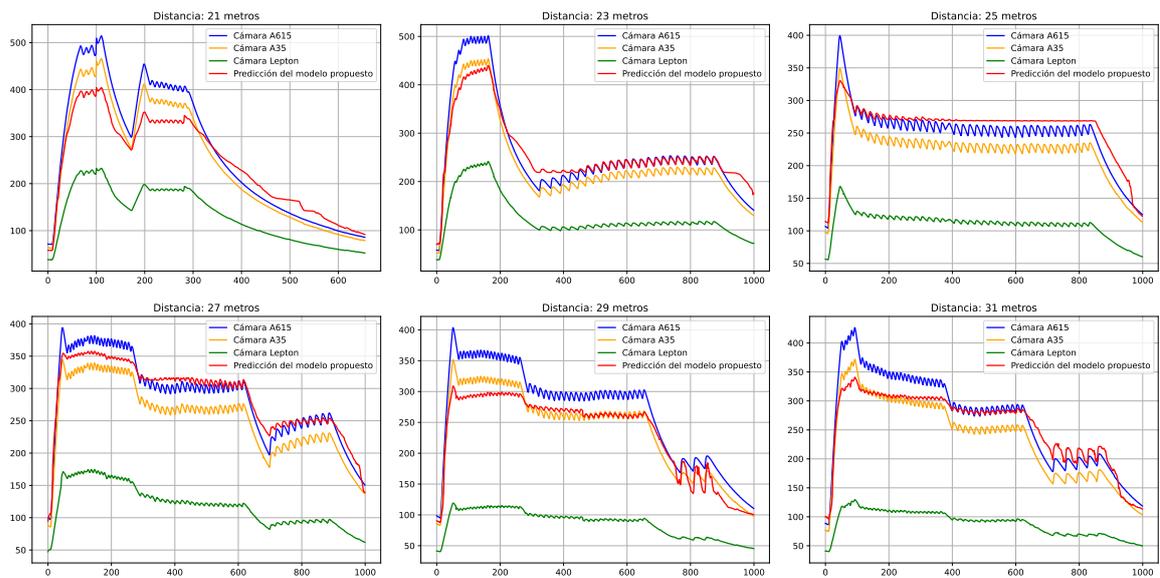


Figura 6.22: Muestras evaluadas del experimento 4 - Distancias: 21 a 31 metros

Según los resultados mostrados en las tablas y gráficas anteriores, es visible que existe una mayor desviación típica en las distancias impares que en las distancias pares, probablemente debido al amplio rango de temperaturas estudiado durante la recolección de datos. Además, se observa que para distancias mayores de 21 metros la predicción llega a superar un MAE de 30 °C, lo que podría denotar que es necesario hacer hincapié en estas distancias durante un futuro entrenamiento de la red neuronal.

6.5. Resultados del experimento 5

Este experimento toma la premisa estudiada en el experimento 4 pero con algunas modificaciones. El objetivo es aplicar un método de súper-resolución espacial a las imágenes de la base de datos TID-II, luego obtener las mismas muestras utilizadas en el experimento 4 y observar cuál ha sido la pérdida promediada para cada distancia de medida. En este caso se ha usado la variante del método EDSR entrenada con la base de datos DIV2K para cada una de las imágenes, ya que en experimentos anteriores se ha visto que la pérdida que introduce este método no es muy alta.

La Tabla 6.16 y la Tabla 6.17 muestran respectivamente que diferencia de grados existe entre los valores de la cámara Lepton y los valores de la cámara A615 para distancias que van de 2 a 18 metros. En la primera tabla se ilustra esta diferencia sin aplicar la red neuronal para la corrección de temperatura, mientras que en la segunda tabla si se aplica esta red neuronal. El MAE obtenido de la imagen original de la cámara Lepton se presenta con un color naranja, mientras que el MAE obtenido después de aplicar el método de súper-resolución espacial es sombreado con un color azul.

Distancia	Valor original del píxel					
	Escala original		Escala x2		Escala x4	
	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)
2 metros	41.45 ± 0.45	11.76 ± 0.15	45.87 ± 0.49	13.18 ± 0.15	45.87 ± 0.49	13.18 ± 0.15
4 metros	52.77 ± 0.96	15.77 ± 0.15	56.94 ± 1.01	17.23 ± 0.15	56.94 ± 1.01	17.23 ± 0.15
6 metros	65.51 ± 1.3	20.44 ± 0.21	69.49 ± 1.35	21.95 ± 0.22	69.49 ± 1.35	21.95 ± 0.22
8 metros	75.15 ± 1.35	24.19 ± 0.17	79.96 ± 1.55	26.14 ± 0.25	79.96 ± 1.55	25.75 ± 0.17
10 metros	93.76 ± 1.64	32.46 ± 0.22	97.33 ± 1.68	34.12 ± 0.22	97.33 ± 1.68	34.12 ± 0.22
12 metros	103.45 ± 1.74	37.52 ± 0.52	106.85 ± 1.78	39.23 ± 0.53	106.85 ± 1.78	39.23 ± 0.53
14 metros	130.1 ± 3.3	53.13 ± 1.39	133.1 ± 3.3	55.02 ± 1.41	133.1 ± 3.3	55.02 ± 1.41
16 metros	139.48 ± 3.21	60.0 ± 1.66	142.31 ± 3.21	61.98 ± 1.67	142.31 ± 3.21	61.98 ± 1.67
18 metros	152.33 ± 2.97	68.6 ± 1.23	155.03 ± 2.99	70.68 ± 1.25	155.03 ± 2.99	70.68 ± 1.25

Tabla 6.16: Resultados del experimento 5 - Distancias: 2 a 18 metros (Valor original)

Distancia	Valor predicho del píxel					
	Escala original		Escala x2		Escala x4	
	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)
2 metros	2.53 ± 0.71	0.65 ± 0.19	7.68 ± 0.63	0.65 ± 0.19	7.68 ± 0.63	1.99 ± 0.19
4 metros	3.09 ± 0.52	0.8 ± 0.13	7.86 ± 0.57	0.8 ± 0.13	7.86 ± 0.57	2.07 ± 0.14
6 metros	1.68 ± 0.78	0.43 ± 0.2	6.26 ± 0.86	0.43 ± 0.2	6.26 ± 0.86	1.65 ± 0.21
8 metros	4.12 ± 0.42	1.06 ± 0.11	2.37 ± 0.64	1.06 ± 0.11	2.37 ± 0.64	0.28 ± 0.11
10 metros	7.01 ± 0.68	1.87 ± 0.17	11.65 ± 0.74	1.87 ± 0.17	11.65 ± 0.74	3.14 ± 0.17
12 metros	5.97 ± 1.71	1.6 ± 0.45	9.29 ± 2.09	1.6 ± 0.45	9.29 ± 2.09	2.51 ± 0.55
14 metros	17.98 ± 3.38	5.04 ± 0.97	22.47 ± 3.35	5.04 ± 0.97	22.47 ± 3.35	6.38 ± 0.98
16 metros	16.3 ± 3.79	5.04 ± 0.97	20.74 ± 3.73	4.59 ± 1.11	20.74 ± 3.73	5.92 ± 1.11
18 metros	15.91 ± 2.62	4.44 ± 0.74	20.03 ± 2.61	4.44 ± 0.74	20.03 ± 2.61	5.92 ± 1.11

Tabla 6.17: Resultados del experimento 5 - Distancias: 2 a 18 metros (Valor predicho)

En esta ocasión se puede observar que al aplicar el método de súper-resolución espacial se obtiene un MAE y un MAPE superior al experimento 4, esto supone una pérdida en los píxeles que se debe tener en cuenta para rangos de temperatura altos. La métrica MAE llega a oscilar entre 1 grado y 6 grados centígrados de diferencia, estas son pérdidas mayores a las observadas en el experimento 3, donde se registraban temperaturas mucho menores. No obstante, en estas tablas se muestra que la escala introducida al método de súper-resolución espacial no aumenta las pérdidas promediadas del MAE, teniendo en cuenta que esta métrica ha sido aplicada a los máximos registrados de la imagen LR y SR.

En la Figura 6.23 se representan visualmente los valores de MAE y MAPE observados en las tablas anteriores. Esta gráfica ha sido dividida en 4 partes. En la parte izquierda se presentan los datos cuando no se ha aplicado el método de súper-resolución espacial, mientras que en la parte derecha se presentan los datos cuando si se ha aplicado. Por otro lado, en la Figura 6.24 se observan los valores usados para la creación de las tablas y la gráfica.

CAPÍTULO 6. RESULTADOS

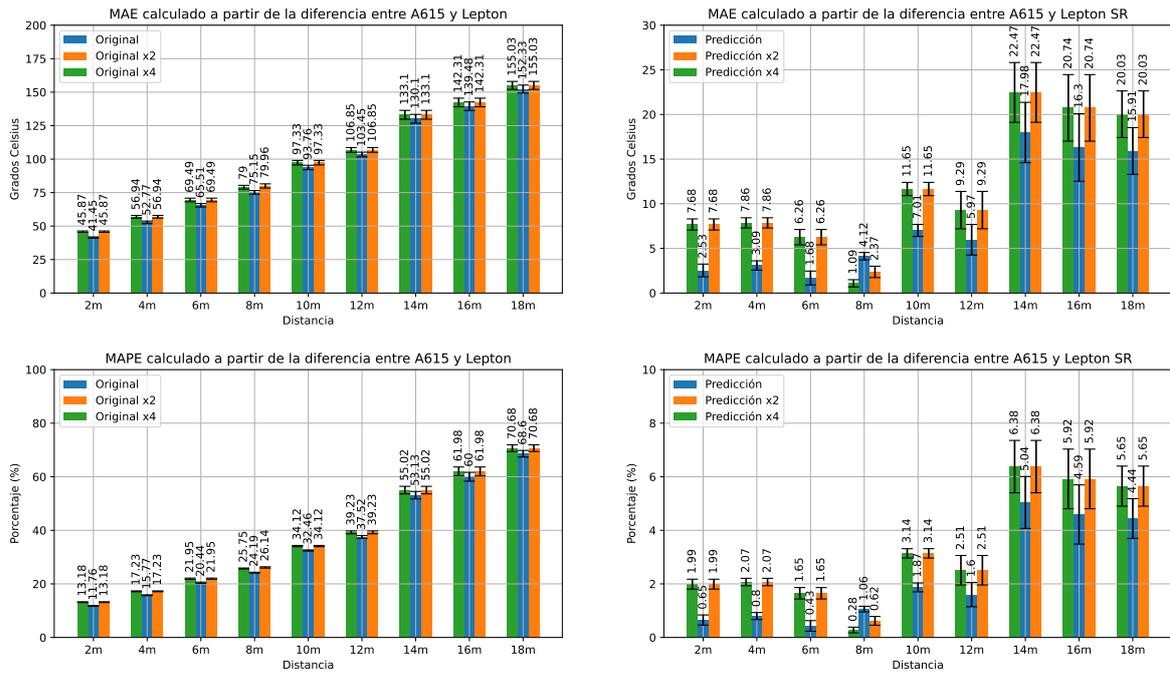


Figura 6.23: Representación gráfica de las Tablas 6.16 y 6.17

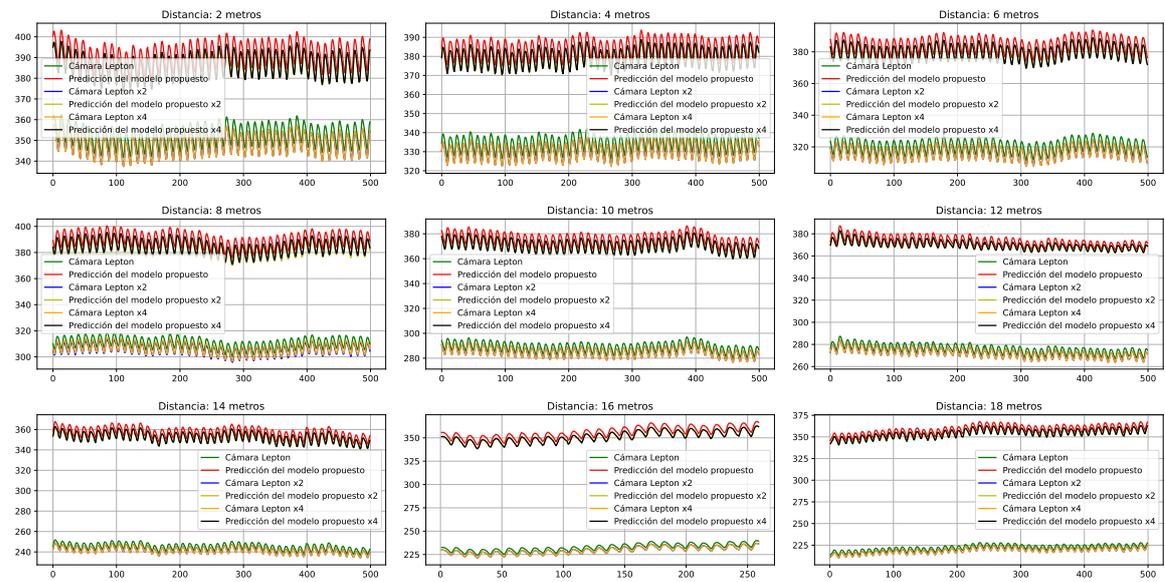


Figura 6.24: Muestras evaluadas del experimento 5 - Distancias: 2 a 18 metros

La Tabla 6.18 y la Tabla 6.19 muestran respectivamente que diferencia de grados centígrados existe entre los valores de la cámara Lepton y los valores de la cámara A615 para las distancias que van de 3 a 19 metros. En la Figura 6.25 se representan visualmente los valores de las tablas y en la Figura 6.26 se presentan los datos usados para la creación de las tablas y la gráfica de barras.

Distancia	Valor original del píxel					
	Escala Original		Escala x2		Escala x4	
	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)
3 metros	22.06 ± 12.22	14.45 ± 0.99	23.88 ± 13.2	15.83 ± 1.04	23.88 ± 13.2	15.83 ± 1.04
5 metros	27.49 ± 15.15	18.11 ± 1.38	29.29 ± 16.05	19.55 ± 1.42	29.29 ± 16.05	19.55 ± 1.42
7 metros	40.74 ± 22.53	22.76 ± 2.29	42.88 ± 23.62	24.26 ± 2.34	42.88 ± 23.62	24.26 ± 2.34
9 metros	40.06 ± 22.24	28.63 ± 2.68	41.69 ± 23.08	30.18 ± 2.76	41.69 ± 23.08	30.18 ± 2.76
11 metros	46.11 ± 27.53	33.54 ± 4.88	47.67 ± 28.29	35.14 ± 4.99	47.67 ± 28.29	35.14 ± 4.99
13 metros	53.49 ± 33.44	40.38 ± 7.16	54.97 ± 34.13	42.07 ± 7.29	54.97 ± 34.13	42.07 ± 7.29
15 metros	55.71 ± 31.66	43.38 ± 4.61	57.18 ± 32.39	45.09 ± 4.72	57.18 ± 32.39	45.09 ± 4.72
17 metros	64.3 ± 35.28	53.05 ± 4.58	65.7 ± 35.99	54.86 ± 4.71	65.7 ± 35.99	54.86 ± 4.71
19 metros	75.49 ± 41.23	72.14 ± 6.86	76.67 ± 41.83	74.13 ± 7.05	76.67 ± 41.83	74.13 ± 7.05

Tabla 6.18: Resultados del experimento 5 - Distancias: 3 a 19 metros (Valor original)

Distancia	Valor predicho del píxel					
	Escala Original		Escala x2		Escala x4	
	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)
3 metros	1.78 ± 2.36	0.91 ± 0.89	3.66 ± 3.37	0.91 ± 0.89	3.66 ± 3.37	1.94 ± 1.05
5 metros	2.48 ± 2.53	1.46 ± 1.04	2.92 ± 3.78	1.46 ± 1.04	2.92 ± 3.78	1.44 ± 1.19
7 metros	4.61 ± 3.21	2.39 ± 1.38	5.0 ± 4.8	2.39 ± 1.38	5.0 ± 4.8	2.16 ± 1.48
9 metros	3.07 ± 3.03	1.91 ± 1.33	3.91 ± 4.3	1.91 ± 1.33	3.91 ± 4.3	2.03 ± 1.37
11 metros	5.18 ± 4.47	3.11 ± 2.02	5.3 ± 5.64	3.11 ± 2.02	5.3 ± 5.64	2.88 ± 2.04
13 metros	9.37 ± 6.67	5.46 ± 2.57	8.89 ± 8.22	5.46 ± 2.57	8.89 ± 8.22	4.9 ± 2.75
15 metros	12.78 ± 7.08	7.85 ± 4.64	11.54 ± 6.43	7.85 ± 4.64	11.54 ± 6.43	7.0 ± 4.27
17 metros	12.08 ± 9.84	6.78 ± 5.18	11.03 ± 9.37	6.78 ± 5.18	11.03 ± 9.37	6.06 ± 4.97
19 metros	11.37 ± 8.11	6.75 ± 3.94	11.32 ± 9.15	6.75 ± 3.94	11.32 ± 9.15	6.67 ± 4.36

Tabla 6.19: Resultados del experimento 5 - Distancias: 3 a 19 metros (Valor predicho)

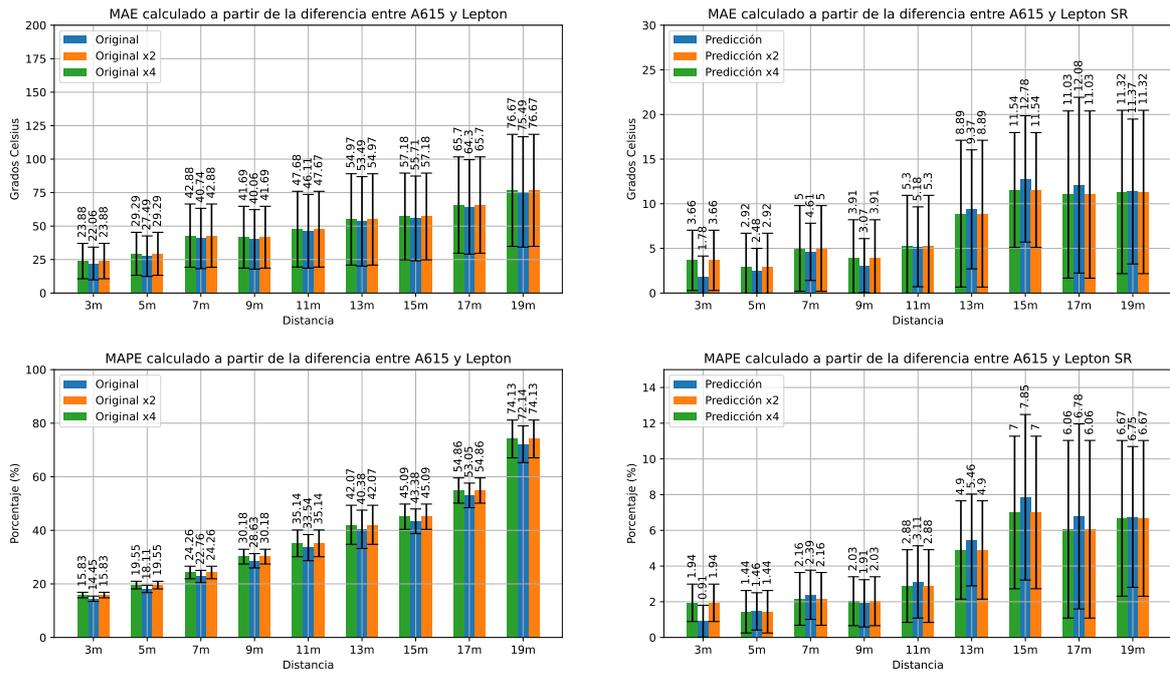


Figura 6.25: Representación gráfica de las Tablas 6.18 y 6.19

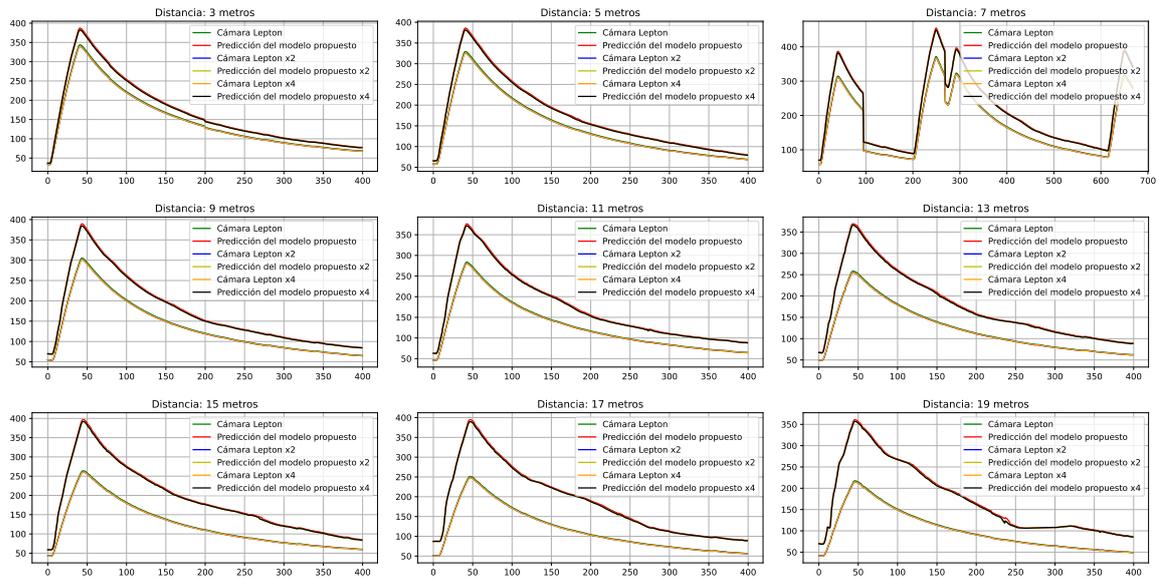


Figura 6.26: Muestras evaluadas del experimento 5 - Distancias: 3 a 19 metros

La Tabla 6.20 y la Tabla 6.21 muestran respectivamente que diferencia de grados existe entre los valores de la cámara Lepton y los valores de la cámara A615 para las distancias que van de 21 a 31 metros. En la Figura 6.27 se representan visualmente los valores de las tablas y en la Figura 6.28 se presentan los datos usados para la creación de las tablas y la gráfica de barras.

