



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación
y Desarrollo Tecnológico

Tesis de Maestría

Desarrollo de un *FrameWork* para la
experimentación con algoritmos de
Súper Resolución

presentada por

Ing. Martha Pamela Cejudo García

como requisito para la obtención del grado
de

Maestra en Ciencias de la Computación

Director de tesis
Dr. Raúl Pinto Elías

Cuernavaca, Morelos, México. Enero de 2020.



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Centro Nacional de Investigación y Desarrollo Tecnológico
Departamento de Ciencias Computacionales

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Mor., 13/enero/2020

OFICIO No. DCC/012/2020

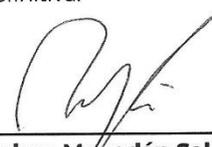
Asunto: Aceptación de documento de tesis
CENIDET-AC-004-M14-OFICIO

C. DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial de la **C. Ing. Martha Pamela Cejudo García**, con número de control M17CE087, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado "**Desarrollo de un FrameWork para la experimentación con algoritmos de Súper Resolución**" y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.



Dr. Raúl Pinto Elías
Doctor en Ciencias en la Especialidad de
Ingeniería Eléctrica
3890453
Director de tesis



Dra. Andrea Magadán Salazar
Doctora en Ciencias Computacionales
10654097
Revisor 1



Dr. José Ruiz Ascencio
Doctor en Ciencias
5009035
Revisor 2

C.c.p. Depto. Servicios Escolares.
Expediente / Estudiante
JGGS/lmz

Interior Internado Palmira S/N, Col. Palmira, C. P. 62490Cuernavaca, Morelos.
Tel. (01) 777 3 62 77 70, ext. 3202, e-mail: dcc@cenidet.edu.mx
www.tecnm.mx | www.cenidet.tecnm.mx

cenidet[®]
Centro Nacional de Investigación
y Desarrollo Tecnológico





EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Centro Nacional de Investigación y Desarrollo Tecnológico
Subdirección Académica

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Cuernavaca, Morelos, 14/enero/2020

OFICIO No. SAC/014/2020

Asunto: Autorización de impresión de tesis

ING. MARTHA PAMELA CEJUDO GARCÍA
CANDIDATA AL GRADO DE MAESTRA EN CIENCIAS
DE LA COMPUTACIÓN
P R E S E N T E

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado "*Desarrollo de un FrameWork para la experimentación con algoritmos de Súper Resolución*", ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
"Conocimiento y tecnología al servicio de México"

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



C.p. M.E. Guadalupe Garrido Rivera. Jefa del Departamento de Servicios Escolares.
Expediente.

GVGR/chg

Interior Internado Palmira S/N, Col. Palmira, C. P. 62490, Cuernavaca, Morelos.
Tel. (01) 777 3 62 77 70, ext. 4106, e-mail: dir_cenidet@tecnm.mx
www.tecna.mx | www.cenidet.edu.mx



Resumen

El ser humano cuenta con una percepción que le permite extraer lo más característico de los objetos y obtener detalles en una resolución eficaz para poder clasificarlos categóricamente, siendo un proceso que se lleva a cabo de manera compleja. Es por ello que la Visión Artificial a través de los años se ha dado a la tarea de emular y automatizar la percepción del ser humano por medio del desarrollo de sistemas computacionales.

En este trabajo de investigación se propone desarrollar un *framework* el cual permite a los usuarios aplicar diferentes técnicas de Súper Resolución (las interpolaciones del vecino más cercano, bilineal, bicúbica y *Lanczos* así como la *Wavelet Haar*) a imágenes con diversas características y diferentes formatos gráficos (*bmp*, *jpg*, *png* y *tiff*), así como la aplicación de las métricas para evaluar los resultados de dichos algoritmos (Error Cuadrático Medio *MSE*, Raíz del Error Cuadrático Medio *RMSE*, Proporción Máxima de Señal a Ruido *PSNR* e Índice de Similitud Estructural *SSIM*). También el sistema permite incorporar nuevos algoritmos de Súper Resolución de imágenes que se deseen agregar a la herramienta de experimentación, cuenta con un intérprete que facilita su uso, y su código y gramática pueden ser modificados por los miembros o integrantes del grupo de inteligencia artificial del Tecnológico Nacional de México campus CENIDET y con ello incrementar su funcionalidad.

Abstract

The human being has a perception that allows him to extract the most characteristic of the objects and to obtain details in an effective resolution to be able to classify them categorically, being a process that is carried out in a complex way. This is why Artificial Vision has been given the task of emulating and automating human perception through the development of computer systems.

In this research work we propose to develop a framework which allows users to apply different Super Resolution techniques (the Nearest Neighbor, Bilinear, Bicubic and Lanczos interpolations as well as the Wavelet Haar) to images with different characteristics and different graphic formats (bmp, jpg, png and tiff), as well as the application of the metrics to evaluate the results of these algorithms (Mean Square Error MSE, Root Mean Square Error RMSE, Peak Signal to Noise Ratio PSNR and Structural Similarity Index SSIM). The system also allows the incorporation of new Super Resolution algorithms of images that are desired to add to the experimental tool, it has an interpreter that facilitates its use, and its code and grammar can be modified by the members of the group of artificial intelligence of the Tecnológico Nacional de México campus CENIDET and with it to increase its functionality.

Dedicatoria

Quiero dedicar esta tesis...

A mi mamá por todo su amor y apoyo incondicional brindado durante toda mi vida, sabes que todos mis logros son gracias a todo tu esfuerzo, *Te amo!!*

A mi papá, sabes que *Te amo!!*

A mi hermano por estar ahí, apoyándome y motivándome siempre, *Te amo!!*

A mi novio por apoyarme y motivarme, sobre todo por tu amor gracias, *Te amo!!*

A mi cuñada por apoyarme y motivarme, *Te quiero!!*

Y a toda mi familia, en especial a mis abuelos y tíos por el apoyo brindado, ya que gracias a ellos concluyo con éxito este logro.