Distancia	Valor original del píxel					
	Escala Original		Escala x2		Escala x4	
	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)
21 metros	135.74 ± 81.45	93.61 ± 22.81	137.35 ± 82.15	95.96 ± 23.15	137.35 ± 82.15	95.96 ± 23.15
23 metros	134.47 ± 52.32	106.02 ± 12.59	136.0 ± 52.84	108.55 ± 12.77	136.0 ± 52.84	108.55 ± 12.77
25 metros	137.89 ± 27.45	124.25 ± 7.39	139.23 ± 27.69	126.99 ± 7.52	139.23 ± 27.69	126.99 ± 7.52
27 metros	168.27 ± 32.81	143.8 ± 15.46	169.71 ± 33.19	146.78 ± 15.65	169.71 ± 33.19	146.78 ± 15.65
29 metros	180.93 ± 56.44	205.44 ± 20.93	181.95 ± 56.73	209.06 ± 21.34	181.95 ± 56.73	209.06 ± 21.34
31 metros	179.63 ± 58.58	196.39 ± 23.94	180.69 ± 58.86	199.91 ± 24.37	180.69 ± 58.86	199.91 ± 24.37

Tabla 6.20: Resultados del experimento 5 - Distancias: 21 a 31 metros (Valor original)

Distancia	Valor predicho del píxel					
	Escala Original		Escala x2		Escala x4	
	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)	MAE (°C)	MAPE (%)
21 metros	41.02 ± 30.8	15.38 ± 6.94	41.77 ± 33.03	15.38 ± 6.94	41.77 ± 33.03	15.49 ± 7.7
23 metros	19.37 ± 22.83	6.94 ± 7.28	20.37 ± 23.17	6.94 ± 7.28	20.37 ± 23.17	7.29 ± 7.0
25 metros	14.56 ± 10.75	5.75 ± 4.37	13.77 ± 10.74	5.75 ± 4.37	13.77 ± 10.74	5.45 ± 4.19
27 metros	12.03 ± 8.25	4.2 ± 2.88	11.49 ± 8.83	4.2 ± 2.88	11.49 ± 8.83	3.99 ± 2.91
29 metros	34.87 ± 19.82	15.34 ± 7.97	37.57 ± 19.52	15.34 ± 7.97	37.57 ± 19.52	16.91 ± 8.16
31 metros	22.43 ± 19.56	8.59 ± 6.35	22.62 ± 20.49	8.59 ± 6.35	22.62 ± 20.49	8.64 ± 6.72

Tabla 6.21: Resultados del experimento 5 - Distancias: 21 a 31 metros (Valor predicho)

CAPÍTULO 6. RESULTADOS

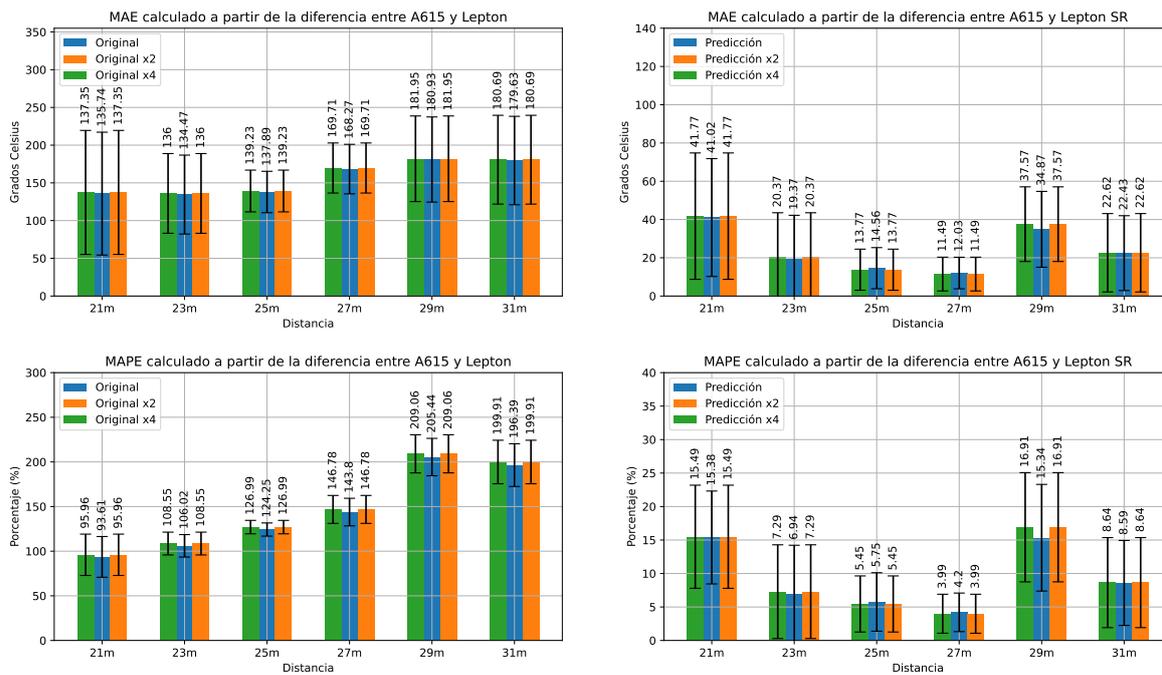


Figura 6.27: Representación gráfica de las Tablas 6.20 y 6.21

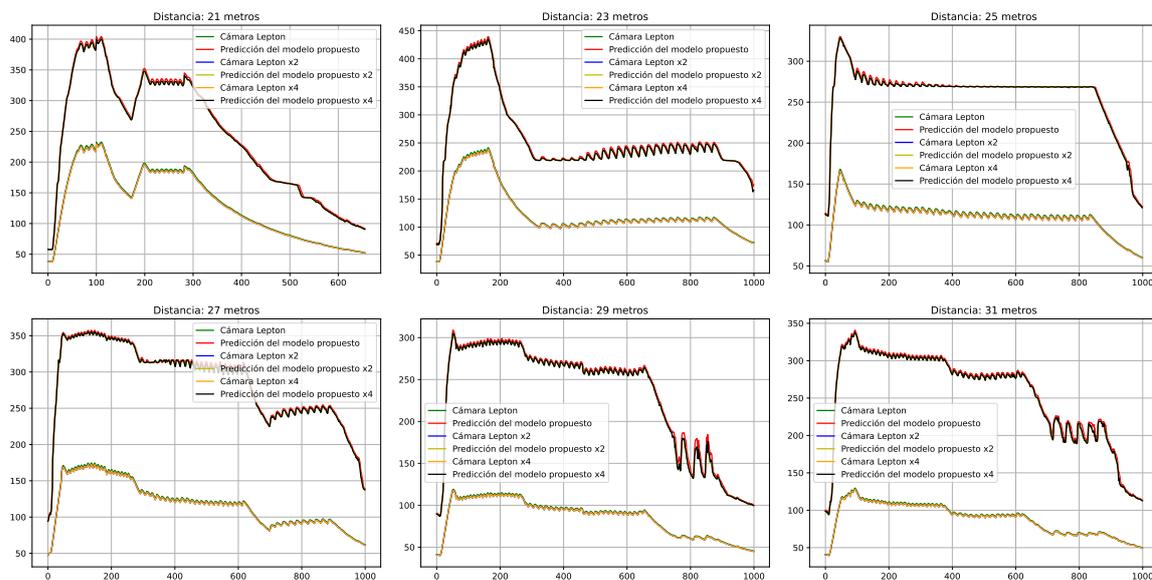


Figura 6.28: Muestras evaluadas del experimento 5 - Distancias: 21 a 31 metros

Capítulo 7

Conclusiones y líneas futuras

7.1. Conclusiones

Partiendo de los objetivos propuestos en este TFM, se han conseguido los siguientes hitos:

- **Estudio del estado de la técnica de los métodos de súper-resolución.**

Durante el trascurso de este TFM se ha realizado un estudio del marco teórico que abarca temas como la termografía, la súper-resolución y las redes neuronales. Por un lado, se ha investigado sobre las bases de la termografía, que factores intervienen en la medición de un cuerpo y que factores son fundamentales para entender cómo funcionan las cámaras termográficas. Por otro lado, a raíz de las limitaciones técnicas que presentan estos dispositivos en su resolución espacial, se han propuesto los métodos de súper-resolución como posible estrategia para mejorar sus prestaciones generales. Por este motivo, se ha estudiado el concepto de súper-resolución desde su concepción hasta la implementación moderna de soluciones basadas en aprendizaje profundo, más concretamente con redes neuronales convolucionales.

En este trabajo se ha realizado una distinción entre los métodos basados en redes neuronales y los métodos de reconstrucción clásicos, estableciendo ejemplos representativos de cada uno de estos y, además, explicando su funcionamiento de forma teórica.

– **Implementación completa de los métodos de súper-resolución.**

Una vez estudiado el estado de la técnica que compete a los métodos de súper-resolución, es necesario contar, por un lado, con unas implementaciones en algún lenguaje de programación y, por otro lado, con unas muestras suficientes para poder evaluar el desempeño general de estos métodos. Los métodos de súper-resolución basados en la reconstrucción de imágenes se han obtenido de una implementación accesible a través de GitHub, mientras que los métodos basados en redes neuronales han sido implementados y referenciados de distintos repositorios de la misma plataforma. Estos últimos cuentan con rutinas predefinidas para llevar a cabo los procesos de entrenamiento y evaluación con distintas bases de datos compuestas por imágenes.

Algunos de los métodos de súper-resolución basados en redes neuronales necesitaban de un entrenamiento por parte del usuario, mientras que otros contaban con modelos y pesos pre-entrenados. Los métodos SRGAN, EDSR y WDSR fueron entrenados en este trabajo con las bases de datos DIV2K y ADAS. La primera contaba con imágenes RGB, mientras que la segunda contaba con imágenes RGB y termográficas.

En este trabajo se ha propuesto un procesado específico para las imágenes termográficas, tanto las usadas en los métodos de reconstrucción como las usadas en los métodos basados en redes neuronales. Este procesado consiste en convertir el rango dinámico de la imagen a 8 bits, conservando referencias de los valores radiométricos que se tenía anteriormente. A partir de este punto, se aplica el método de súper-resolución a la imagen de 8 bits. Finalmente, a la imagen SR resultante, se le realiza un proceso inverso para recuperar los valores radiométricos. Este procesado permite que métodos de súper-resolución concebidos para imágenes RGB puedan ser utilizados en imágenes termográficas, lo que facilita la realización de una futura comparación entre estos.

– **Concepción de la súper-resolución radiométrica y creación de distintas bases de datos propias.**

En este TFM se han estudiado distintos métodos de súper-resolución que mejoran la calidad visual de las imágenes, a estos métodos se les denominan métodos de súper-resolución espacial. Este campo ha sido estudiado durante mucho tiempo y comprende

una gran cantidad de implementaciones de libre acceso que cuentan con un gran desempeño. No obstante, en el estado de la técnica apenas se profundiza en el valor de estos píxeles, ya que se suelen tratar de imágenes RGB que referencian valores numéricos comprendidos entre 0 y 255. En este aspecto, no se suele trabajar tanto con el concepto de imágenes termográficas, que ofrecen datos estimados de la temperatura de un objeto y se codifican en valores numéricos de 16 bits. Por este motivo, es muy importante observar cómo afectan estos métodos de súper-resolución espacial a las imágenes termográficas, observando si existe alguna pérdida de temperatura durante este proceso. Para alcanzar el objetivo propuesto, una base de datos de 160 imágenes termográficas ha sido generada a partir las cámaras A615, A35 y Lepton. Esta base de datos se ha denominado DASSR y ha sido segmentada según el escenario de las medidas para llevar a cabo una evaluación de los métodos de súper-resolución espacial.

Por otro lado, para complementar el desempeño de los métodos de súper-resolución espacial, se propone lo que en este trabajo se ha denominado súper-resolución radiométrica. Este proceso implica que además de mejorar la calidad visual de las imágenes con los métodos anteriores, se intente mejorar el propio valor de los píxeles que ofrece una cámara termográfica de baja resolución. Durante esta primera aproximación se ha propuesto que la corrección de los píxeles se realice introduciendo el parámetro de distancia entre la cámara y el objeto. Para poder entrenar y evaluar el modelo propuesto se ha generado una base de datos llamada TID, contando con imágenes de las cámaras A615, A35, Tau 2 y Lepton.

Para el entrenamiento de la red neuronal propuesta se plantea un aprendizaje supervisado, donde las entradas corresponden a los valores de la cámara Lepton y las etiquetas corresponden a los valores de la cámara A615. Por el momento, las imágenes de la A35 y Tau 2 no se usan en esta primera aproximación a la súper-resolución radiométrica. Una parte de la base de datos TID es destinada a la evaluación de la red neuronal, consiguiendo los resultados comentados a continuación.

– Evaluación de los resultados.

Los resultados de este trabajo han sido presentados en el capítulo 6. En este capítulo se puede observar, durante las discusiones llevadas a cabo en los experimentos 1, 2 y 3, que métodos de súper-resolución espacial tienen menos pérdidas a la hora de procesar una imagen termográfica, además de los métodos que consiguen una mejor calidad visual final. Durante el desarrollo de estos experimentos se ha observado que variantes de los métodos de súper-resolución espacial como el SRGAN, EDSR y el WDSR, entrenadas con imágenes RGB de la base de datos DIV2K, tienen un desempeño similar que sus variantes entrenadas con imágenes termográficas de 8 bits a partir de la base de datos ADAS.

La red neuronal propuesta para la súper-resolución radiométrica ha sido probada durante los experimentos 4 y 5. Esta red neuronal se ha evaluado a través de una gran cantidad de muestras de la base de datos TID, observando que existe una mejora notable en los valores de la cámara Lepton una vez se aplica el flujo de trabajo planteado. El uso de esta red neuronal mejora el promedio general de la cámara para distancias acotadas de 1 a 32 metros, siendo sus resultados comparables a las muestras obtenidas de la cámara A35, una cámara de mejores prestaciones. Por último, se ha unido el desempeño de un método de súper-resolución espacial, el EDSR, con el modelo de súper-resolución radiométrica, observando que se mejora tanto la calidad visual de la imagen como los valores de temperatura que esta ofrece.

Se puede concluir que al unir estos dos tipos de súper-resolución se ha mejorado, al menos para distancias cercanas, el desempeño general de una cámara termográfica de bajo coste como es la Lepton, superando así algunas limitaciones inherentes que existen en esta por el conjunto óptica-sensor integrado. Este factor podría abrir una nueva línea de investigación que estableciera una comparativa mucho más rigurosa entre las 4 cámaras usadas durante el desarrollo de este trabajo, con una posible conclusión que contemplara el uso de la cámara Lepton para cierto tipo de aplicaciones en lugar de alternativas más caras del mercado.

De hecho, sólo para tener una referencia de precios, el coste en el mercado de las cámaras termográficas usadas en este TFM se muestra en la Tabla 7.1.

Cámaras	Precio (€)
A615	16698.0
Tau 2	7619.0
A35	4999.0
Lepton 3.5	175.84

Tabla 7.1: Precio de mercado de las cámaras de FLIR utilizadas

Como se observa en la tabla anterior, debido a la gran diferencia de precios en el mercado, podría resultar de interés la implementación de una cámara de bajo coste con los métodos anteriormente nombrados para así disminuir las inversiones en recursos hardware de un proyecto.

– Métricas usadas para la evaluación.

Las métricas usadas en este TFM han seguido la línea de las propuestas por otros autores en el estado de la técnica. Sin embargo, se han usado otras métricas, no tan presentes en el estado de la técnica, para poder parametrizar la pérdida que existe al aplicar métodos de súper-resolución espacial en imágenes termográficas de 16 bits.

Por un lado, se han usado métricas objetivas como la PSNR y el SSIM. La primera muestra un desempeño correcto al evaluar cómo se ha degradado la imagen LR procesada por un método de súper-resolución espacial respecto a su equivalente HR. No obstante, la segunda no muestra una información tan clara para imágenes de 16 bits. Para las imágenes SR termográficas planteadas en la base de datos DASSR se ha llegado a conseguir un SSIM de 1 con una desviación típica nula. Esto sólo debería ser posible en la teoría, ya que se sabe que la imagen LR ha sido degradada previamente y no podrá asemejarse a la imagen original. Ante estos resultados se plantea que el SSIM no es una métrica que ofrezca valores realistas sobre imágenes de 16 bits, algo que podría deberse al amplio rango dinámico de estas imágenes (valores numéricos que oscilan entre 0 y 65535). Por otro lado, para evaluar las pérdidas en grados centígrados de una imagen SR respecto a su imagen HR original, se han usado diferencias absolutas entre los máximos de cada imagen y diferencias promediadas de sus matrices de datos. Estas dos métricas propuestas ayudan a entender de forma intuitiva cuál ha sido la pérdida inherente de aplicar distintos métodos de súper-resolución espacial.