Agradecimientos

Agradezco al Tecnológico Nacional de México / CENIDET por haberme aceptado como alumna de esta institución y por el aprendizaje que recibí.

Agradezco al Consejo Nacional de Ciencia y Tecnología CONACYT por el apoyo económico brindado durante dos años de maestría para el desarrollo de este tema de tesis.

Agradezco a Fundación Telmex por su programa educativo brindado durante el desarrollo de la maestría.

Agradezco a mi director de tesis el Dr. Raúl Pinto Elías por su asesoría y comentarios durante la implementación de este proyecto de investigación.

Agradezco a mi comité revisor, la Dra. Andrea Magadán Salazar y el Dr. José Ruiz Ascencio por las observaciones, recomendaciones y comentarios brindados ya que fueron de gran utilidad para el desarrollo y conclusión de esta tesis.

Contenido

Resumen	i
Abstract	ii
Dedicatoria	iii
Agradecimientos	iv
Lista de figuras	viii
Lista de tablas	x
Acrónimos	xi
Glosario	xii
Capítulo 1. Framework para Súper Resolución	1
1.1 Introducción.....	1
1.2 Delimitación del problema	2
1.3 Objetivos	2
1.3.1 Objetivo general	2
1.3.2 Objetivos específicos	2
1.4 Alcances y limitaciones	3
1.5 Organización de la tesis	3
Capítulo 2 . Análisis del problema y propuesta de solución	5
2.1 Introducción	5
2.2 Antecedentes del proyecto	5
2.3 Marco conceptual y estado del arte	6
2.4 Visión Artificial	6
2.4 Framework	7
2.5 Súper Resolución.....	8
2.6 Algoritmos de Súper Resolución.....	9
2.6.1 Interpolación del vecino más cercano	9
2.6.2 Interpolación bilineal.....	10
2.6.3 Interpolación bicúbica	11
2.6.4 Interpolación <i>Lanczos</i>	12
2.6.5 <i>Wavelet</i>	12
2.7 Intérprete	13
2.7.1 Flex	14
2.7.2 Bison.....	14

2.8 Métricas de evaluación de la calidad de la imagen	15
2.8.1 MSE	15
2.8.2 RMSE	15
2.8.3 PSNR	16
2.8.4 SSIM	16
2.9 Planteamiento de la solución	16
2.10 Frameworks existentes	17
2.11 Conclusión	18
Capítulo 3 . Análisis, diseño e implementación del sistema	19
3.1 Introducción	19
3.2 Análisis de requerimientos	19
3.3 Software	20
3.3.1 Eclipse	20
3.3.2 C/C++	21
3.3.3 OpenCV	21
3.3.4 Flex	21
3.3.5 Bison	21
3.3.6 Qt Creator	21
3.4 Diseño de la interfaz	22
3.5 Desarrollo de los algoritmos clásicos de Súper Resolución	24
3.6 Diseño e implementación del intérprete	26
3.7 Implementación de las métricas de evaluación	30
3.8 Desarrollo de técnicas de preprocesamiento	31
3.9 Conclusión	34
Capítulo 4 . Experimentación y resultados	35
4.1 Introducción	35
4.2 Banco de imágenes	35
4.3 Implementación de pruebas	35
4.3.1 Prueba 01	35
4.3.2 Prueba 02	36
4.3.3 Prueba 03	37
4.3.4 Prueba 04	41
4.3.5 Prueba 05	45
4.3.6 Prueba 06	47

4.3.7 Prueba 07	48
4.3.8 Prueba 08	52
4.3.9 Prueba 09	53
4.3.10 Prueba 10	54
4.4 Compromisos alcanzados y otros logros	57
Capítulo 5 . Conclusiones	58
5.1 Conclusiones	58
5.2 Trabajos futuros.....	59
Referencias	60
Anexos.....	65
Anexo A. Intérprete	65
A.1 Expresiones Regulares	65
A.2 Reglas léxicas	65
A.3 Símbolos terminales	66
A.4 Reglas gramaticales.....	66
Anexo B. Banco de imágenes.....	67
B.1 Set5.....	67
B.2 Set14.....	67
B.3 BSD100	68
B.4 Urban100.....	69
B.5 Sun-Hays80	70

Lista de figuras

Figura 1.1. a) Imagen de baja resolución. b) Imagen de alta resolución.	1
Figura 2.1. Esquema de Visión Artificial (dado por Gonzalez & Woods, 2002).	7
Figura 2.2. Estimación de píxeles para Súper Resolución (Nakamura, 2006).	8
Figura 2.3. Interpolación del vecino más cercano (Nakamura, 2006).	9
Figura 2.4. Interpolación Bilineal (Nakamura, 2006).	10
Figura 2.5. Interpolación bicúbica (Nakamura, 2006).	11
Figura 2.6. Descomposición de la transformada Wavelet.	12
Figura 2.7. Ejemplo de la descomposición de la transformada Wavelet.	13
Figura 2.8. Esquema de un intérprete (Luengo, 2002).	13
Figura 3.1. Módulos del sistema.	22
Figura 3.2. Componentes de la Interfaz.	23
Figura 3.3. Menús y submenús.	23
Figura 3.4. R-05 Algoritmos clásicos de Súper Resolución.	24
Figura 3.5. Resultados de los algoritmos de interpolación.	25
Figura 3.6. Resultados del algoritmo de la wavelet haar.	26
Figura 3.7. R-06 Métricas para evaluar la calidad de las imágenes.	30
Figura 3.8. Imágenes a evaluar mediante las métricas.	30
Figura 3.9. Transformaciones de color.	31
Figura 3.10. Ecuación del histograma.	32
Figura 3.11. Operaciones morfológicas.	32
Figura 3.12. Transformaciones geométricas.	33
Figura 4.1. Prueba 01: Elementos de la interfaz de usuario.	36
Figura 4.2. Prueba 02: Menús y submenús de la interfaz de usuario.	37
Figura 4.3. Prueba 03: Abrir imágenes y script.	38
Figura 4.4. Prueba 03: Mostrar imágenes y script.	39
Figura 4.5. Prueba 03: Guardar script.	39
Figura 4.6. Prueba 03: Guardar imagen.	40
Figura 4.7. Prueba 03: Guardar imagen y script.	40
Figura 4.8. Prueba 03: Copiar y pegar una imagen.	41
Figura 4.9. Prueba 04: Interpolación del vecino más cercano 4X.	42

Figura 4.10. Prueba 04: Interpolación bilineal 4X.....	42
Figura 4.11. Prueba 04: Interpolación bicúbica 4X.	43
Figura 4.12. Prueba 04: Interpolación Lanczos 4X.	43
Figura 4.13. Prueba 04: Descomposición Wavelet haar 4X.	44
Figura 4.14. Prueba 04: Wavelet haar 4X.....	44
Figura 4.15. Prueba 05: Métricas de evaluación.....	45
Figura 4.16. Prueba 06: Nuevas técnicas de Súper Resolución.	47
Figura 4.17. Resta de imágenes.	48
Figura 4.18. Prueba 07: Transformaciones de color.	49
Figura 4.19. Prueba 07: Ecuilización del histograma.....	50
Figura 4.20. Prueba 07: Operaciones morfológicas.....	50
Figura 4.21. Prueba 07: Transformaciones geométricas.....	51
Figura 4.22. Prueba 07: Menú de ayuda, manual del uso del sistema.	51
Figura 4.23. Prueba 08: Ventana de historial.....	52
Figura 4.24. Prueba 09: Funcionalidad del intérprete.....	54
Figura 4.25. Prueba 10: Mosaico.	55
Figura 4.26. Prueba 10: Cascada.....	56
Figura 4.27. Prueba 10: Lista de subventanas.....	56
Figura B.1. Banco de imágenes Set5.	67
Figura B.2. Banco de imágenes Set14.	67
Figura B.3. Muestra del banco de imágenes BSD100.	68
Figura B.4. Muestra del banco de imágenes Urban100.	69
Figura B.5. Banco de imágenes Sun-Hays80.....	70

Lista de tablas

Tabla 3.1. Requerimientos del sistema.	19
Tabla 3.2. Gramática para uso del intérprete en el framework.	27
Tabla 3.3. Resultados de las métricas para evaluar las imágenes resultantes de la Figura 3.8.	31
Tabla 4.1. Resultados de aplicar las métricas a las imágenes mostradas en la prueba 04.	46
Tabla 4.2. Requerimientos y pruebas.	57
Tabla A.0.1. Expresiones regulares.	65
Tabla A.0.2. Reglas léxicas.	65
Tabla A.0.3. Reglas gramaticales.	66

Acrónimos

Listado alfabético de los acrónimos utilizados en este documento.

- PSI** Por sus siglas en inglés.
- Flex** Analizador léxico. PSI: *Fast Lexical Analyzer*.
- HR** Alta resolución. PSI: *High Resolution*.
- HSL** Matiz, Saturación, Luminosidad. PSI: *Hue Saturation Lightness*.
- HSV** Matiz, Saturación, Valor o Brillo. PSI: *Hue, Saturation, Value*.
- LR** Baja resolución. PSI: *Low Resolution*.
- MSE** Error cuadrático medio. PSI: *Mean Square Error*.
- PSNR** Proporción Máxima de Señal a Ruido. PSI: *Peak Signal to Noise Ratio*.
- RMSE** Raíz del Error Cuadrático Medio. PSI: *Root Mean Squared Error*.
- SR** Súper resolución. PSI: *Super Resolution*.
- SSIM** Índice de Similitud Estructural. PSI: *Structural Similarity Index*.
- Yacc** Otro generador de compiladores más. PSI: *Yet Another Compiler-Compiler*.

Glosario

Listado alfabético de términos técnicos utilizados en este documento.

- Framework** Conjunto de clases cooperativas que construyen un diseño reutilizable para un tipo específico de software. Proporciona la arquitectura partiendo el diseño en clases abstractas y definiendo sus responsabilidades y colaboraciones (Galindo & Camps, 2008).
- Interpolación** El proceso de interpolación es la obtención de nuevos puntos (píxeles) para generar una imagen de alta resolución, los cuales se obtienen a partir de la información contenida en la imagen de baja resolución (Gonzalez & Woods, 2002). “La interpolación es la obtención de nuevos valores dentro del intervalo de valores discretos conocidos”.
- Píxel** La palabra píxel es una combinación de *pictures* (pics o pix) y *element* (*picture element*, elemento de la imagen). Es la superficie homogénea más diminuta que forma parte de una imagen digital. Una imagen digital se compone de un número finito de elementos, cada uno con un lugar y valor específicos. Estos elementos son llamados píxeles (Mejía, 2005).
- Resolución** Es la cantidad de detalles que puede observarse en una imagen. El concepto de resolución en una imagen se refiere a la densidad de píxeles que contiene. Tener mayor resolución se traduce en obtener una imagen con más detalle o calidad visual (Pérez, 2018).
- Súper Resolución** Es el proceso por el cual se obtiene una imagen digital de alta resolución espacial a partir de una o varias imágenes digitales de baja resolución de una determinada escena (Park et al., 2003).

Capítulo 1. Framework para Súper Resolución

1.1 Introducción

Hoy en día las imágenes son utilizadas en diversos dispositivos, tales como computadoras, televisores, teléfonos celulares, entre muchos otros. En ciertos casos, la baja resolución de las imágenes impide la correcta interpretación visual de la información, para compensar esta deficiencia, existen técnicas de Súper Resolución (SR). La SR es un proceso que permite obtener más información (más píxeles) en un área de interés, por lo que un método de SR hace el proceso de estimar información desde la imagen de Baja Resolución (LR), y agregarla en una imagen de Alta Resolución (HR), como se puede observar en la Figura 1.1 (Álvarez et al., 2017).