Finalmente, para evaluar el desempeño de la red neuronal planteada para la súper-resolución radiométrica, se han propuesto métricas como el MAE y el MAPE. Estas métricas evalúan el desempeño promedio de la red neuronal a través de la diferencia entre el valor predicho por esta y el valor de la etiqueta insertado. La primera métrica ofrece una diferencia absoluta media, mientras que la segunda ofrece información de cuanto se parece un valor a otro en términos de %.

7.2. Líneas futuras

De las conclusiones expuestas en el apartado anterior se especifican las siguientes líneas futuras:

– **Integración en un sistema embebido.**

Durante el transcurso de este TFM se ha realizado el procesado y las operaciones de súper-resolución en un portátil Windows. No obstante, es posible plantear una implementación completa de los métodos anteriormente nombrados en un sistema embebido, sobre todo para tareas donde plataformas de poco consumo y pequeñas dimensiones sean necesarias. En la actualidad, la tecnología de microcontroladores como la Raspberry Pi permite la implementación de la cámara Lepton en un sistema de muy pequeñas dimensiones. Por otro lado, como los métodos de súper-resolución anteriores son alojados en entornos virtuales separados, no existe a priori ningún problema con su implementación en un sistema derivado de Debian GNU/Linux.

– **Convergencia de los métodos de súper-resolución espacial.**

Uno de los mayores problemas a la hora de implementar los métodos de súper-resolución basados en redes neuronales es su compatibilidad con las versiones de distintas bibliotecas software. Es decir, cada uno de los métodos nombrados en este trabajo ha sido implementado por distintos usuarios a través de distintas versiones de Python, numpy y OpenCV. De la misma manera, muchos de ellos se han implementado en bibliotecas como TensorFlow, Keras y Pytorch, también de distintas versiones. Por este motivo ha sido necesario separar su implementación local en distintos entornos virtuales, activando y desactivando estos según la necesidad. Una posible línea futura de este

trabajo es la implementación completa de los métodos de súper-resolución espacial que se consideren de mayor interés, como los nombrados en el capítulo 6, bajo una misma versión de Python y una misma biblioteca de aprendizaje automático, centralizando así el uso de estos métodos.

– **Otras campañas de medida.**

En este TFM se ha desarrollado una campaña de medidas en un entorno de interior controlado. Es decir, las muestras destinadas al entrenamiento y la evaluación de la red neuronal para la súper-resolución radiométrica estaban adecuadas a este entorno. Por esta razón, una forma de validar el desempeño de la red neuronal de forma más precisa es llevar a cabo una campaña de medidas en exteriores, donde condiciones climatológicas adversas podrían intervenir en la predicción de la red neuronal propuesta. De la misma forma, esta campaña de medidas podría ser utilizada para seguir alimentando la red neuronal con nuevas muestras durante futuros procesos de entrenamiento.

– **Optimización de la red neuronal propuesta para la súper-resolución radiométrica.**

La red neuronal propuesta en este TFM es una primera aproximación o prototipo de súper-resolución radiométrica aplicada a imágenes termográficas de baja resolución. Si bien es cierto que ofrece un buen desempeño sobre las muestras de evaluación presentadas en el capítulo 6, es necesario seguir trabajando en su optimización. Por un lado, se podrían probar distintas configuraciones de capas y neuronas, aumentando o disminuyendo estas para así observar si existe una mejora perceptible en las prestaciones generales de la red neuronal. Por otro lado y con el mismo propósito, podría ser de interés modificar algún hiperparámetro como el tamaño del lote utilizado, la tasa de aprendizaje del optimizador o el número de épocas. De la misma manera, comparar la red neuronal presentada en este trabajo con otra red neuronal compilada a partir de una función de pérdida distinta a la función MAE, como puede ser la función MSE o RMSE, también podría ofrecer otra perspectiva de cara a una futura optimización.

Parte II

Bibliografía

Bibliografía

- [1] I. Newton, *Opticks: or, A treatise of the reflections, refractions, inflections, and colours of light*. Good Press, 2019.
- [2] W. Kemp, «Energy and the Electromagnetic Spectrum,» en *Organic Spectroscopy*. London: Macmillan Education UK, 1991, págs. 1-17, ISBN: 978-1-349-15203-2. DOI: 10.1007/978-1-349-15203-2_1. dirección: https://doi.org/10.1007/978-1-349-15203-2_1.
- [3] J. Luque Ordóñez, «Espectro electromagnético y espectro radioeléctrico,» dirección: https://www.acta.es/medios/articulos/ciencias_y_tecnologia/062017.pdf.
- [4] C. Meola y G. M. Carlomagno, «Recent advances in the use of infrared thermography,» vol. 15, n.º 9, R27-R58, jul. de 2004. DOI: 10.1088/0957-0233/15/9/r01. dirección: <https://doi.org/10.1088/0957-0233/15/9/r01>.
- [5] G. Gaussorgues y S. Chomet, *Infrared thermography*. Springer Science & Business Media, 1993, vol. 5.
- [6] R. Gade y T. B. Moeslund, «Thermal cameras and applications: a survey,» *Machine Vision and Applications 2013 25:1*, vol. 25, n.º 1, págs. 245-262, 2013, ISSN: 1432-1769. DOI: 10.1007/S00138-013-0570-5. dirección: <https://link.springer.com/article/10.1007/s00138-013-0570-5>.
- [7] *FLIR Commercial Systems - PDF Free Download*. dirección: <https://docplayer.net/24170467-Flir-commercial-systems.html> (visitado 07-09-2021).
- [8] F. Systems, «5 Factors influencing radiometric temperature measurements,» FLIR Systems Inc., inf. téc., 2016.
- [9] W. Minkina y D. Klecha, «Atmospheric transmission coefficient modelling in the infrared for thermovision measurements,» en *JSSS*, 2016, vol. 5, págs. 17-2.

- [10] F. Amon y C. Pearson, «Chapter 5 Thermal Imaging in Firefighting and Thermography Applications,» en dic. de 2010, vol. 43, págs. 279-331, ISBN: 9780123750914. DOI: 10.1016/S1079-4042(09)04305-7.
- [11] T. Luhmann, J. Ohm, J. Piechel y T. Roelfs, «International Archives of Photogrammetry,» *Remote Sensing and Spatial Information Sciences*, vol. XXXVIII, 2010.
- [12] T. Luhmann, J. Piechel y T. Roelfs, «Geometric Calibration of Thermographic Cameras,» *Remote Sensing and Digital Image Processing*, vol. 17, págs. 27-42, 2013. DOI: 10.1007/978-94-007-6639-6_2. dirección: https://link.springer.com/chapter/10.1007/978-94-007-6639-6_2.
- [13] F. Liu, P. Han, Y. Wang, X. Li, L. Bai y X. Shao, «Super resolution reconstruction of infrared images based on classified dictionary learning,» *Infrared Physics & Technology*, vol. 90, págs. 146-155, mayo de 2018, ISSN: 1350-4495. DOI: 10.1016/J.INFRARED.2018.03.008.
- [14] A. Rogalski, P. Martyniuk y M. Kopytko, «Challenges of small-pixel infrared detectors: a review,» *Reports on Progress in Physics*, vol. 79, n.º 4, pág. 046 501, mar. de 2016, ISSN: 0034-4885. DOI: 10.1088/0034-4885/79/4/046501. dirección: <https://iopscience.iop.org/article/10.1088/0034-4885/79/4/046501>.
- [15] Y. Zhao, Q. Chen, X. Sui y G. Gu, «A Novel Infrared Image Super-Resolution Method Based on Sparse Representation,» *Infrared Physics & Technology*, vol. 71, jul. de 2015. DOI: 10.1016/j.infrared.2015.06.017.
- [16] J. Wang e Y. Liu, «Infrared image super-resolution enhancement based on convolutional neural network,» *Proceedings - International Joint Conference on Information, Media, and Engineering, IJCIME 2019*, págs. 272-275, dic. de 2019. DOI: 10.1109/IJCIME49369.2019.00061.
- [17] K. Nasrollahi y T. B. Moeslund, «Super-resolution: a comprehensive survey,» *Machine Vision and Applications 2014 25:6*, vol. 25, n.º 6, págs. 1423-1468, jun. de 2014, ISSN: 1432-1769. DOI: 10.1007/S00138-014-0623-4. dirección: <https://link.springer.com/article/10.1007/s00138-014-0623-4>.

- [18] A. Lukin, A. Krylov y A. Nasonov, «Image Interpolation by Super-Resolution,» dic. de 2010.
- [19] T. Gotoh y M. Okutomi, «Direct super-resolution and registration using raw CFA images,» *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2004. DOI: 10.1109/CVPR.2004.1315219.
- [20] P. Gaidhani, «SUPER-RESOLUTION,»
- [21] M. Cristani, D. S. Chengy, V. Murino y D. Pannullo, «Distilling information with super-resolution for video surveillance,» *VSSN'04 - Proceedings of the ACM Second International Workshop on Video Surveillance and Sensor Networks*, págs. 2-11, 2004. DOI: 10.1145/1026799.1026803.
- [22] M. U. Müller, N. Ekhtiari, R. M. Almeida y C. Rieke, «Super-Resolution of Multispectral Satellite Images Using Convolutional Neural Networks,» *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 5, n.º 1, págs. 33-40, ago. de 2020. DOI: 10.5194/ISPRS-ANNALS-V-1-2020-33-2020.
- [23] L. Yue, H. Shen, J. Li, Q. Yuan, H. Zhang y L. Zhang, «Image super-resolution: The techniques, applications, and future,» *Signal Processing*, vol. 128, págs. 389-408, nov. de 2016, ISSN: 0165-1684. DOI: 10.1016/J.SIGPRO.2016.05.002.
- [24] Y. H. Wang, J. Qiao, J. B. Li, P. Fu, S. C. Chu y J. F. Roddick, «Sparse representation-based MRI super-resolution reconstruction,» *Measurement*, vol. 47, n.º 1, págs. 946-953, ene. de 2014, ISSN: 0263-2241. DOI: 10.1016/J.MEASUREMENT.2013.10.026.
- [25] H. Greenspan, «Super-resolution in medical imaging,» *Computer Journal*, vol. 52, n.º 1, págs. 43-63, ene. de 2009. DOI: 10.1093/COMJNL/BXM075.
- [26] R. Willett, I. Jermyn, R. Nowak y J. Zerubia, «Wavelet-based superresolution in astronomy,» *undefined*, 2003.
- [27] L. Li, Q. Yu, Y. Yuan, Y. Shang, H. Lu y X. Sun, «Super-resolution reconstruction and higher-degree function deformation model based matching for Chang'E-1 lunar images,» *Science in China Series E: Technological Sciences* 2009 52:12,

- vol. 52, n.º 12, págs. 3468-3476, dic. de 2009, ISSN: 1862-281X. DOI: 10.1007/S11431-009-0334-7. dirección: <https://link.springer.com/article/10.1007/s11431-009-0334-7>.
- [28] *Low-cost, low-resolution IR system with super-resolution interpolation of thermal images for industrial applications - Measurement Automation Monitoring - Tom Vol. 64, No. 4 (2018) - BazTech - Yadda*. dirección: <http://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-88f954f1-3719-43c7-ae76-87675cb14e8d> (visitado 08-09-2021).
- [29] P. Kansal y S. Nathan, «A multi-level supervision model: A novel approach for thermal image super resolution,» *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2020-June, págs. 426-431, jun. de 2020. DOI: 10.1109/CVPRW50498.2020.00055.
- [30] V. Chudasama, H. Patel, K. Prajapati y col., «TherISuRNet - A Computationally Efficient Thermal Image Super-Resolution Network,» en *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, págs. 388-397. DOI: 10.1109/CVPRW50498.2020.00051.
- [31] P. Cascarano, F. Corsini, S. Gandolfi y col., «Super-Resolution of Thermal Images Using an Automatic Total Variation Based Method,» *Remote Sensing*, vol. 12, n.º 10, 2020, ISSN: 2072-4292. DOI: 10.3390/rs12101642. dirección: <https://www.mdpi.com/2072-4292/12/10/1642>.
- [32] W. Minkina, «Theoretical basics of radiant heat transfer – practical examples of calculation for the infrared (IR) used in infrared thermography measurements,» *Quantitative InfraRed Thermography Journal*, vol. 18, n.º 4, págs. 269-282, 2021. DOI: 10.1080/17686733.2020.1738164. dirección: <https://doi.org/10.1080/17686733.2020.1738164>.
- [33] F. Systems, «User’s manual FLIR Ax35 series.,» FLIR Commercial Systems, inf. téc., 2015.
- [34] D. S. Tezcan, S. Eminoglu, O. S. Akar y T. Akin, «A low cost uncooled infrared microbolometer focal plane array using the CMOS n-well layer,» *Proceedings of the IEEE Micro Electro Mechanical Systems (MEMS)*, págs. 566-569, 2001. DOI: 10.1109/MEMSYS.2001.906604.

- [35] R. A. Wood, «Uncooled thermal imaging with monolithic silicon focal planes,» en *Infrared Technology XIX*, B. F. Andresen y F. D. Shepherd, eds., International Society for Optics y Photonics, vol. 2020, SPIE, 1993, págs. 322-329. DOI: 10.1117/12.160553. dirección: <https://doi.org/10.1117/12.160553>.
- [36] X. Chen y J. Li, «A novel VOx thermosensitive thin film for microbolometer arrays,» *2011 Symposium on Photonics and Optoelectronics, SOPO 2011*, 2011. DOI: 10.1109/SOPO.2011.5780577.
- [37] H. R. Beratan, C. M. Hanson y E. G. Meissner, «Low-cost uncooled ferroelectric detector,» en *Infrared Detectors: State of the Art II*, R. E. Longshore, ed., International Society for Optics y Photonics, vol. 2274, SPIE, 1994, págs. 147-156. dirección: <https://doi.org/10.1117/12.189240>.
- [38] A. Tanaka, S. Matsumoto, N. Tsukamoto y col., «Infrared focal plane array incorporating silicon IC process compatible bolometer,» *IEEE Transactions on Electron Devices*, vol. 43, n.º 11, págs. 1844-1850, nov. de 1996. DOI: 10.1109/16.543017.
- [39] P. Chavez-Burbano, V. Guerra, J. Rabadan, D. Rodríguez-Esparragón y R. Perez-Jimenez, «Experimental Characterization of Close-Emitter Interference in an Optical Camera Communication System,» *Sensors*, vol. 17, n.º 7, 2017, ISSN: 1424-8220. DOI: 10.3390/s17071561. dirección: <https://www.mdpi.com/1424-8220/17/7/1561>.
- [40] R. Bhan, R. Saxena, C. Jalwania y S. Lomash, «Uncooled Infrared Microbolometer Arrays and their Characterisation Techniques,» *Defence Science Journal*, vol. 59, pág. 580, nov. de 2009. DOI: 10.14429/dsj.59.1562.
- [41] E. Serrano Cañizares, «Súper-resolución de imágenes en PFGAs para procesamiento a bordo en satélites.» 2017. dirección: <https://ruidera.uclm.es/xmlui/handle/10578/15428>.
- [42] J. Tian y K.-K. Ma, «A survey on super-resolution imaging,» *Signal, Image and Video Processing 2011 5:3*, vol. 5, n.º 3, págs. 329-342, feb. de 2011, ISSN: 1863-1711. DOI: 10.1007/S11760-010-0204-6. dirección: <https://link.springer.com/article/10.1007/s11760-010-0204-6>.

- [43] T. S. H. y R. Y. Tsay, «Multiple frame image restoration and registration,» en *Advances in Computer Vision and Image Processing*, 1.^a ed., Jai Press, 1984, págs. 317-339.
- [44] S. C. Park, M. K. Park y M. G. Kang, «Super-resolution image reconstruction: a technical overview,» *IEEE Signal Processing Magazine*, vol. 20, n.º 3, págs. 21-36, 2003. DOI: 10.1109/MSP.2003.1203207.
- [45] A. Gilman, D. G. Bailey y S. R. Marsland, «Interpolation models for image super-resolution,» *Proceedings - 4th IEEE International Symposium on Electronic Design, Test and Applications, DELTA 2008*, págs. 55-60, 2008. DOI: 10.1109/DELTA.2008.104.
- [46] Y. Romano, J. Isidoro y P. Milanfar, «RAISR: Rapid and Accurate Image Super Resolution,» *IEEE Transactions on Computational Imaging*, vol. 3, n.º 1, págs. 110-125, jun. de 2016. arXiv: 1606.01299. dirección: <https://arxiv.org/abs/1606.01299v3>.
- [47] K. Srinivasan y J. Kanakaraj, «An Overview of SR Techniques Applied to Images, Videos and Magnetic Resonance Images,» *Smart Computing Review*, vol. 4, págs. 181-201, jun. de 2014. DOI: 10.6029/smartcr.2014.03.005.
- [48] V. Bannore, «Regularization for Super-Resolution Image Reconstruction,» en *Knowledge-Based Intelligent Information and Engineering Systems*, B. Gabrys, R. J. Howlett y L. C. Jain, eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, págs. 36-46, ISBN: 978-3-540-46539-3.
- [49] HADAMARD y J., «Sur les problemes aux derivees partielles et leur signification physique,» *Princeton University Bulletin*, págs. 49-52, 1902. dirección: <http://ci.nii.ac.jp/naid/10030001312/en/>.
- [50] M. Elad y A. Feuer, «Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images,» *IEEE Transactions on Image Processing*, vol. 6, n.º 12, págs. 1646-1658, 1997. DOI: 10.1109/83.650118.
- [51] Y. Shashi Rathore y /. Harshalatha, «Iterative Back-Projection Algorithm Based Signal Processing Approach To Enhance An Image Resolution,» *International Journal of Engineering Research and Applications (IJERA)*, vol. 2,

- págs. 1449-1454, dirección: https://www.ijera.com/papers/Vol2_issue6/HI2614491454.pdf.
- [52] H. Song, X. He, W. Chen e Y. Sun, «An Improved Iterative Back-Projection Algorithm for Video Super-Resolution Reconstruction,» en *2010 Symposium on Photonics and Optoelectronics*, 2010, págs. 1-4. DOI: 10.1109/SOP0.2010.5504209.
- [53] M. Toghi, K. Kose, A. E. Cetin y M. Sloan, «Projection Onto Convex Sets (POCS) Based Signal Reconstruction Framework with an associated cost function,» 2015.
- [54] E. Quevedo, *Contribuciones al proceso de súper-resolución mediante técnicas de filtros selectivos, topología de macro-bloques adaptable y sistemas multi-cámara*, 2015. dirección: <https://accedacris.ulpgc.es/handle/10553/17288>.
- [55] A. J. Patti, M. I. Sezan y A. Murat Tekalp, «Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time,» *IEEE Transactions on Image Processing*, vol. 6, n.º 8, págs. 1064-1076, 1997. DOI: 10.1109/83.605404.
- [56] R. Escalante y R. Porras, «El método de las proyecciones alternas aplicado al problema de super-resolución,» mayo de 2011.
- [57] A. A. Samsonov, E. G. Kholmovski, D. L. Parker y C. R. Johnson, «POCSENSE: POCS-based reconstruction for sensitivity encoded magnetic resonance imaging,» *Magnetic Resonance in Medicine*, vol. 52, n.º 6, págs. 1397-1406, 2004. DOI: <https://doi.org/10.1002/mrm.20285>. dirección: <https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.20285>.
- [58] A. Papoulis, «A new algorithm in spectral analysis and band-limited extrapolation,» *IEEE Transactions on Circuits and Systems*, vol. 22, n.º 9, págs. 735-742, 1975. DOI: 10.1109/TCS.1975.1084118.
- [59] R. W. Gerchberg, «Super-resolution through Error Energy Reduction,» *Journal of Modern Optics*, vol. 21, págs. 709-720, 1974.
- [60] P. Chatterjee, S. Mukherjee, S. Chaudhuri y G. Seetharaman, «Application Of Papoulis-Gerschberg Method In Image Super-Resolution and Inpainting,» *Comput. J.*, vol. 52, págs. 80-89, 2009. DOI: 10.1093/comjnl/bxm050.

- [61] P. Vandewalle, S. Süsstrunk y M. Vetterli, «Superresolution Images Reconstructed from Aliased Images,» *Proceedings of SPIE - The International Society for Optical Engineering*, 2003. DOI: 10.1117/12.506874.
- [62] A. Zomet, A. Rav-Acha y S. Peleg, «Robust super-resolution,» en *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2001, págs. I-645, ISBN: 0-7695-1272-0. DOI: 10.1109/CVPR.2001.990535.
- [63] S. Farsiu, M. D. Robinson, M. Elad y P. Milanfar, «Fast and robust multiframe super resolution,» *IEEE Transactions on Image Processing*, vol. 13, n.º 10, págs. 1327-1344, 2004. DOI: 10.1109/TIP.2004.834669.
- [64] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*. MIT Press, 2016.
- [65] O. Aleksander, *Understanding neural networks 1: The concept of neurons / by Aleksander Obuchowski / Becoming Human: Artificial Intelligence Magazine*, 2019. dirección: <https://becominghuman.ai/understanding-neural-networks-1-the-concept-of-neurons-287be36d40f> (visitado 29-10-2021).
- [66] Agrawal Rochak, *Shallow Neural Networks*. dirección: <https://towardsdatascience.com/shallow-neural-networks-23594aa97a5> (visitado 15-12-2021).
- [67] IBM Cloud Education, *What are Neural Networks? | IBM*, 2020. dirección: <https://www.ibm.com/cloud/learn/neural-networks> (visitado 29-10-2021).
- [68] D. P. Kingma y J. Ba, *Adam: A Method for Stochastic Optimization*, 2017. arXiv: 1412.6980 [cs.LG].
- [69] D. E. Rumelhart, G. E. Hinton y R. J. Williams, «Learning representations by back-propagating errors,» *Nature*, vol. 323, n.º 6088, págs. 533-536, 1986, ISSN: 1476-4687. DOI: 10.1038/323533a0. dirección: <https://doi.org/10.1038/323533a0>.
- [70] O. Aleksander, *Understanding neural networks 2: The math of neural networks in 3 equations / by Aleksander Obuchowski / Becoming Human: Artificial Intelligence Magazine*, 2020. dirección: <https://becominghuman.ai/understanding-neural-networks-2-the-math-of-neural-networks-in-3-equations-6085fd3f09df> (visitado 29-10-2021).

- [71] M. Marzoa Tanco, «Super-resolución en imágenes,» 2019. dirección: <https://hdl.handle.net/20.500.12008/26226>.
- [72] A. Puertas Cabedo, «Super-Resolución inter-sensor para el realce espacial de imágenes Sentinel,» 2018. dirección: <http://repositori.uji.es/xmlui/handle/10234/180320>.
- [73] C. Dong, C. C. Loy, K. He y X. Tang, *Image Super-Resolution Using Deep Convolutional Networks*, 2015. arXiv: 1501.00092 [cs.CV].
- [74] e. a. Dong Chao, *Learning a Deep Convolutional Network for Image Super-Resolution*, 2015. dirección: <http://mmlab.ie.cuhk.edu.hk/projects/SRCNN.html>.
- [75] W. Yang, X. Zhang, Y. Tian, W. Wang y J.-H. Xue, «Deep Learning for Single Image Super-Resolution: A Brief Review,» *IEEE Transactions on Multimedia*, vol. 21, n.º 12, págs. 3106-3121, 2018. DOI: 10.1109/tmm.2019.2919431. arXiv: 1808.03344. dirección: <https://arxiv.org/abs/1808.03344v3>.
- [76] S. Anwar, S. H. Khan y N. Barnes, «A Deep Journey into Super-resolution: A survey,» *CoRR*, vol. abs/1904.0, 2019. dirección: <http://arxiv.org/abs/1904.07523>.
- [77] K. He, X. Zhang, S. Ren y J. Sun, «Deep Residual Learning for Image Recognition,» *CoRR*, vol. abs/1512.0, 2015. dirección: <http://arxiv.org/abs/1512.03385>.
- [78] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza y col., *Generative Adversarial Networks*, 2014. arXiv: 1406.2661 [stat.ML].
- [79] Orhan G. Yalçın, *Top 6 Deep Learning Models You Should Master for Killer AI Applications*. dirección: <https://towardsdatascience.com/top-6-deep-learning-models-you-should-master-for-killer-ai-applications-13c7dfa68a3> (visitado 16-12-2021).
- [80] Z. Wang, J. Chen y S. C. H. Hoi, «Deep Learning for Image Super-resolution: A Survey,» *CoRR*, vol. abs/1902.0, 2019. dirección: <http://arxiv.org/abs/1902.06068>.

- [81] W. Shi, J. Caballero, F. Huszár y col., *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network*, 2016. arXiv: 1609.05158 [cs.CV].
- [82] J. Kim, J. K. Lee y K. M. Lee, «Accurate Image Super-Resolution Using Very Deep Convolutional Networks,» *CoRR*, vol. abs/1511.0, 2015. dirección: <http://arxiv.org/abs/1511.04587>.
- [83] C. Ledig, L. Theis, F. Huszar y col., «Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,» *CoRR*, vol. abs/1609.0, 2016. dirección: <http://arxiv.org/abs/1609.04802>.
- [84] K. Simonyan y A. Zisserman, «Very Deep Convolutional Networks for Large-Scale Image Recognition,» *arXiv 1409.1556*, 2014.
- [85] B. Lim, S. Son, H. Kim, S. Nah y K. M. Lee, «Enhanced Deep Residual Networks for Single Image Super-Resolution,» *CoRR*, vol. abs/1707.0, 2017. dirección: <http://arxiv.org/abs/1707.02921>.
- [86] S. Ioffe y C. Szegedy, «Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,» *CoRR*, vol. abs/1502.03167, 2015. arXiv: 1502.03167. dirección: <http://arxiv.org/abs/1502.03167>.
- [87] J. Yu, Y. Fan, J. Yang y col., «Wide Activation for Efficient and Accurate Image Super-Resolution,» *CoRR*, vol. abs/1808.0, 2018. dirección: <http://arxiv.org/abs/1808.08718>.
- [88] X. Wang, K. Yu, S. Wu y col., «ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks,» *CoRR*, vol. abs/1809.0, 2018. dirección: <http://arxiv.org/abs/1809.00219>.
- [89] X. Wang, L. Xie, C. Dong e Y. Shan, *Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data*, 2021. arXiv: 2107.10833 [eess.IV].
- [90] *42: A blog on A.I.* dirección: https://nchlis.github.io/2019_10_30/page.html (visitado 16-12-2021).

- [91] E. Agustsson y R. Timofte, «NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study,» en *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.
- [92] D. newbie, *FLIR Thermal Images Dataset | Kaggle*. dirección: <https://www.kaggle.com/deepnewbie/flir-thermal-images-dataset> (visitado 29-11-2021).
- [93] AlgorithmHub, *exampleSR: Super Resolution example*. dirección: https://github.com/AlgorithmHub/example_sr (visitado 29-11-2021).
- [94] Lornatang, *VDSR-PyTorch: PyTorch implements “Accurate Image Super-Resolution Using Very Deep Convolutional Networks”*. dirección: <https://github.com/Lornatang/VDSR-PyTorch> (visitado 29-11-2021).
- [95] Leftthomas, *SRGAN: A PyTorch implementation of SRGAN based on CVPR 2017 paper “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network”*. dirección: <https://github.com/leftthomas/SRGAN> (visitado 29-11-2021).
- [96] Krasserm, *super-resolution: Tensorflow 2.x based implementation of EDSR, WDSR and SRGAN for single image super-resolution*. dirección: <https://github.com/krasserm/super-resolution> (visitado 29-11-2021).
- [97] Saafke, *EDSR Tensorflow at mdsr*. dirección: https://github.com/Saafke/EDSR_Tensorflow/tree/mdsr (visitado 29-11-2021).
- [98] Xinntao, *Real-ESRGAN: Real-ESRGAN aims at developing Practical Algorithms for General Image Restoration*. dirección: <https://github.com/xinntao/Real-ESRGAN> (visitado 29-11-2021).
- [99] ULPGC, *Boletín Oficial de la Universidad de Las Palmas de Gran Canaria*, 2021. dirección: https://www.ulpgc.es/sites/default/files/ArchivosULPGC/boulpgc/BOULPGC/boulpgc_1_julio_2021.pdf.
- [100] COIT, *Derechos de visado 2021*, 2021. dirección: https://www.coit.es/sites/default/files/20210204_tarifas_2021_visado.pdf.

Parte III

Presupuesto

Presupuesto

P.1. Introducción

En las secciones siguientes se desarrolla el valor monetario asignado a las distintas partidas presupuestarias consideradas en este trabajo.

- Recursos materiales.
 - Recursos hardware.
 - Recursos software.
- Recursos humanos.
- Redacción del documento.
- Derechos de visado del COIT (Colegio Oficial Ingenieros de Telecomunicaciones).
- Gastos de tramitación y envío.

Analizados estos puntos, se aplicarán los impuestos y se calculará el coste total.

P.2. Recursos Materiales

Con este concepto se agrupan los gastos que vienen dados por el uso de los recursos hardware y software. La forma de modelar el coste final será mediante la amortización lineal (ecuación P.1), teniendo en cuenta que no se tomará en consideración el valor de mantenimiento y que el valor residual (valor de los elementos al transcurrir su vida útil) es nulo. El coste final solo tendrá en cuenta el periodo de uso de cada uno de los equipos durante el transcurso del TFM.

$$C_f = \frac{V_{ad} - V_{res}}{A_u} \quad (\text{P.1})$$

Donde:

C_f es el coste final, V_{ad} el valor de adquisición, V_{res} el valor residual y A_u los años de vida útil.

P.2.1. Equipos Hardware

Para calcular los costes se tiene en cuenta que los equipos fueron adquiridos antes de la realización de este TFM. Se considera que la vida útil de las cámaras y el ordenador usado es de 6 años, la vida útil de la placa Purethermal 2 y el módulo USB3 para la Tau 2 es de 4 años. Mientras que para la placa vitrocerámica se tomará una vida útil de 2 años. El resultado de los costes asociados a los recursos materiales hardware son los que se muestran en la Tabla P.1.

Tabla P.1: Coste de los recursos hardware

Recursos Hardware				
Descripción	Tiempo	Valor de adquisición	Amortización anual	Coste final
FLIR Lepton 3.5	3 meses	175.84 €	29.31 €	9.77 €
PureThermal 2	3 meses	89.85 €	14.98 €	4.99 €
FLIR A615	2 meses	16698.0 €	4174.5 €	2087.25 €
FLIR A35	2 meses	4999.0 €	833.17 €	416.58 €
FLIR Tau 2	2 meses	7619.0 €	1269.83 €	634.92 €
Módulo USB3 para la FLIR Tau 2	2 meses	400.0 €	100 €	50 €
Vitrocerámica PCE-5000	2 meses	39.9 €	19.95 €	9.98 €
Portátil msi	4 meses	779.0 €	129.83 €	32.46 €
Total:		30800.59 €	6571.57 €	3245.94 €

Por consiguiente, el coste final para equipos hardware es de TRES MIL DOSCIENTOS CUARENTA Y CINCO EUROS CON NOVENTA Y CUATRO CÉNTIMOS.

P.2.2. Equipos Software

Para la realización de este TFM se dispusieron los recursos software mostrados en la Tabla P.2. Como se hizo en la parte de recursos hardware, el coste final depende del tiempo de utilización de cada uno de ellos durante el período de trabajo correspondiente, considerándose en este caso una vida útil de 3 años para cada uno de ellos.

Tabla P.2: Coste de los recursos software

Recursos Software				
Descripción	Tiempo	Valor de adquisición	Amortización anual	Coste final
MATLAB R2020b	4 meses	2000.00 €	666.67 €	222.22 €
SO Linux OpenSUSE	4 meses	0.00 €	0.00 €	0.00 €
SO Windows 10	4 meses	145.0 €	48.33 €	12.08 €
Microsoft Office	4 meses	67.0 €	22.33 €	5.58 €
Pycharm	4 meses	0.00 €	0.00 €	0.00 €
SDK FLIR Lepton	4 meses	0.00 €	0.00 €	0.00 €
SDK FLIR A615	4 meses	0.00 €	0.00 €	0.00 €
SDK FLIR A35	4 meses	0.00 €	0.00 €	0.00 €
SDK FLIR Tau 2	4 meses	0.00 €	0.00 €	0.00 €
Total:		2212 €	6737.33 €	239.89 €

Por consiguiente, el coste final para equipos software es de DOSCIENTOS TREINTA Y NUEVE EUROS CON OCHENTA Y NUEVE CÉNTIMOS.

P.3. Recursos Humanos

Este es el coste asociado al proyectado, que ha empleado 4 meses a media jornada en las tareas de estudio previo, integración de la cámara, realización de medidas y do-

cumentación. Para el cálculo se ha tenido en cuenta que el proyecto se desarrolla dentro del entorno de investigación de la Universidad de Las Palmas de Gran Canaria, el tipo de tarifa aplicada es la correspondiente al personal investigador técnico (máster) según la tabla de contrataciones de personal con cargo de proyectos, programas, convenios y contratos, correspondiente al BOULPGC de julio de 2021 [99]. Así, el coste a tiempo completo (7.5 horas diarias) es el que se muestra en la Tabla P.3.

Tabla P.3: Coste de los recursos humanos

Recursos Humanos			
Personal	Coste total mensual	Tiempo	Total
Ingeniero	1801.02 €	4 meses	7204.08 €

El coste total de recursos humanos es de SIETE MIL DOSCIENTOS CUATRO EUROS CON OCHO CÉNTIMOS.

P.4. Redacción del documento

El importe de la redacción del proyecto se calcula de acuerdo con la siguiente expresión:

$$R = 0.07 \cdot P \cdot C_n \quad (\text{P.2})$$

Donde:

P es el presupuesto del proyecto.

C_n es el coeficiente de ponderación en función del presupuesto. En este caso el coeficiente C_1 está fijado a 1.00 para presupuestos menores a 30.050 €.

En la Tabla P.4 se muestra el presupuesto calculado hasta el momento. Por lo que los costes derivados de la redacción del documento se obtienen sustituyendo los valores en la ecuación P.2.

Tabla P.4: Presupuesto parcial

Recurso	Coste
Recursos Materiales Hardware	3245.94 €
Recursos Materiales Software	239.89 €
Recursos Humanos	7204.08 €
Total:	10689.91 €

$$R = 0.07 \times 10689.91 \times 1 = 748.30 \text{ €}$$

El coste de redacción asciende a un total de SETECIENTOS CUARENTA Y OCHO EUROS CON TREINTA CÉNTIMOS.

P.5. Derechos de visado del COIT

Para proyectos de carácter general [100], los derechos de visado se calculan de acuerdo con la siguiente fórmula:

$$0.006 \cdot P_1 \cdot C_1 + 0.003 \cdot P_2 \cdot C_2 \quad (\text{P.3})$$

Donde:

P_1 = Presupuesto general para este proyecto.

P_2 = Presupuesto de ejecución material que corresponde a la obra civil.

C_1 = Coeficientes Reductores (C) correspondientes a P1.

C_2 = Coeficientes Reductores (C) correspondientes a P2.

Por lo dicho anteriormente el coeficiente C_1 está fijado a 1.00. En este caso, para el desarrollo de este TFM no se ha requerido obra civil por lo que el valor de P_2 es 0.00 €. De la misma manera, el valor del presupuesto P_1 se calcula sumando los costes de las secciones anteriores. Esto viene incluido en la Tabla P.5.

Tabla P.5: Presupuesto con el coste de redacción

Recurso	Coste
Recursos Materiales Hardware	3245.94 €
Recursos Materiales Software	239.89 €
Recursos Humanos	7204.08 €
Redacción del documento	748.30 €
Total:	11438.21 €

Se sustituyen los valores en la ecuación P.3.

$$R = 0.006 \times 11438.21 \times 1 = 686.29 \text{ €}$$

El coste por derecho de visado es de SEISCIENTOS OCHENTA Y SEIS EUROS CON VEINTINUEVE CÉNTIMOS.

P.6. Gastos de tramitación y envío

Cada documento visado por vía telemática tiene un coste de 6.00 € según [100].

P.7. Aplicación de impuestos y coste final

El resultado final con todos los apartados desglosados en la Tabla P.6, estableciendo el coste total del trabajo presentado.

Tabla P.6: Presupuesto total

Presupuesto Final	
Partidas	Totales
Recursos Materiales	
- Recursos Hardware	3245.94 €
- Recursos Software	239.89 €
Recursos Humanos	7204.08 €
Total parcial:	10689.91 €
Redacción del documento	748.64 €
Visado del COIT	686.29 €
Gastos de tramitación y envío	6.0 €
Total:	12130.49 €
IGIC (7 %):	849.13 €
Total después de impuestos:	12979.63 €

El presupuesto total teniendo en cuenta los impuestos asciende a DOCE MIL NOVECIENTOS SETENTA Y NUEVE EUROS CON SESENTA Y TRES CÉNTIMOS.

Las Palmas de Gran Canaria a 27 de enero de 2022

Fdo.:

Parte IV

Anexos

Anexo A

Resultado visual de los métodos

En este anexo se presentarán distintas imágenes donde se apreciará el resultado de aplicar todos los métodos de súper-resolución espaciales nombrados en este TFM, tanto para la cámara A615 como Lepton. Para la representación de los resultados se utilizará la misma imagen de la base de datos DASSR que se usó en el capítulo de resultados.

A.1. Cámara A615

Los resultados de los métodos de reconstrucción para la cámara A615 se muestran en las Figuras A.1, A.2, A.3, A.4 y A.5.