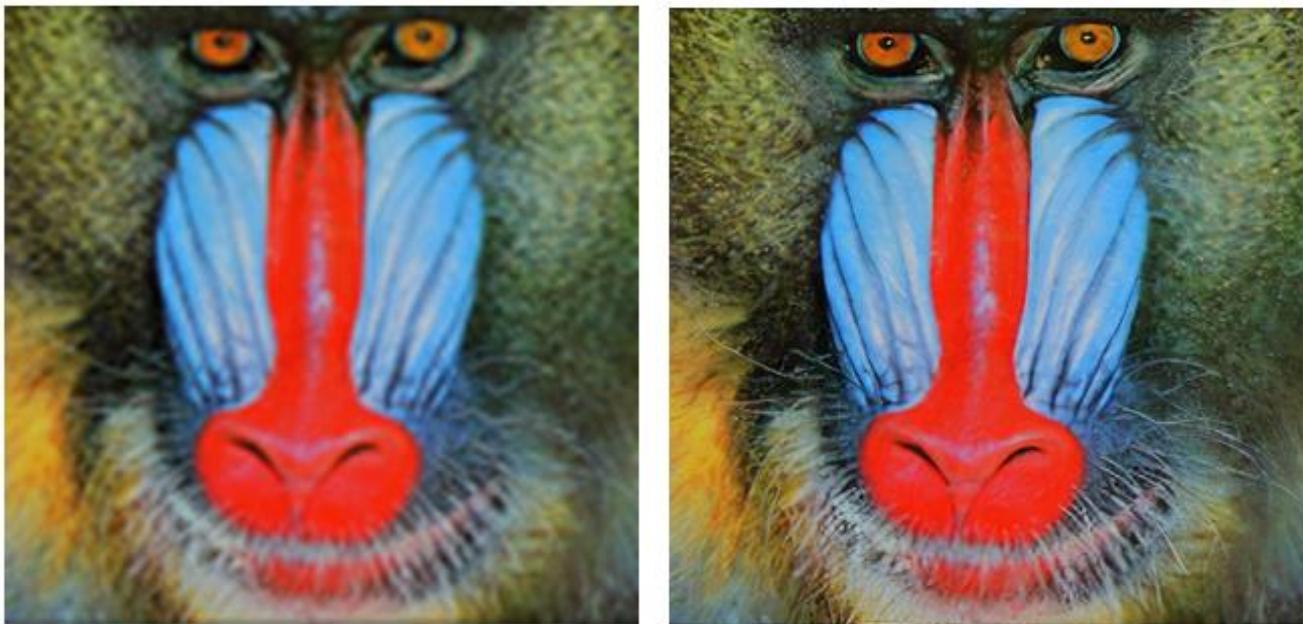


Figura 1.1. a) Imagen de baja resolución. b) Imagen de alta resolución.

La SR resulta de utilidad cuando no ha sido posible tomar imágenes de una calidad adecuada debido a problemas de coste, limitaciones físicas del sistema u otras causas. La SR es uno de los temas de tratamiento digital de imágenes que se encuentra abierto hoy en día a pesar de los numerosos trabajos desarrollados (Sánchez, 2008), (Morera et al., 2018). Por tal motivo, es conveniente contar con un *framework* que permita experimentar y comparar resultados entre algoritmos de SR.

1.2 Delimitación del problema

El Centro Nacional de Investigación y Desarrollo Tecnológico requiere un *framework* el cual permita a los usuarios aplicar diferentes técnicas de Súper Resolución a imágenes con diversas características y que facilite el uso mediante un intérprete, así como la aplicación de las métricas para evaluar los resultados de dichos algoritmos; poder incorporar nuevos algoritmos de Súper Resolución de imágenes que se deseen agregar a la herramienta de experimentación, que sea gratuito, y que su código y gramática puedan ser modificados por los miembros o integrantes del grupo de inteligencia artificial del instituto y con ello incrementar su funcionalidad.

1.3 Objetivos

A continuación, se describe el objetivo general, así como los objetivos específicos planteados para el desarrollo de este tema de tesis.

1.3.1 Objetivo general

Diseñar e implementar un entorno de experimentación (*Framework*) para la manipulación de imágenes con el enfoque de Súper Resolución.

1.3.2 Objetivos específicos

- Estudiar e implementar algoritmos clásicos de Súper Resolución.
- Estudiar e implementar rutinas de lectura y escritura de archivos con formatos gráficos específicos.
- Emplear bibliotecas y entornos de desarrollo libres/*GNU* o equivalentes.
- Estudiar e implementar un intérprete utilizando una herramienta *LEX-YACC*.

1.4 Alcances y limitaciones

- Estudiar e implementar, al menos, 4 algoritmos clásicos de Super Resolución.
- Incorporar un entorno de intérprete minimalista para dar flexibilidad a la experimentación.
- Realizar el desarrollo en *C/C++* usando *Eclipse* y las bibliotecas de desarrollo *OpenCV*, *R* y similares.
- Estudiar e implementar herramientas de *LEX-YACC* para seleccionar una.
- Implementar las funciones de lectura y escritura de, al menos, los formatos gráficos: *BMP*, *JPG*, *PNG*, *TIFF*.
- Seleccionar el banco de imágenes de acuerdo con trabajos realizados anteriormente en CENIDET.

1.5 Organización de la tesis

En el capítulo 2 se lleva a cabo el análisis del problema mencionado, así como el planteamiento de la propuesta de solución, también contiene la información obtenida de las investigaciones realizadas en el estado del arte. Incluye aportaciones referentes a técnicas clásicas de Súper Resolución, métricas para evaluar la calidad de las imágenes resultantes, así como información de utilidad para la implementación de un intérprete mediante los analizadores léxico y sintáctico.

El capítulo 3 contiene el análisis de requerimientos necesarios para la implementación del *framework*, así como el diseño e implementación de la interfaz de usuario y la gramática para el intérprete, también el desarrollo de los algoritmos clásicos de SR, las métricas para la evaluación de las imágenes y técnicas de preprocesamiento, también la funcionalidad para agregar nuevas técnicas al sistema.

En el capítulo 4 se muestran los resultados obtenidos de aplicar el plan de pruebas para la validación del correcto funcionamiento del *framework*, se detalla el análisis de los resultados obtenidos, los compromisos alcanzados y otros logros. Se mencionan los bancos de imágenes descargados y utilizados para las pruebas realizadas.

El capítulo 5 se detallan las conclusiones finales de esta tesis y se plantean los trabajos futuros relacionados al tema desarrollado.

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García.

También se presenta una sección de referencias para una consulta a mayor detalle de la información utilizada para el desarrollo de esta tesis.

Finalmente se incluyen dos anexos, el Anexo A muestra los elementos para la implementación del intérprete y el Anexo B el cual ejemplifica los bancos de imágenes descargados y utilizados para las pruebas realizadas.

Capítulo 2 . Análisis del problema y propuesta de solución

2.1 Introducción

Este capítulo contiene información recabada de las investigaciones realizadas en la literatura y el estado del arte, también incluye aportaciones referentes a los algoritmos clásicos de SR, así como las métricas para evaluar la calidad de las imágenes resultantes, e información de utilidad para la implementación del intérprete a través de los analizadores léxico y sintáctico. También incluye el planteamiento de la propuesta de solución.

2.2 Antecedentes del proyecto

El Tecnológico Nacional de México campus CENIDET cuenta con tres trabajos de investigación, los cuales se relacionan con el tema de tesis desarrollado:

1. Tesis de maestría, concluida. “Implementación de algoritmos de Súper Resolución, en hardware dedicado, para imágenes de resonancia magnética”; Pérez (2018) desarrolló un sistema de procesamiento de imágenes, que mejora la resolución de imágenes de resonancia magnética mediante el uso de métodos clásicos de súper resolución, implementado en un sistema embebido de alto rendimiento.

Realizó la evaluación de tres algoritmos clásicos de Súper Resolución (vecino más cercano, interpolación bilineal e interpolación bicúbica). Se llevó a cabo la implementación del método de Súper Resolución por interpolación bilineal en un sistema embebido de alto rendimiento por GPU, las pruebas las implementó en *cientos de imágenes* de fuentes públicas y privadas, reportó mejores resultados en cuanto a la calidad de la imagen, que los métodos del vecino más cercano e interpolación bicúbica, y obtuvo un tiempo menor a la *quinta parte* en contraste con las implementaciones en CPU.

2. Tesis de maestría, en proceso. “Desarrollo e implementación de algoritmos de Súper Resolución empleando Wavelets”; Sánchez (2018) plantea implementar y evaluar técnicas de Súper Resolución que empleen *wavelets*. Para lo cual, se estudian e implementan, *al menos, tres* técnicas de Súper Resolución, también se debe comprender e implementar, *al menos, tres* transformadas *wavelets*, así como llevar a cabo la implementación de una metodología de Súper Resolución empleando *wavelets*, para después comparar la calidad visual de las imágenes procesadas con la metodología propuesta.
3. Tesis doctoral, en proceso. “Método de Súper Resolución Híbrido para la Representación de Imágenes de Resonancia Magnética” Morera (2018) pretende mejorar los parámetros de similitud estructural en imágenes de resonancia magnética, mediante la consideración de altas escalas de amplificación y dirección de bandas de frecuencia.

Los trabajos mencionados anteriormente permitieron analizar y comprender la introducción al tema de Súper Resolución de imágenes, explicando algoritmos clásicos de SR (vecino más cercano, interpolación bilineal, interpolación bicúbica, transformadas *wavelets*) que se aplican para la obtención de imágenes con una mayor resolución.

2.3 Marco conceptual y estado del arte

A continuación se muestra la recopilación de información de acuerdo a las investigaciones realizadas y el análisis en la literatura y el estado del arte, referente a temas de súper resolución (Dong et al., 2016), (Zhang & Sze, 2017), (Yuan et al., 2018), (Morera et al., 2018), (Wang et al., 2018), métricas para evaluar la calidad de las imágenes (Blanes & Gorricho, 2015); así como información necesaria para la implementación del intérprete mediante las herramientas Flex y Bison (Paxson, 1995), (Aaby, 2003), (Béjar, 2005), (Levine, 2009) y (Gómez & Salamanca, 2012).

2.4 Visión Artificial

En (Platero, 2009) se establece que la visión artificial o también llamada visión por computador, pretende capturar la información visual del entorno físico para extraer características relevantes visuales, utilizando procedimientos automáticos. Es una representación visual del mundo. Infiere las características o propiedades del mundo a partir de una o más imágenes. Aunque cada aplicación tiene sus especificidades, existe un tronco común de etapas, como se visualiza en la Figura 2.1.

La información de bajo nivel está dada por las etapas de adquisición y procesado, las de medio nivel son las de segmentación y extracción de las características y las de alto nivel con las etapas de reconocimiento e interpretación.