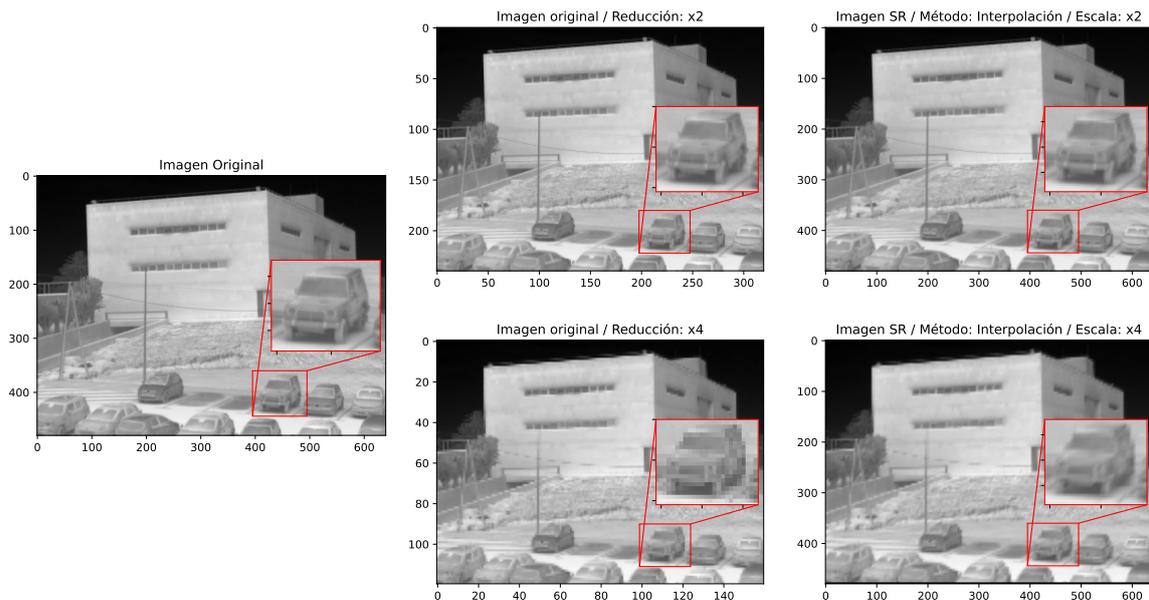


Figura A.1: Resultados del método de interpolación en la cámara A615

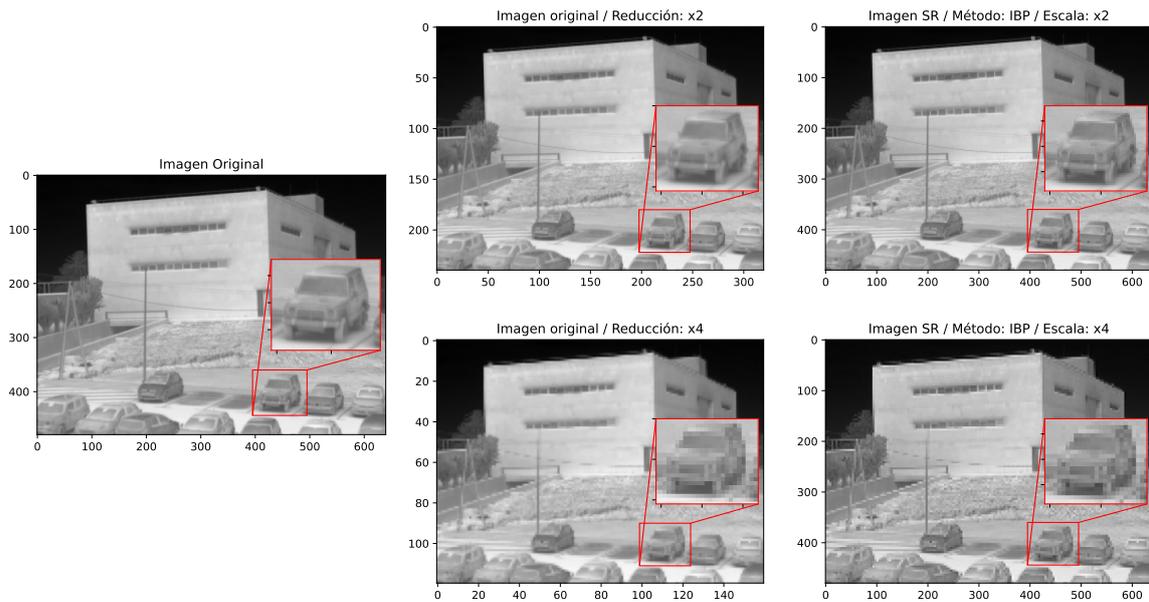


Figura A.2: Resultados del método IBP en la cámara A615

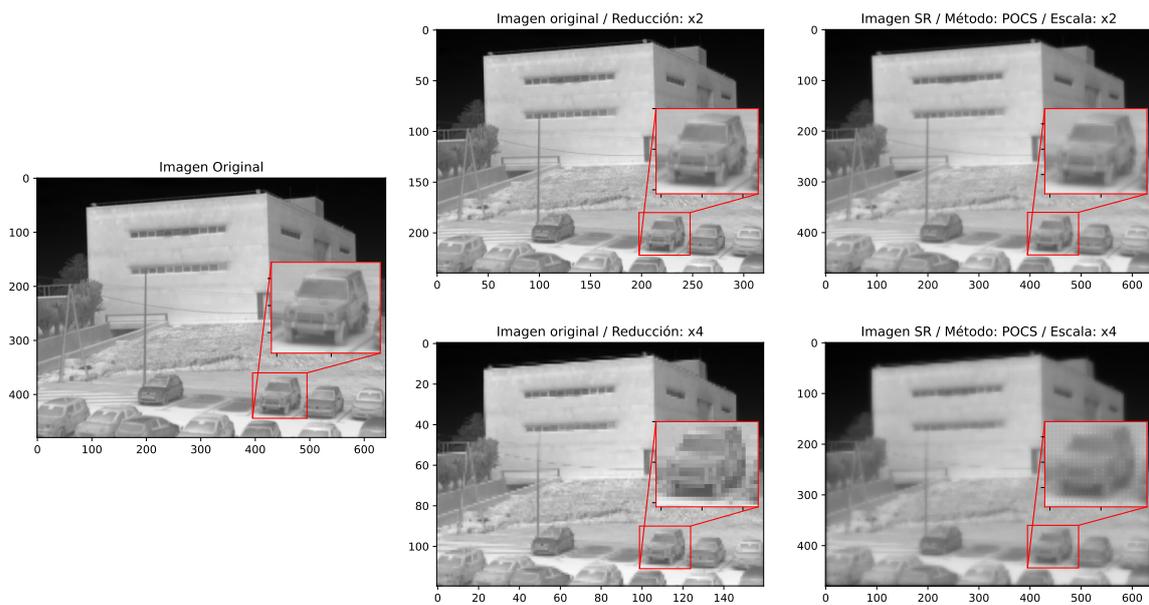


Figura A.3: Resultados del método POCS en la cámara A615

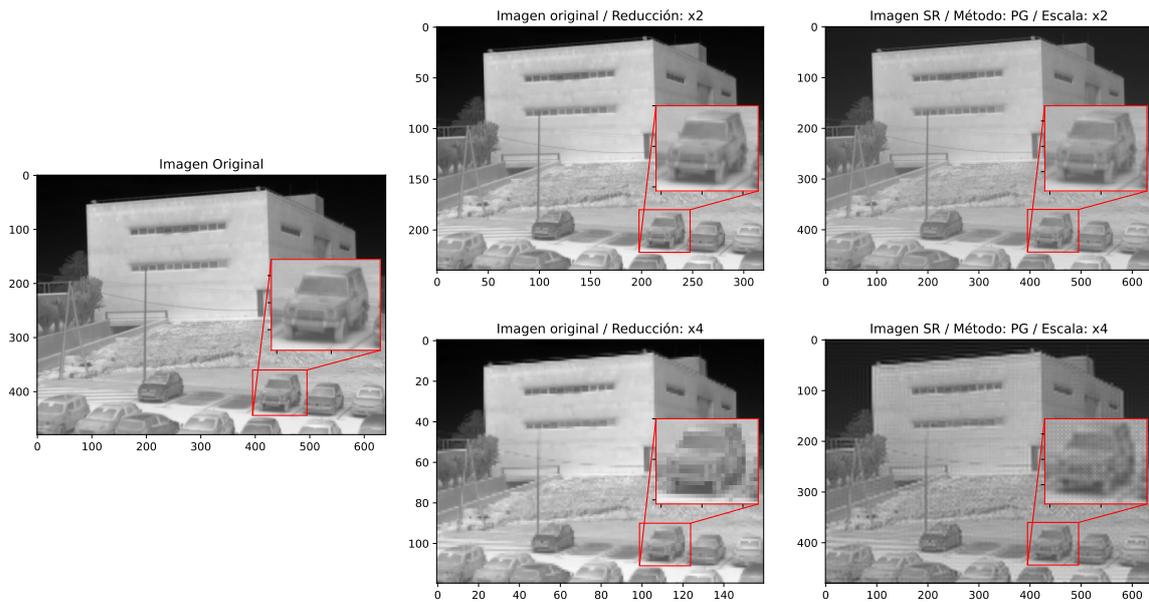


Figura A.4: Resultados del método de Papoulis-Gerchberg en la cámara A615

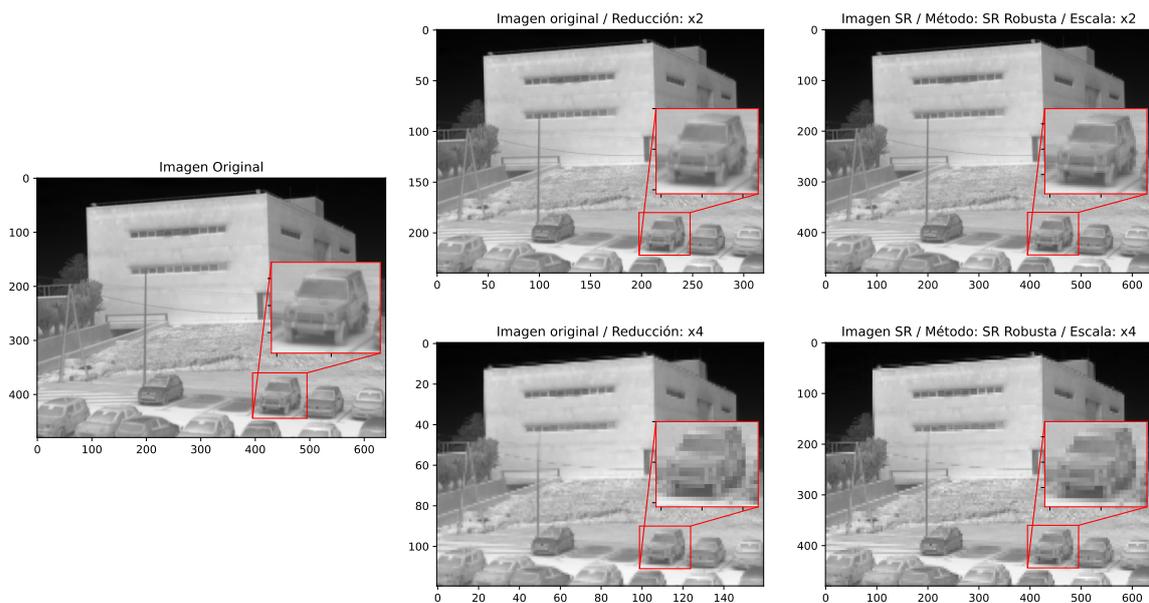


Figura A.5: Resultados del método de súper-resolución robusta en la cámara A615

ANEXO A. RESULTADO VISUAL DE LOS MÉTODOS

Los resultados de los métodos basados en redes neuronales para la cámara A615 se muestran en las Figuras A.6, A.7, A.8, A.9, A.10, A.11, A.12, A.13, A.14 y A.15.

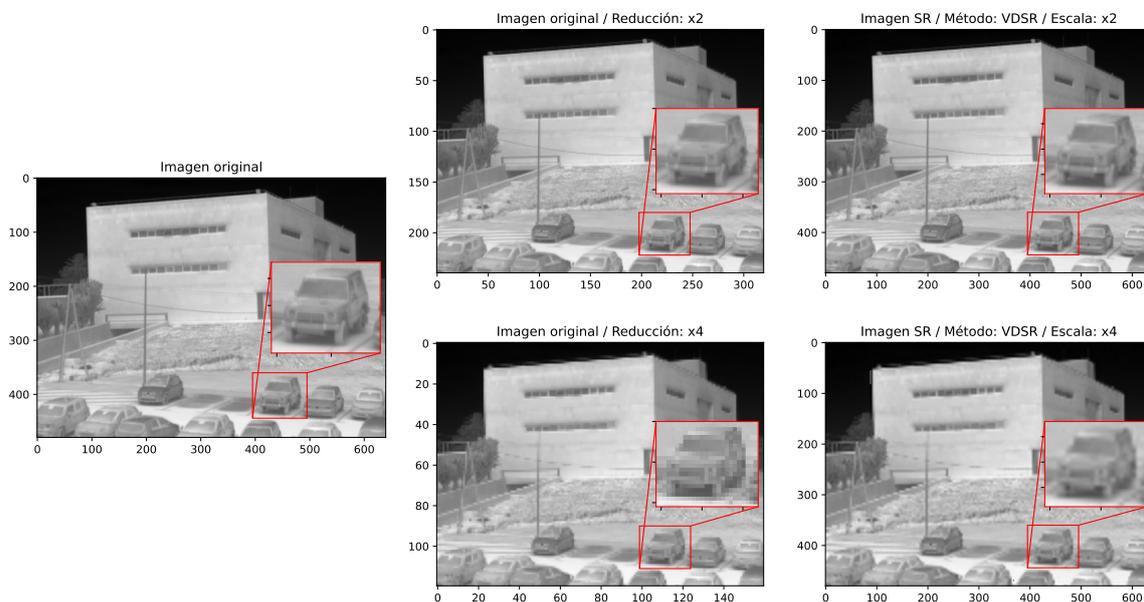


Figura A.6: Resultados del método VDSR en la cámara A615

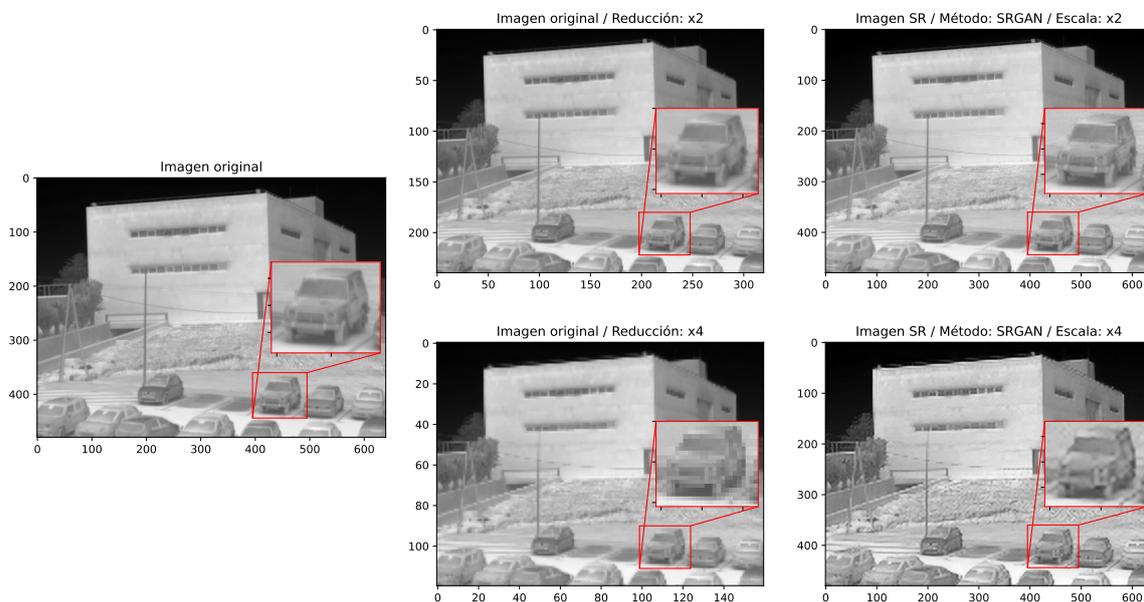


Figura A.7: Resultados del método SRGAN en la cámara A615

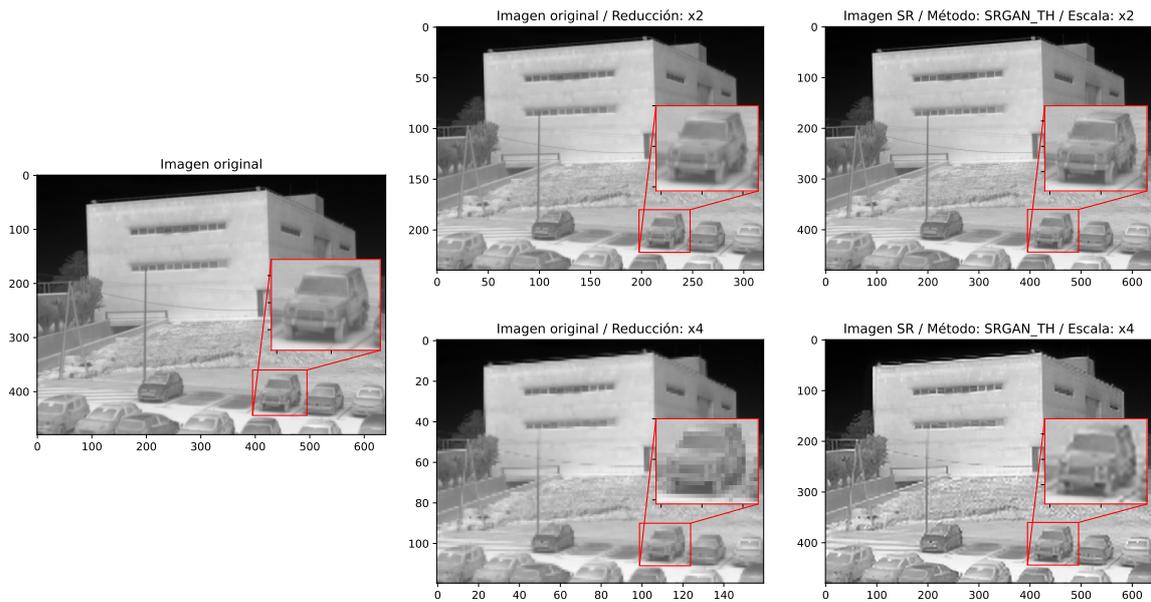


Figura A.8: Resultados del método SRGAN_TH en la cámara A615

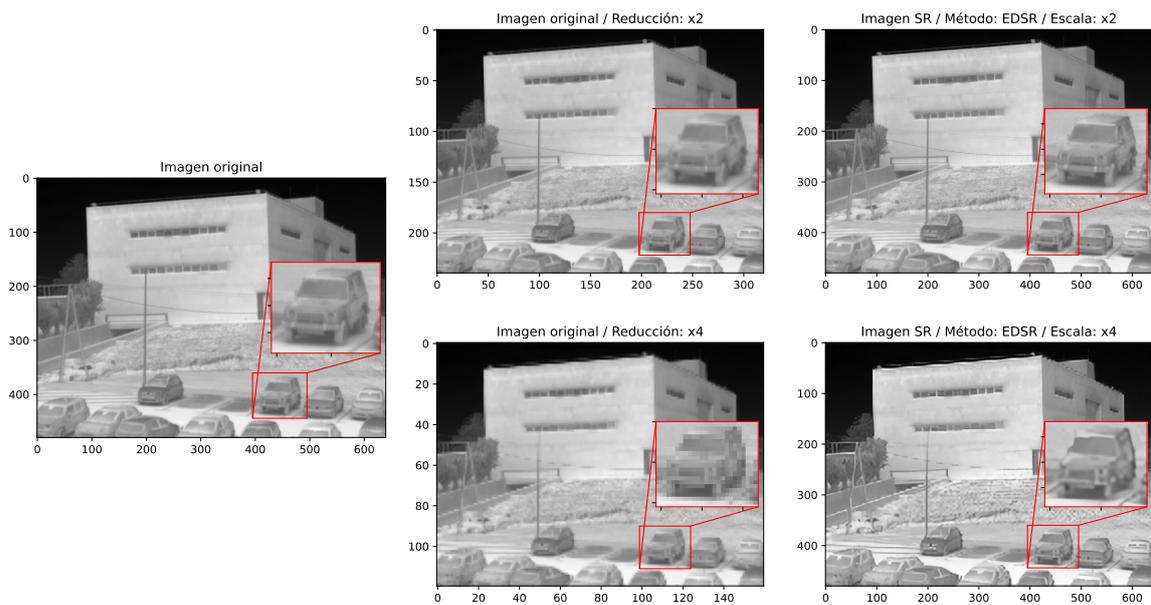


Figura A.9: Resultados del método EDSR en la cámara A615

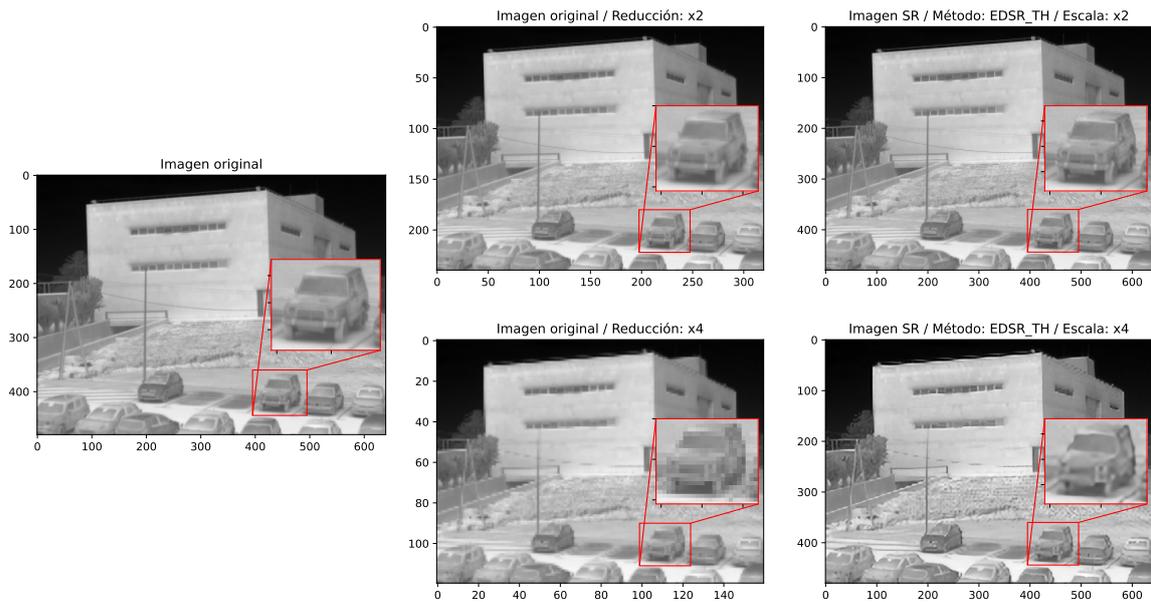


Figura A.10: Resultados del método EDSR_TH en la cámara A615

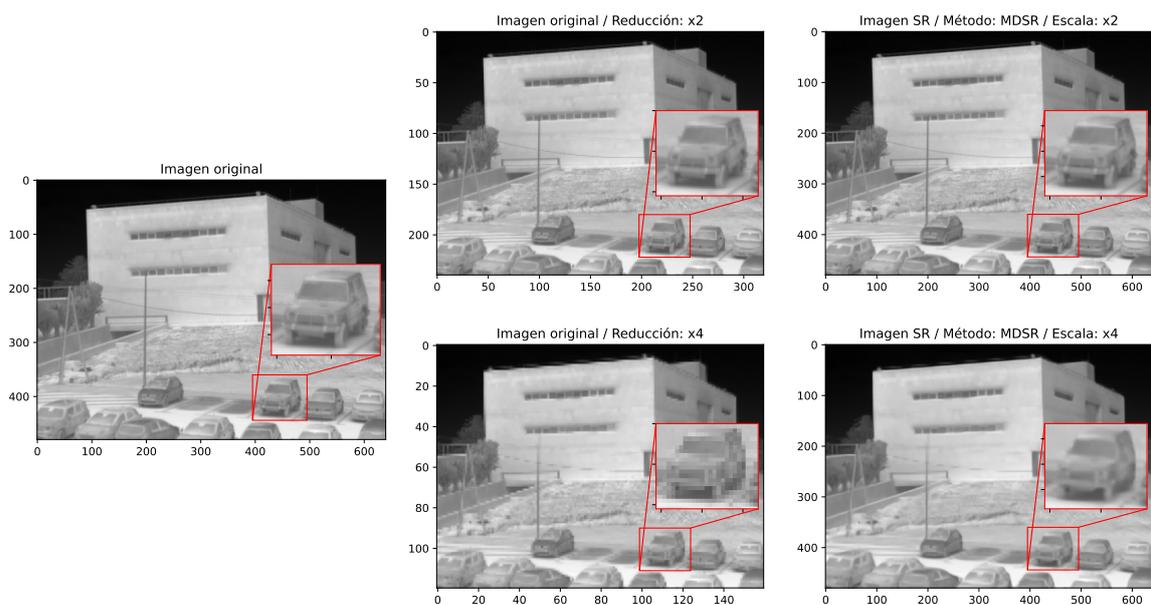


Figura A.11: Resultados del método MDSR en la cámara A615

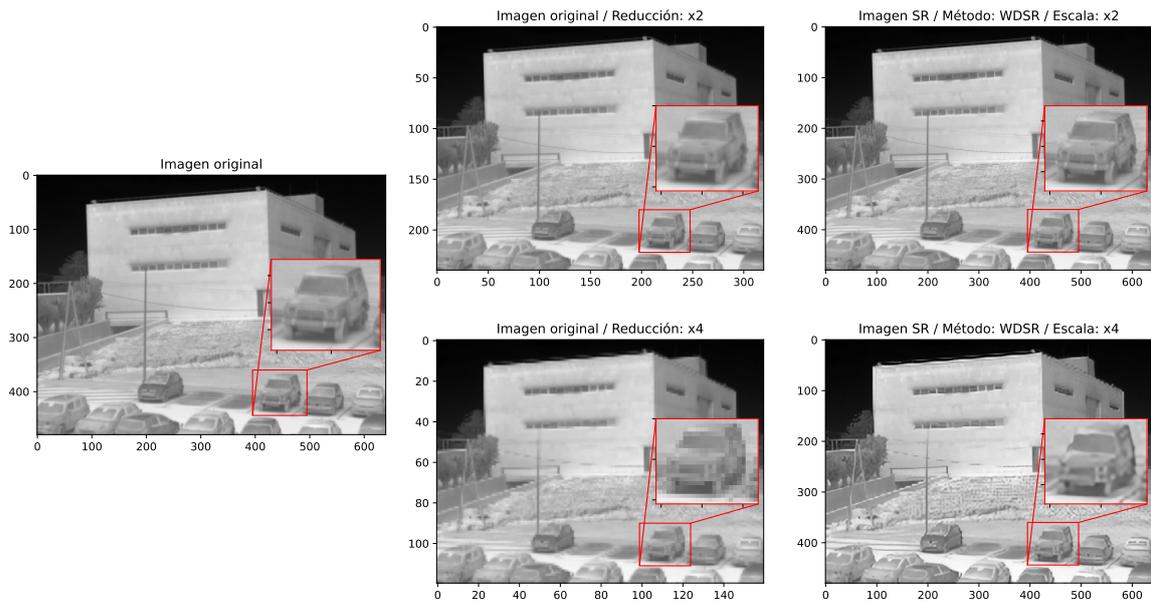


Figura A.12: Resultados del método WDSR en la cámara A615

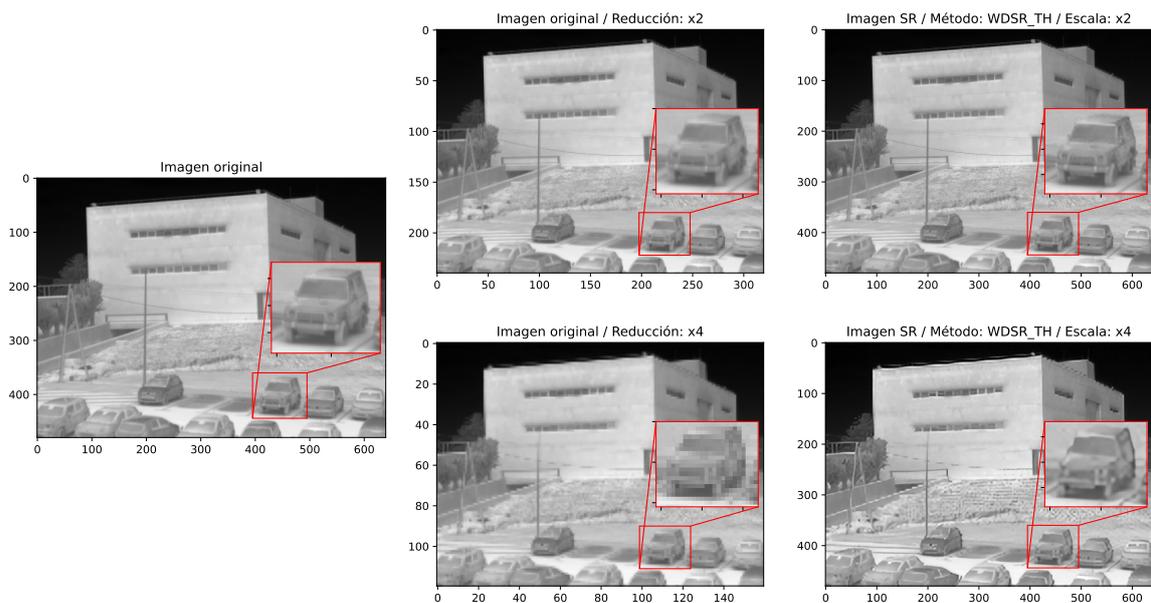


Figura A.13: Resultados del método WDSR_TH en la cámara A615

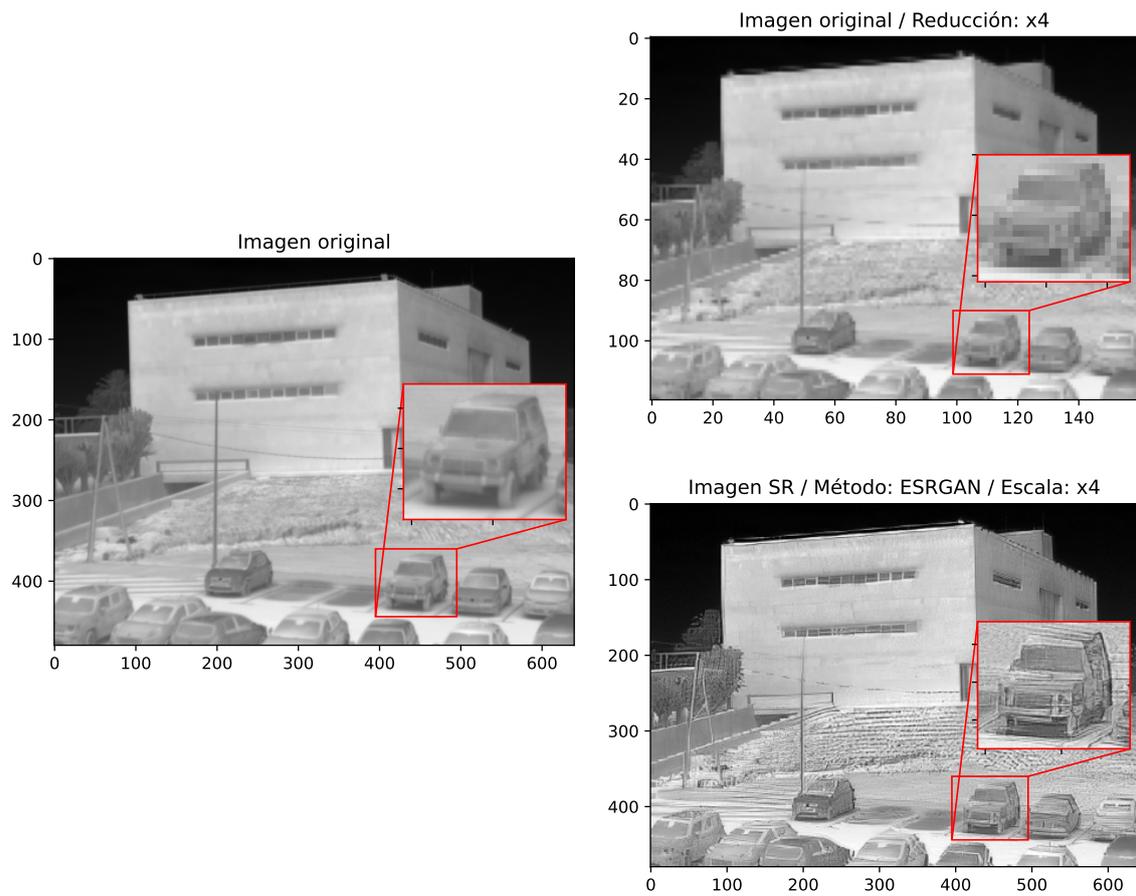


Figura A.14: Resultados del método ESRGAN en la cámara A615

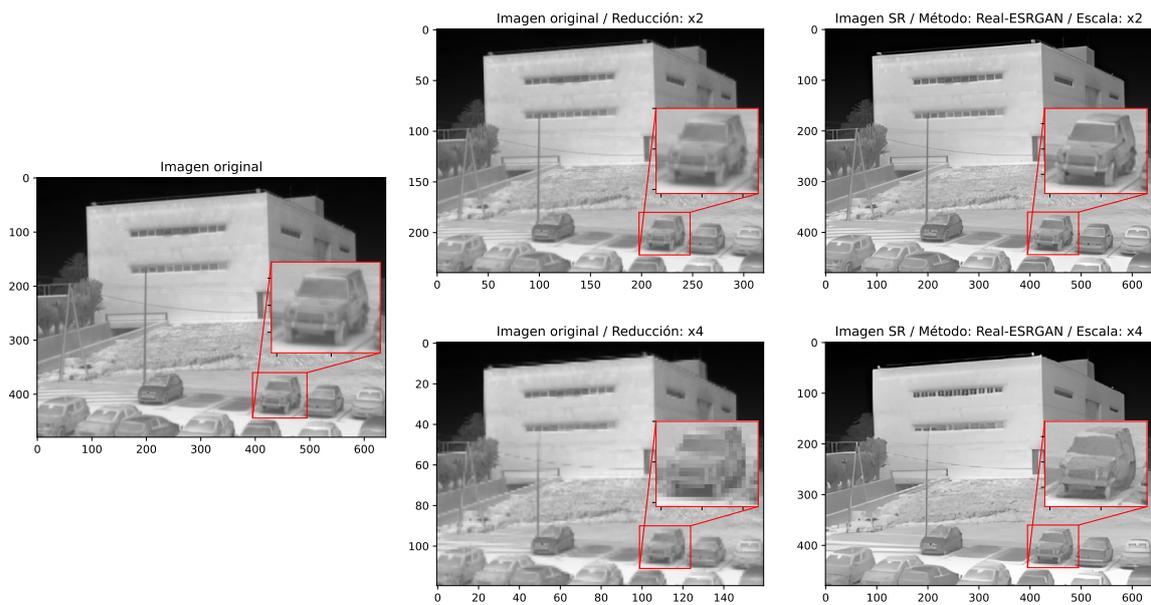


Figura A.15: Resultados del método Real-ESRGAN en la cámara A615

A.2. Cámara Lepton

Los resultados de los métodos de reconstrucción para la cámara Lepton se muestran en las Figuras A.16, A.17, A.18, A.19 y A.20.

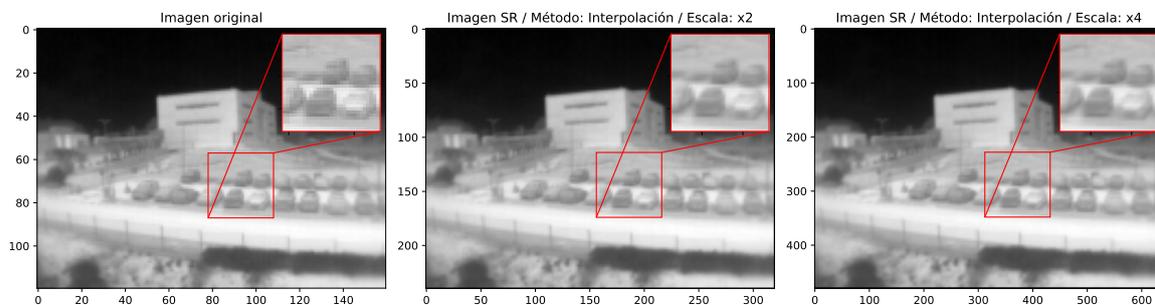


Figura A.16: Resultados del método de interpolación en la cámara Lepton

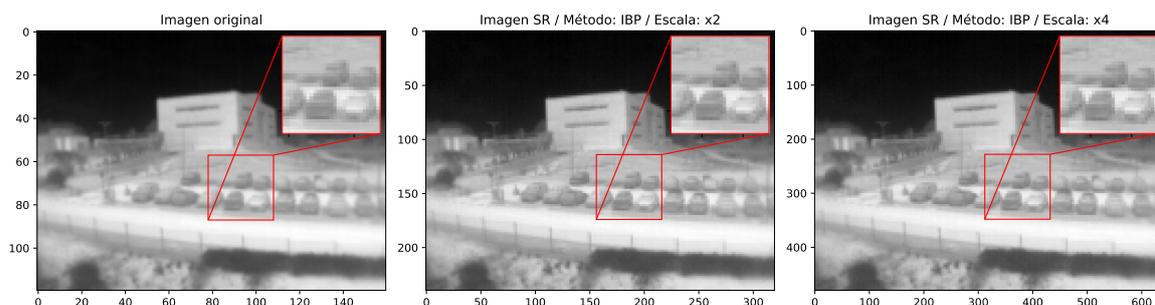


Figura A.17: Resultados del método IBP en la cámara Lepton

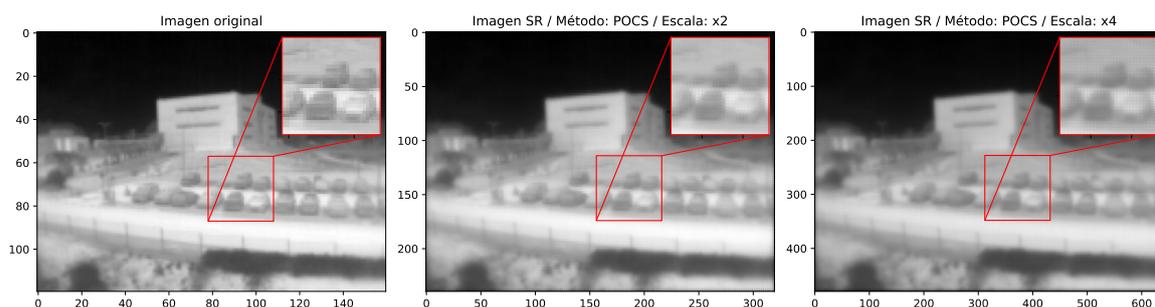


Figura A.18: Resultados del método POCS en la cámara Lepton

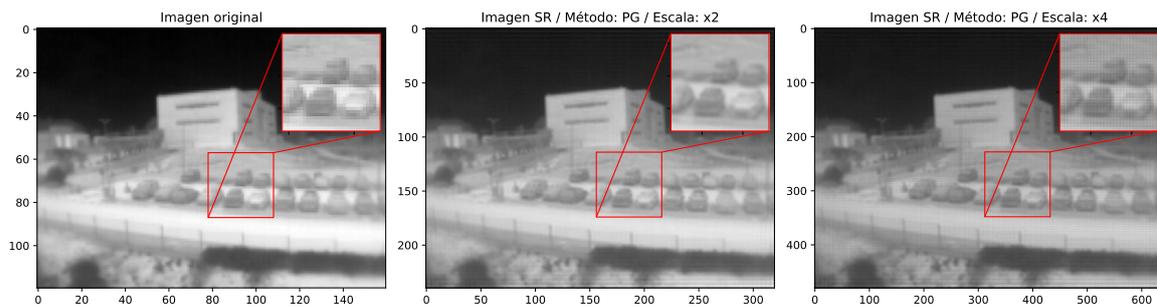


Figura A.19: Resultados del método de Papoulis-Gerchberg en la cámara Lepton

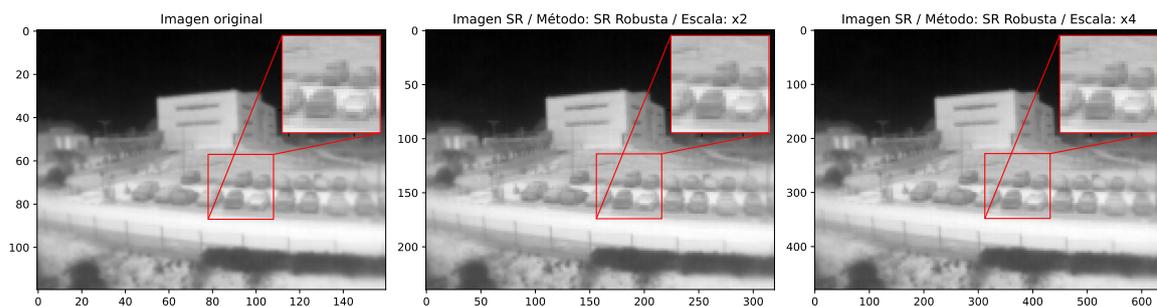


Figura A.20: Resultados del método de súper-resolución robusta en la cámara Lepton

Los resultados de los métodos basados en redes neuronales para la cámara Lepton se muestran en las Figuras A.21, A.22, A.23, A.24, A.25, A.26, A.27, A.28, A.29 y A.30.