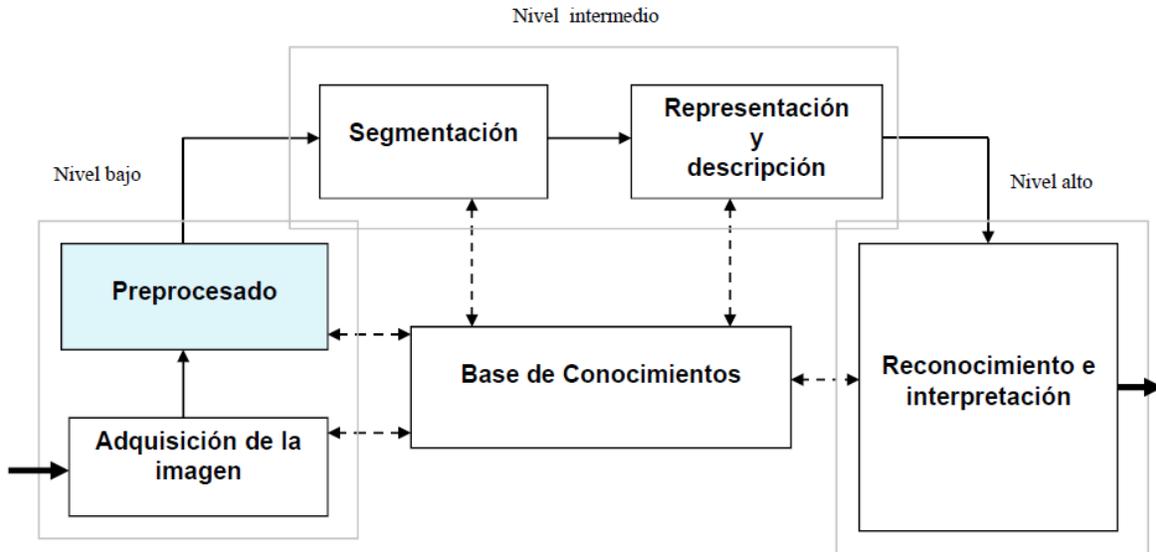


Figura 2.1. Esquema de Visión Artificial (dado por Gonzalez & Woods, 2002).

La primera etapa es la construcción del sistema de formación de las imágenes. Su objetivo es realzar, mediante técnicas fotográficas (iluminación, óptica, cámaras, filtros, pantallas), las características visuales de los objetos (formas, texturas, colores, sombras). El éxito de muchas aplicaciones depende de un buen diseño en esta primera etapa.

Una vez adquirida la imagen se pasará a la etapa de preprocesado. El objetivo es mejorar la calidad informativa de la imagen adquirida. Se incluyen operaciones de mejora de la relación señal-ruido (*denoising*), SNR, de atenuar las imperfecciones de la adquisición debido a la función de transferencia del sistema de captación de imágenes (*deconvolution*), de regularizar la imagen, de mejorar el contraste o de optimizar la distribución de la intensidad (*enhancement*) o de realzar algunas características de la imagen, como bordes o regiones. Es en esta etapa en la que se lleva a cabo el proceso de implementación de las técnicas de Súper Resolución.

2.4 Framework

Un *framework* es una implementación compuesta de un conjunto de componentes y librerías empleados para desarrollar una estructura estándar de una aplicación, que, sumado a una metodología de uso y su correspondiente documentación permite diseñar, analizar, implementar y desplegar aplicaciones de

forma estándar, de alta calidad y rapidez. También se pueden definir como soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución) (Rodríguez, 2012).

Un *framework*, es una abstracción en la que cierto código común provee una funcionalidad genérica que puede ser sobrescrita o especializada de forma selectiva por medio de código con funcionalidad específica provisto por los clientes del *framework* (Gutiérrez, 2010).

2.5 Súper Resolución

El proceso de SR es aquel en el cual el número de píxeles de una imagen se incrementa a fin de aumentar su resolución espacial, mediante la estimación del valor de los píxeles desconocidos, a partir del valor de los píxeles conocidos, contenidos en la imagen bajo análisis, como se observa en la Figura 2.2. Este proceso se lleva a cabo mediante la interpolación de la imagen bajo análisis la cual se conoce como la imagen de baja resolución (Castro et al., 2017).

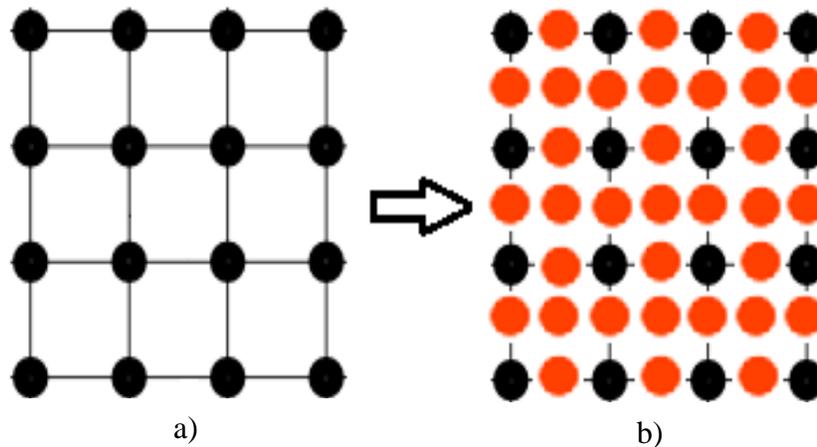


Figura 2.2. Estimación de píxeles para Súper Resolución (Nakamura, 2006).

- a) Imagen en baja resolución, los puntos negros son los píxeles originales.
- b) Imagen en alta resolución, los puntos negros son los píxeles originales y los puntos rojos son los píxeles estimados.

Existen diferentes esquemas de interpolación que han sido propuestos, en general, aunque, los algoritmos clásicos tienen baja complejidad computacional, estos no proporcionan la calidad requerida lo que ha provocado que sean poco usados para la generación de imágenes de SR en diversas aplicaciones prácticas. Sin embargo, los algoritmos clásicos han sido usados frecuentemente en las etapas de preprocesamiento de algoritmos de SR que proporcionan imágenes con mejor calidad.

Durante los últimos años han sido propuestos diversos algoritmos empleando esquemas, en el dominio espacial, en el dominio de la frecuencia o en el dominio *wavelet*. Entre los algoritmos que emplean la transformada *wavelet* discreta (*DWT*) o la transformada *wavelet* estacionaria (*SWT*) han recibido mucha atención durante los últimos años, ya que éstos proporcionan imágenes de SR con buena calidad y con un costo computacional razonable para aplicaciones en tiempo real. Sin embargo, en algunas aplicaciones se requiere mejorar aún más la calidad de las imágenes de SR sin incrementar significativamente la complejidad computacional del algoritmo (Castro et al., 2017).

2.6 Algoritmos de Súper Resolución

2.6.1 Interpolación del vecino más cercano

Asigna a cada píxel interpolado el valor del píxel más cercano. La Figura 2.3 muestra la interpolación del vecino más cercano en el espacio bidimensional (el resultado de interpolación, P_i , tiene el mismo valor que el píxel de la imagen original, P_1).