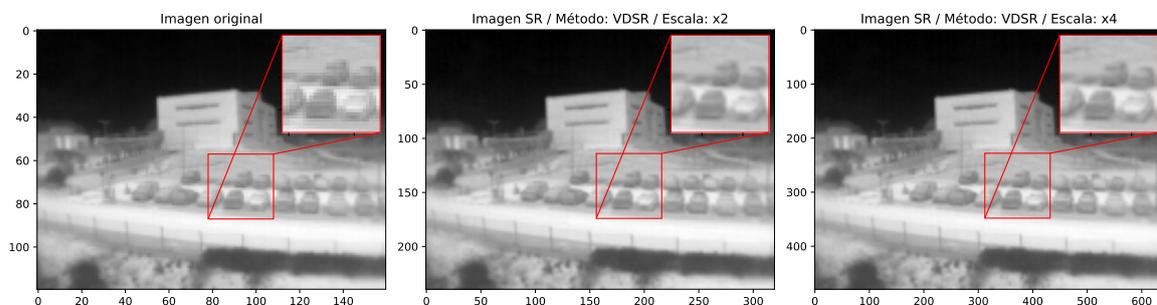


Figura A.21: Resultados del método VDSR en la cámara Lepton

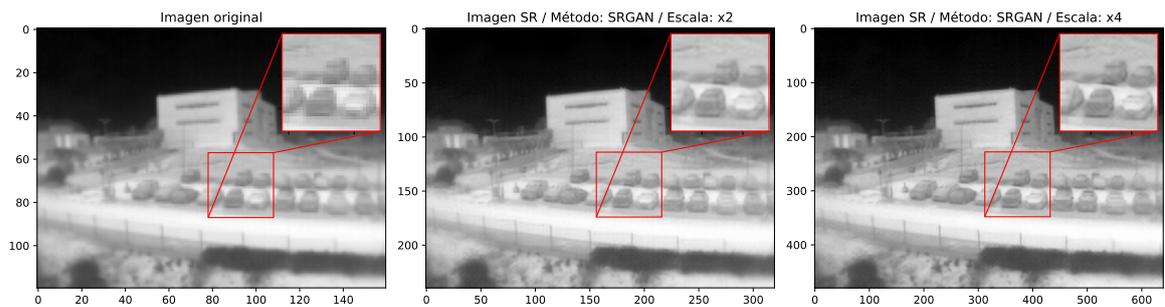


Figura A.22: Resultados del método SRGAN en la cámara Lepton

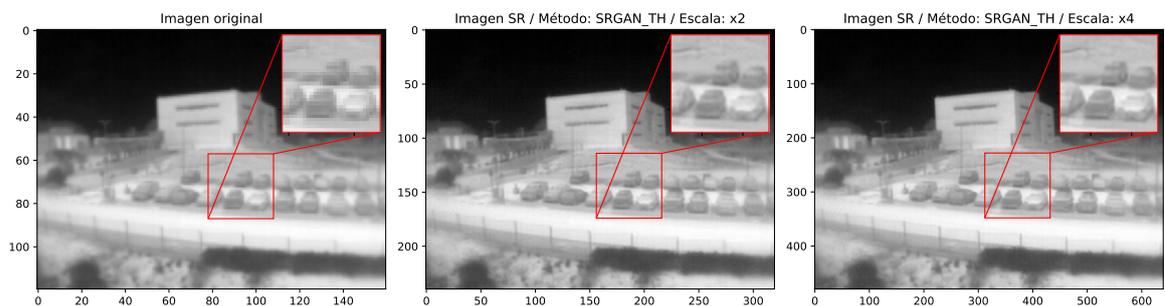


Figura A.23: Resultados del método SRGAN_TH en la cámara Lepton

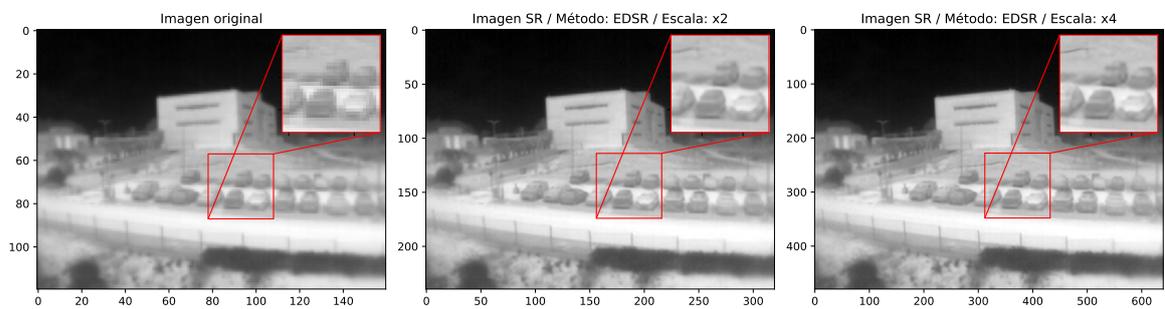


Figura A.24: Resultados del método EDSR en la cámara Lepton

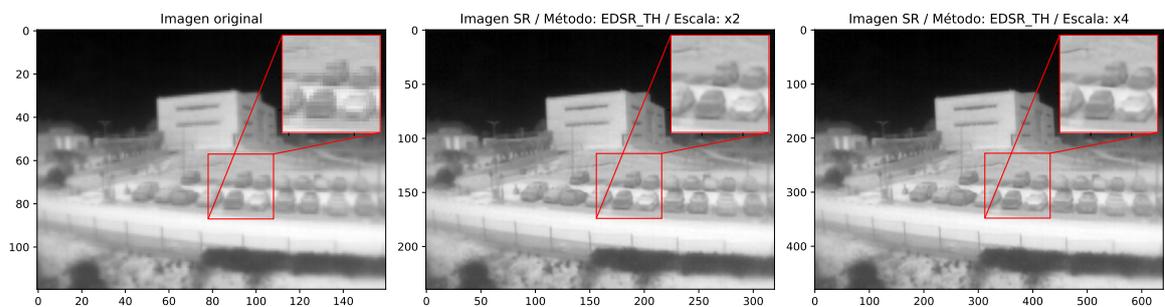


Figura A.25: Resultados del método EDSR_TH en la cámara Lepton

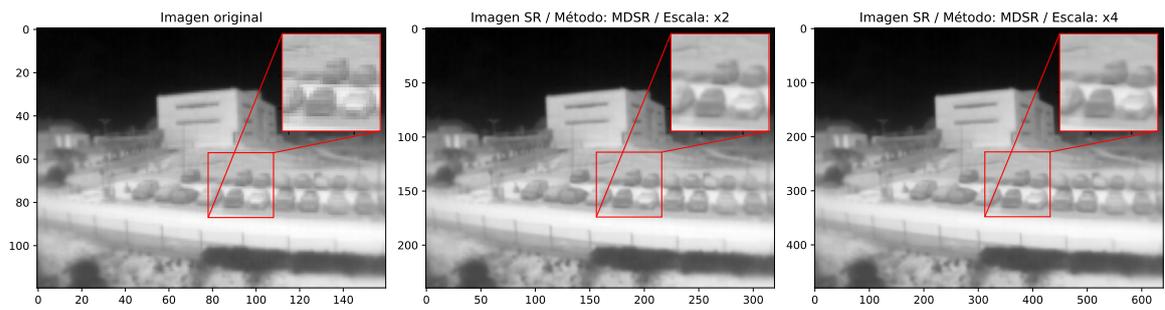


Figura A.26: Resultados del método MDSR en la cámara Lepton

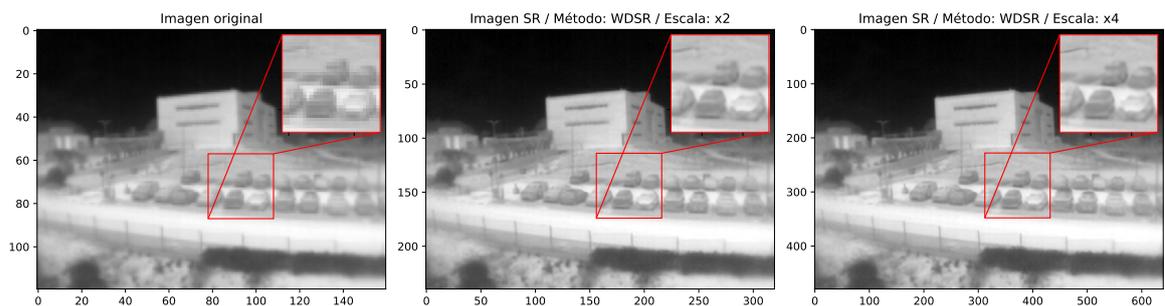


Figura A.27: Resultados del método WDSR en la cámara Lepton

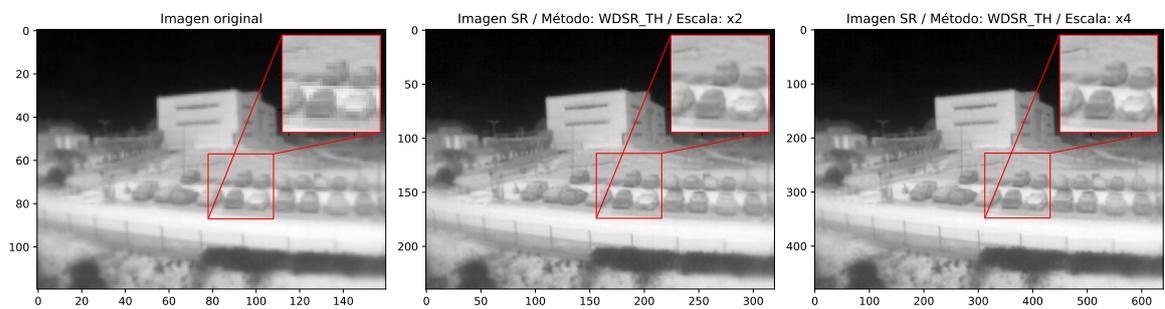


Figura A.28: Resultados del método WDSR_TH en la cámara Lepton

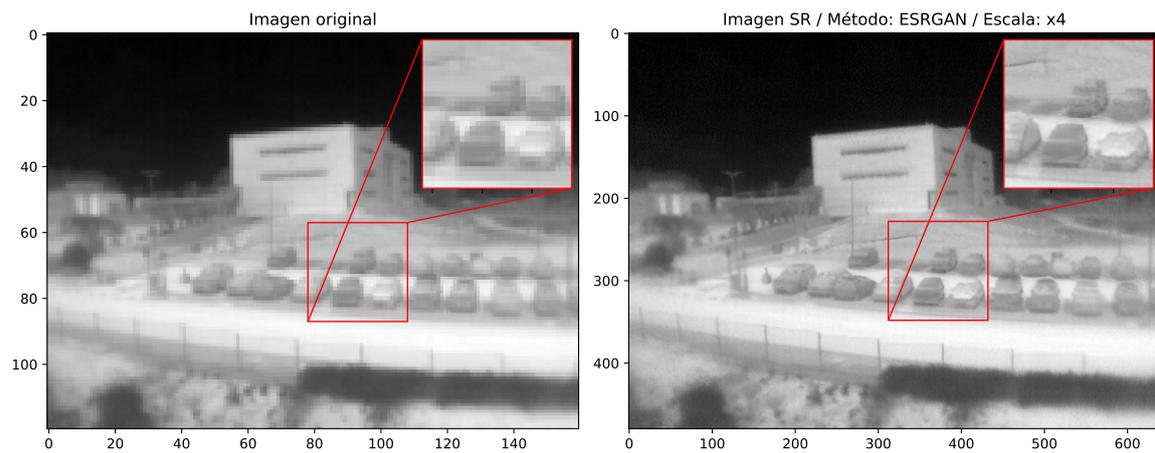


Figura A.29: Resultados del método ESRGAN en la cámara Lepton

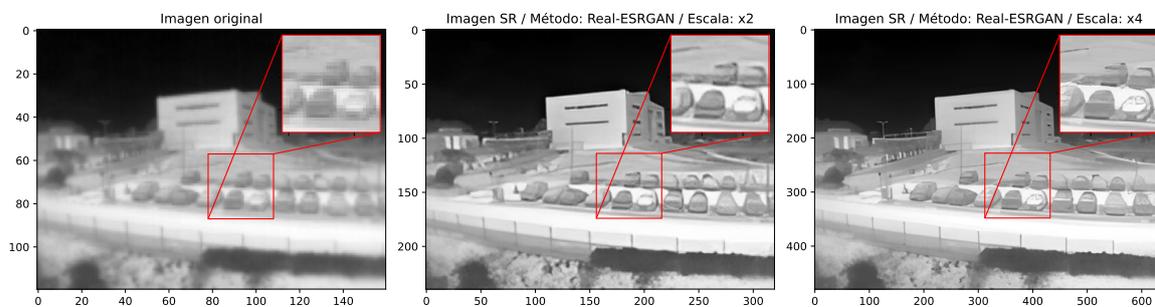


Figura A.30: Resultados del método Real-ESRGAN en la cámara Lepton

Anexo B

Creación del demostrador

En este anexo se muestra el diseño e implementación de un demostrador a partir del lenguaje Python para aplicar los métodos de súper-resolución espacial basados en redes neuronales y el modelo propio para realizar la súper-resolución radiométrica. Este demostrador ha sido implementado a través de la librería PyQt, en su versión PyQt5, que ofrece una serie de facilidades a la hora de crear una interfaz visual para una gran variedad de aplicaciones. En este aspecto, una herramienta que facilita el diseño del demostrador es *Qt Designer*, que no es más que la herramienta de Qt para diseñar y construir interfaces gráficas de usuario. Esta herramienta es mostrada en la Figura B.1.

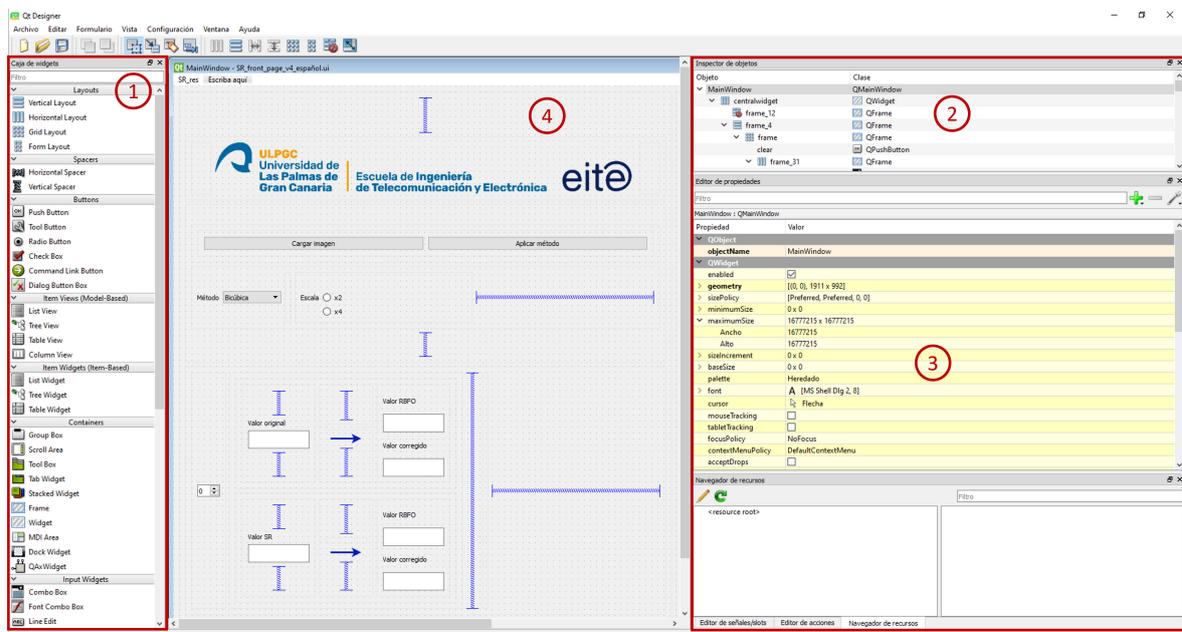


Figura B.1: Visualización de la herramienta *Qt Designer*

En la figura anterior se observan las ventanas disponibles que permiten la creación de distintos componentes (1), su modificación de parámetros (2) y (3) y la visualización general de la interfaz gráfica (4), entre otros factores.

El primer vistazo al demostrador se ilustra en la Figura B.2, donde se pueden observar los componentes principales que conforman la interfaz visual diseñada. Por un lado, el recuadro rojo (1) muestra el logo institucional de la ULPGC y de la EITE (Escuela de Ingeniería de Telecomunicación y Electrónica). El recuadro (2) se compone por el lienzo o *canvas* de la librería matplotlib, formando el *backend* que permitirá la visualización de las distintas imágenes usadas en este TFM. De la misma forma, los recuadros (3) y (4) representan los componentes necesarios para aplicar y visualizar los distintos métodos de súper-resolución de este trabajo.