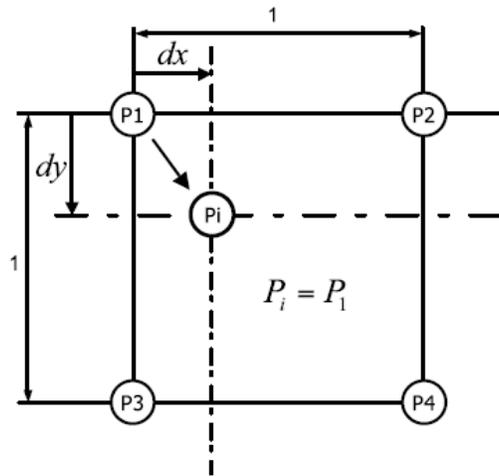


Figura 2.3. Interpolación del vecino más cercano (Nakamura, 2006).

La distancia Euclidiana (Ecuación 2.1) es utilizada para determinar qué píxel es el más cercano:

$$d_E(P1, P2) = \sqrt{(x_2, x_1)^2 + (y_2, y_1)^2} \quad (2.1)$$

Donde:

$P1$ y $P2$ son los píxeles de los cuales se obtendrá la distancia euclidiana,

(x_1, y_1) y (x_2, y_2) son las coordenadas del píxel $P1$ y del píxel $P2$, respectivamente.

La función de interpolación del vecino más cercano se expresa en la Ecuación 2.2:

$$\begin{aligned} f(x) &= 1 & 0 \leq x < 0.5 \\ f(x) &= 0 & 0.5 \leq x \end{aligned} \quad (2.2)$$

Donde:

x representan la distancia a la que se encuentra el píxel que se va a interpolar del píxel original.

Es el método más simple y requiere poco recurso computacional, el inconveniente es la mala calidad de la imagen interpolada (Nakamura, 2006).

2.6.2 Interpolación bilineal

La interpolación lineal utiliza dos píxeles adyacentes para obtener el valor de píxel interpolado. Cuando se aplica una interpolación lineal a una imagen, se llama interpolación bilineal, en la cual se utilizan cuatro píxeles circundantes para estimar un píxel.

La Figura 2.4 muestra la interpolación bilineal (se toman en cuenta los cuatro píxeles (P_1 , P_2 , P_3 y P_4) que rodean al píxel a interpolar (P_{ij}), se realiza una interpolación lineal entre los píxeles superiores (P_1 y P_2) y otra interpolación entre los dos píxeles inferiores (P_3 y P_4), por último, se hace una interpolación entre los dos valores interpolados previamente (P_a y P_b) para obtener el valor del píxel interpolado (Souverville, 2016)).

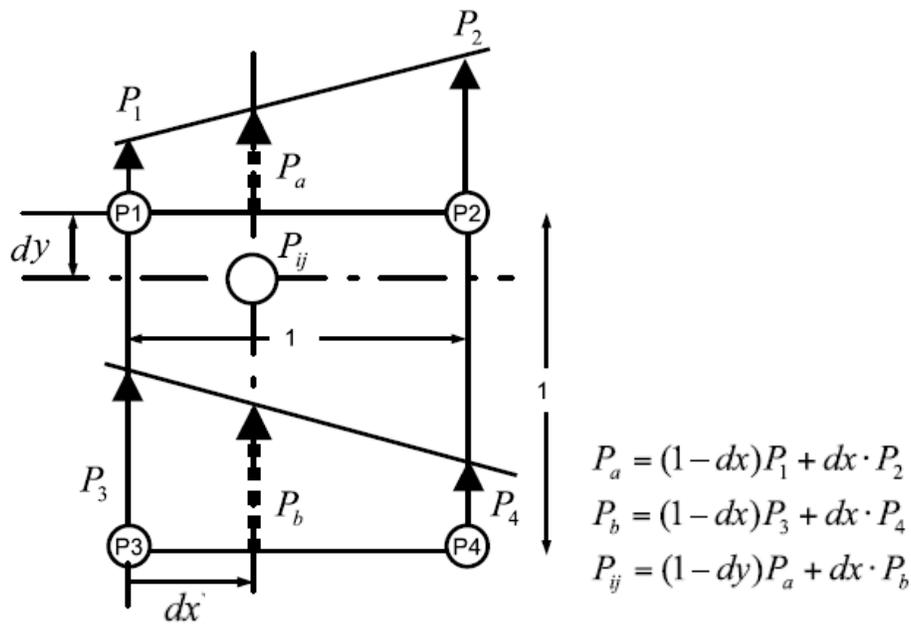


Figura 2.4. Interpolación Bilineal (Nakamura, 2006).

La función de interpolación bilineal se expresa en la Ecuación 2.3:

$$\begin{aligned} f(x) &= 1 - x & 0 \leq x < 1 \\ f(x) &= 0 & 1 \leq x \end{aligned} \quad (2.3)$$

La interpolación bilineal es un método relativamente simple, y la calidad de imagen resultante es buena (Nakamura, 2006).

2.6.3 Interpolación bicúbica

Usa 16 píxeles circundantes para estimar el valor del píxel a interpolar, como se observa en la Figura 2.5.