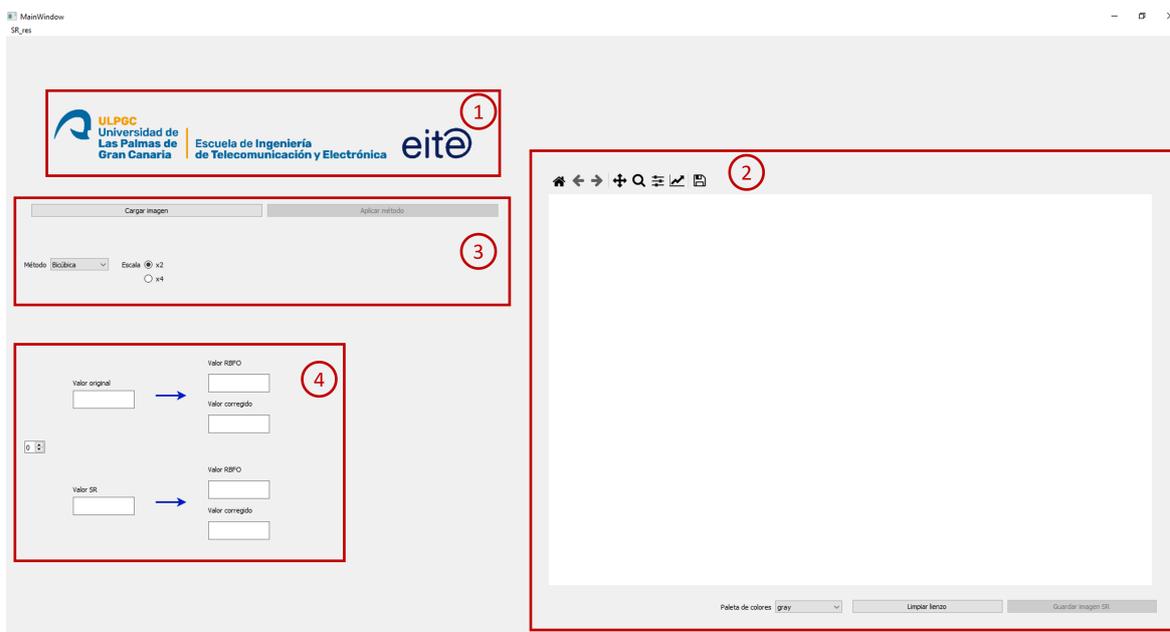


Figura B.2: Visualización general del demostrador

A partir de este punto se desglosarán las funciones de cada uno de los componentes de este demostrador. Por ejemplo, en la Figura B.3 se muestra cuál es el proceso de visualización de una imagen LR en el lienzo de matplotlib. El recuadro (1) se sitúa sobre el botón que se ha de presionar para que aparezca la ventana desplegable del recuadro (2), donde se podrá elegir la imagen de evaluación de la base de datos TID-II. En este caso, se ha escogido una imagen de la cámara Lepton a 5 metros de distancia de la superficie radiante estudiada en el capítulo 4.

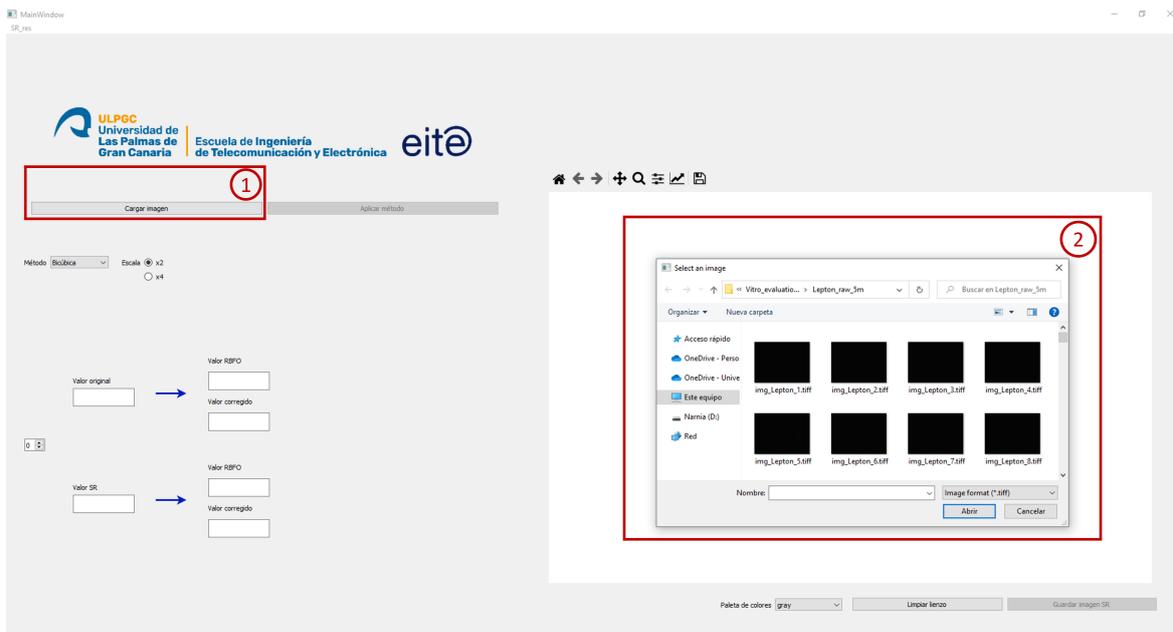


Figura B.3: Proceso de visualización de la imagen LR del demostrador

El siguiente paso en el flujo de trabajo propuesto es aplicar un método de súper-resolución espacial a la imagen LR. En la Figura B.4, más concretamente en el recuadro ①, se puede observar que la imagen LR queda representada en el lienzo de matplotlib.

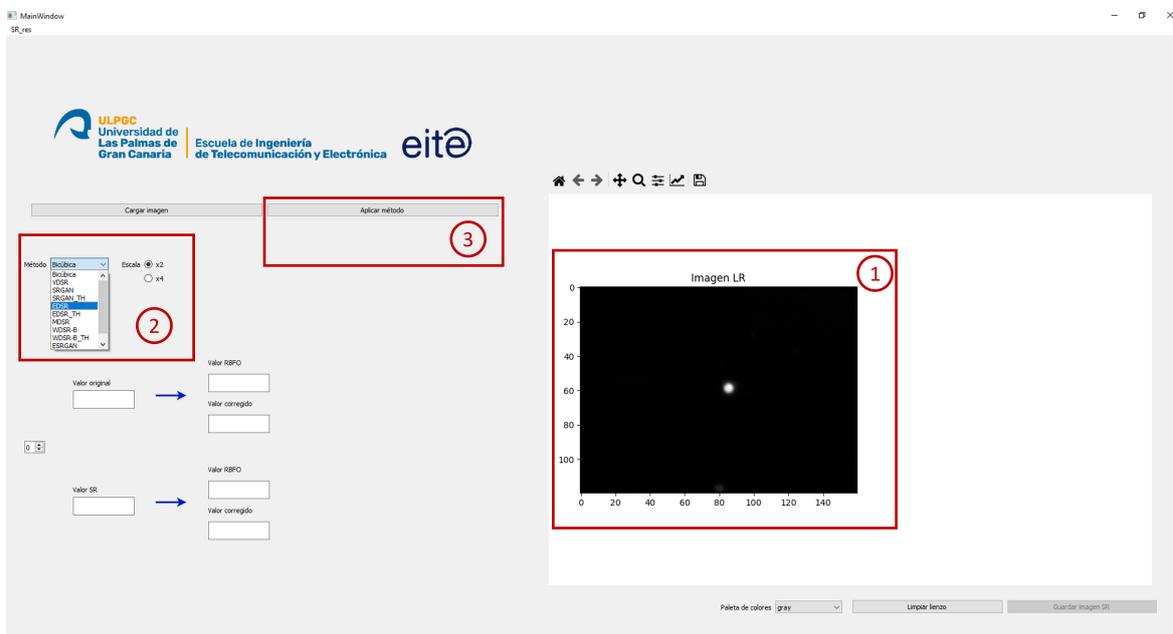


Figura B.4: Proceso de para aplicar la súper-resolución espacial del demostrador

De la misma manera, el recuadro ② muestra dos componentes, una lista desplegable con los métodos de súper-resolución implementados y unos botones para aplicar el factor de escala x2 o x4. Por último, el recuadro ③ presenta un botón que activa las

funciones de Python necesarias para aplicar cada uno de estos métodos y visualizar la imagen SR de salida al lado de la imagen LR de entrada.

Una vez se ha realizado la operación de súper-resolución se tiene algo parecido a lo mostrado en la Figura B.5, donde se puede ver la imagen SR en el recuadro (1). En este aspecto, para observar mejor algunas variaciones de intensidad en los píxeles que componen la imagen, se pueden elegir varias paletas de colores de matplotlib. Estas paletas de colores se han implementado en una lista desplegable dentro del recuadro (2), donde automáticamente, solo con elegir la paleta de la lista, el lienzo queda actualizado.

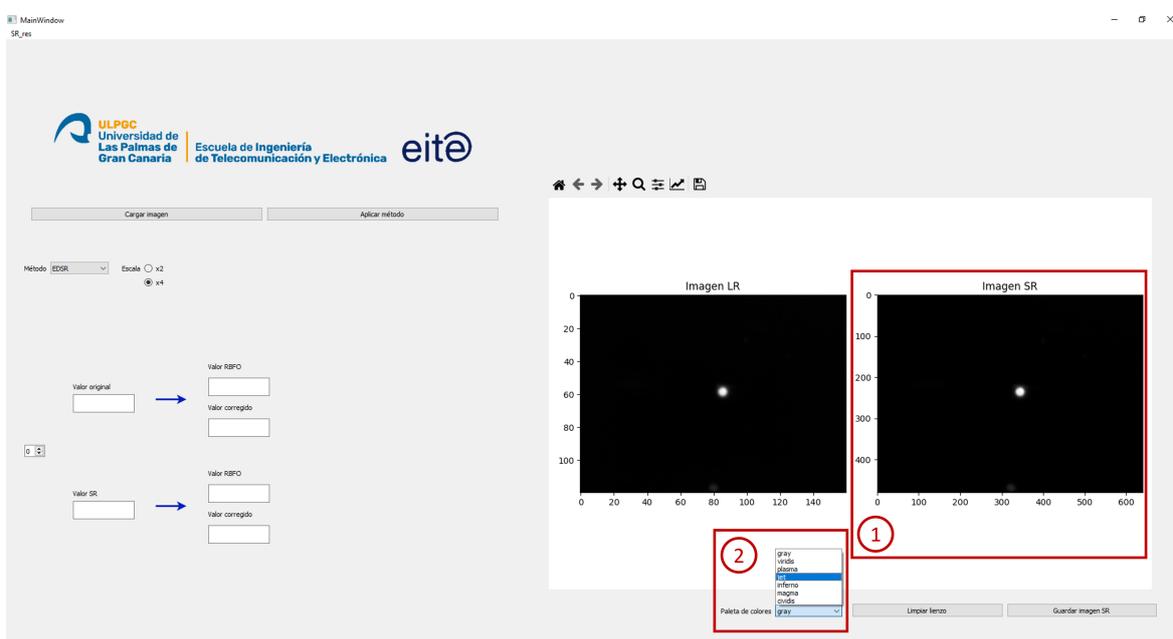


Figura B.5: Paletas de colores implementadas en el demostrador

El cambio de la paleta de colores se muestra en la Figura B.6, dentro del recuadro (1). En esta misma figura se resalta el último componente de importancia para el demostrador, este es el que se encarga de aplicar la ecuación RBFO y la súper-resolución radiométrica a píxeles dados en las imágenes LR y SR.

Para llevar a cabo este proceso, se han implementado funciones que permiten obtener el valor de un píxel concreto tanto de la imagen LR como SR. El valor radiométrico que ofrece la cámara Lepton en su formato de ganancia baja se muestra en los recuadros (2) y (4). Los recuadros (3) y (5) muestran los valores de temperatura estimados al aplicar la ecuación RBFO del fabricante FLIR con sus parámetros por defecto.



Figura B.6: Valores de los píxeles deseados y su transformación en temperatura (I)

Sin embargo, es en la Figura B.7 dónde se observa la implementación del modelo de súper-resolución radiométrica. En este caso, el recuadro (1) muestra un componente donde se puede aumentar o disminuir el término de distancia que entra a la red neuronal junto al valor del píxel.

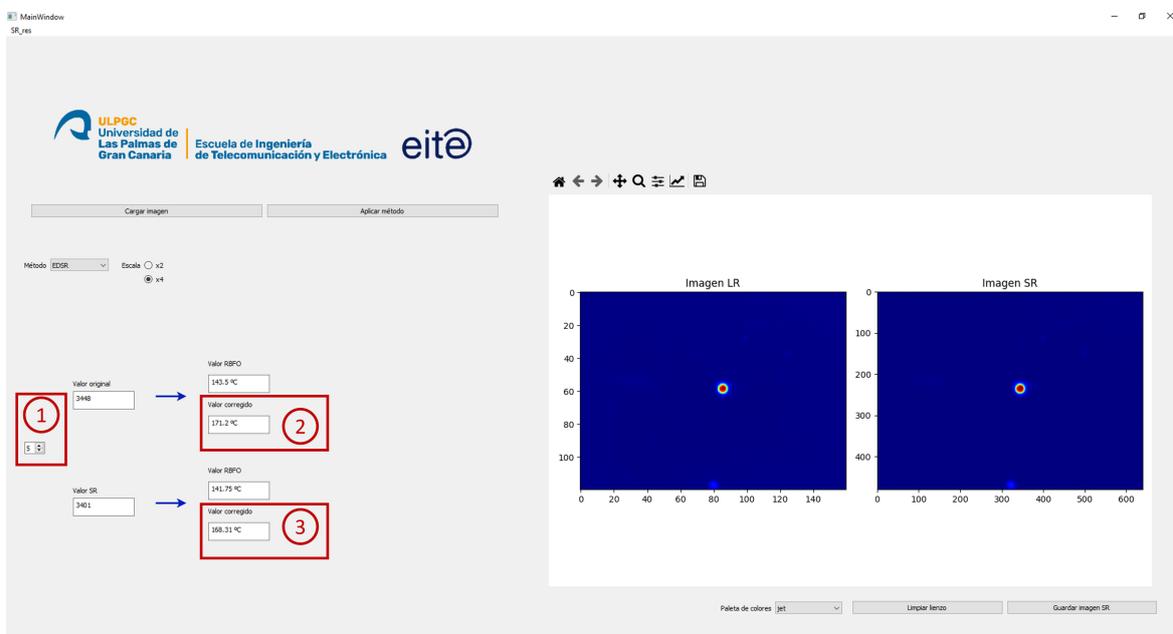


Figura B.7: Valores de los píxeles deseados y su transformación en temperatura (II)

Por otro lado, los recuadros (2) y (3) muestran cuál ha sido la predicción de la red neuronal casi a tiempo real tanto para la imagen LR como SR, observando de esta

manera la diferencia entre aplicar este modelo y la ecuación RBFO, así como la pérdida en los valores que existe al aplicar ciertos métodos de súper-resolución espacial.

Con la intención de almacenar la imagen SR resultante, se ha implementado una función que almacena los datos radiométricos en formato TIFF (*Tagged Image File Format*). El uso de esta función se observa en la Figura B.8, donde el recuadro ① muestra el botón que se encarga de llamar al método mientras que el recuadro ② señala el directorio donde se desea almacenar la imagen.

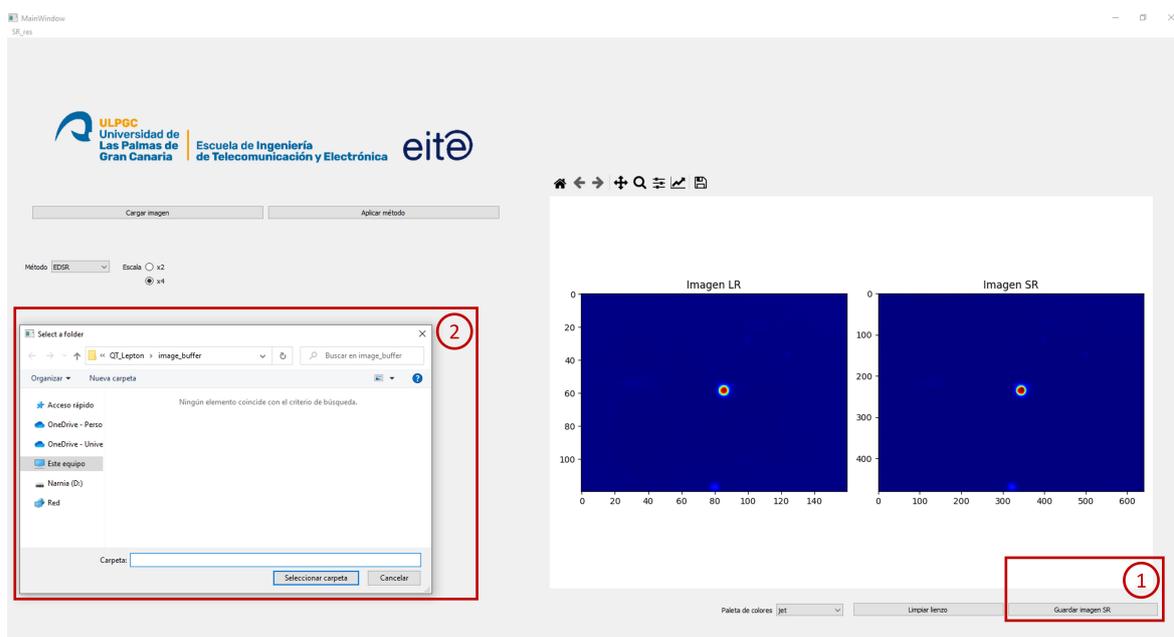


Figura B.8: Rutina para almacenar imágenes SR en el demostrador

Finalmente, cabe resaltar la capacidad de almacenar el lienzo actual gracias a la barra de navegación de matplotlib, así como la rutina implementada que borra todas las variables del demostrador e inicializa desde cero el lienzo, para poder empezar de nuevo la visualización si así se desea. Estas dos particularidades se muestran en los recuadros ①, ② y ③ de la Figura B.9 respectivamente.

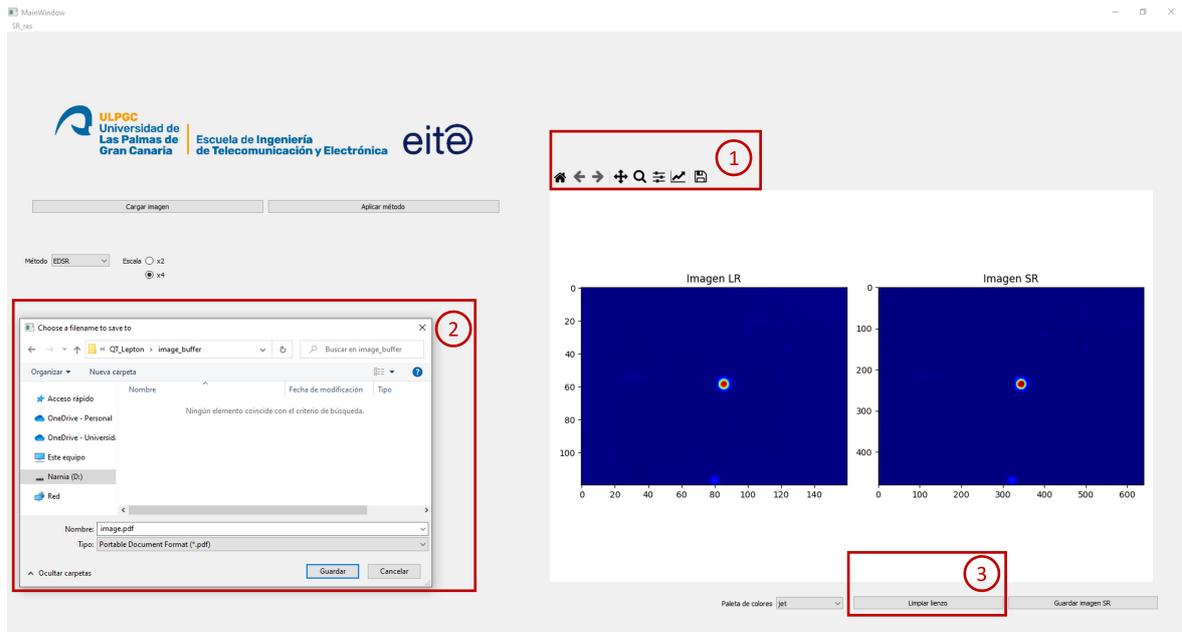


Figura B.9: Proceso de almacenamiento por parte de matplotlib y borrado del lienzo