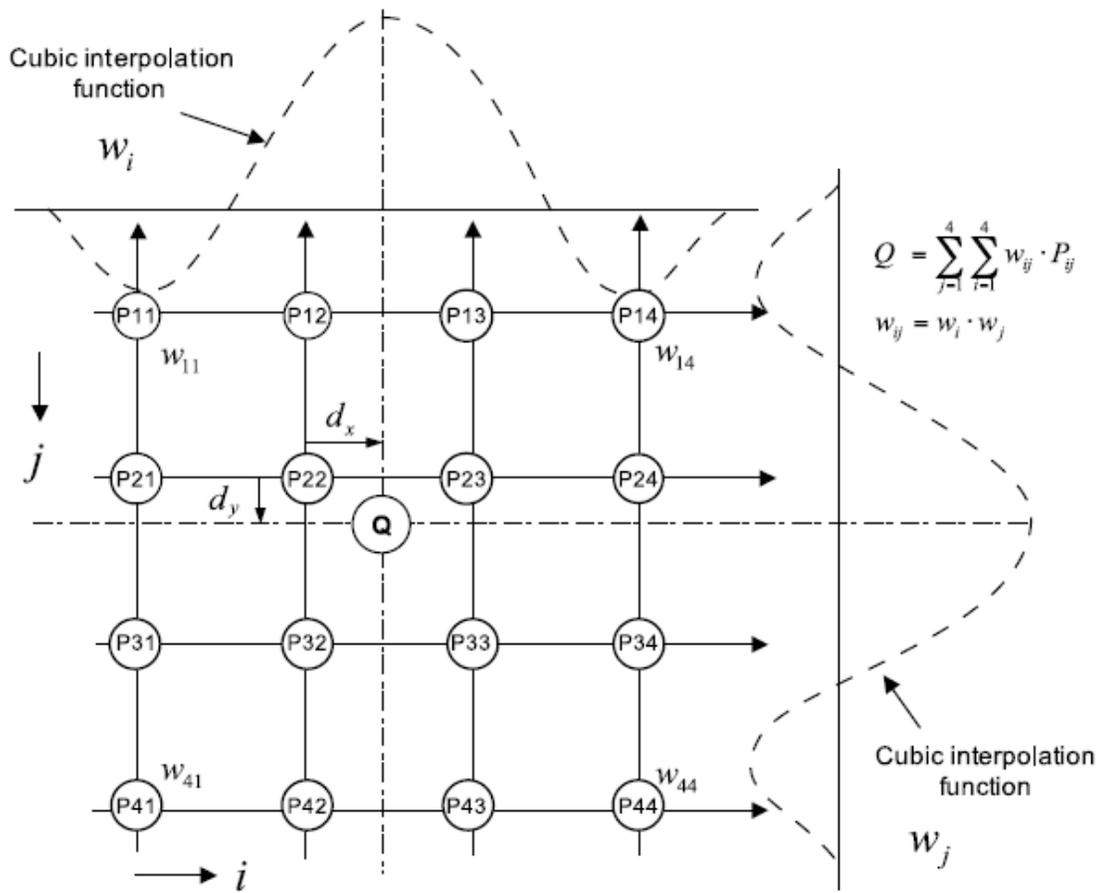


Figura 2.5. Interpolación bicúbica (Nakamura, 2006).

La función de interpolación bicúbica se expresa en la Ecuación 2.4:

$$\begin{aligned} f(x) &= (1 - x)(1 + x - x^2) & 0 \leq x < 1 \\ f(x) &= (1 - x)(2 - x)^2 & 1 \leq x < 2 \\ f(x) &= 0 & 2 \leq x \end{aligned} \quad (2.4)$$

La función de interpolación bicúbica proporciona una mejor calidad de imagen que los dos métodos descritos anteriormente, pero a costa de requisitos de recursos computacionales mucho más altos (Nakamura, 2006).

2.6.4 Interpolación *Lanczos*

Toma un intervalo de la función *Sinc(x)*, en otras palabras es una versión truncada de *Sinc(x)*, si es *Lanczos3* el radio del núcleo respectivo es de 3 y si es *Lanczos4* su radio es de 4. Puede obtenerse una mejor estimación de la imagen incrementando el radio del núcleo, este incremento también lleva a un mayor tiempo de procesamiento (Tinoco, 2014).

La función de interpolación bicúbica se expresa en la Ecuación 2.5:

$$\begin{aligned} f(x) &= \text{sinc}(x)\text{sinc}(x/a) \quad x \in (-a, a) \\ f(x) &= 0 \quad \text{en otro caso} \end{aligned} \quad (2.5)$$

Donde:

$$\text{sinc}(x) = \frac{\text{sinc}(x)}{x}$$

a representa el parámetro que ajusta la forma del kernel.

2.6.5 *Wavelet*

La Transformada Discreta de *Wavelets* (TDW), permite la descomposición de la imagen en diferentes escalas de coeficientes a la vez que se preserva la información. Las Figuras 2.6 y 2.7 muestran la forma de representar los resultados de la descomposición de la DWT de una imagen (Dumic et al., 2007).

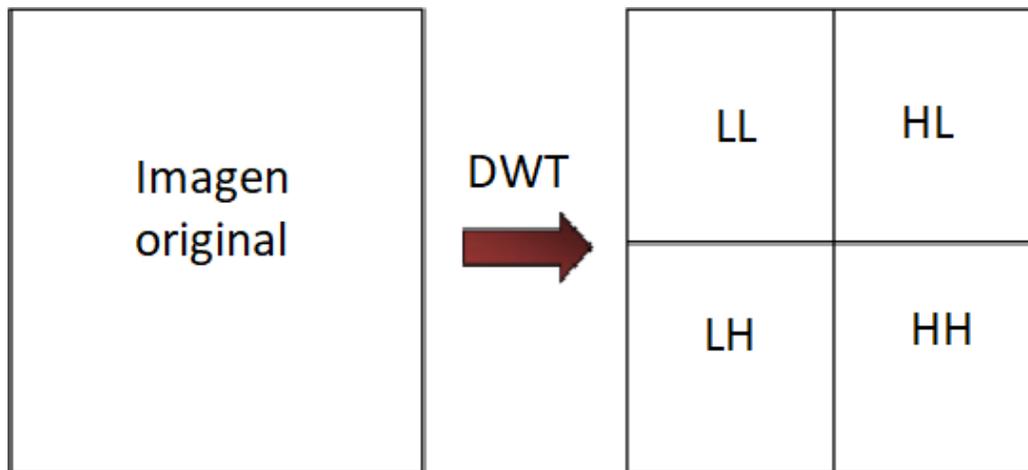


Figura 2.6. Descomposición de la transformada *Wavelet*.

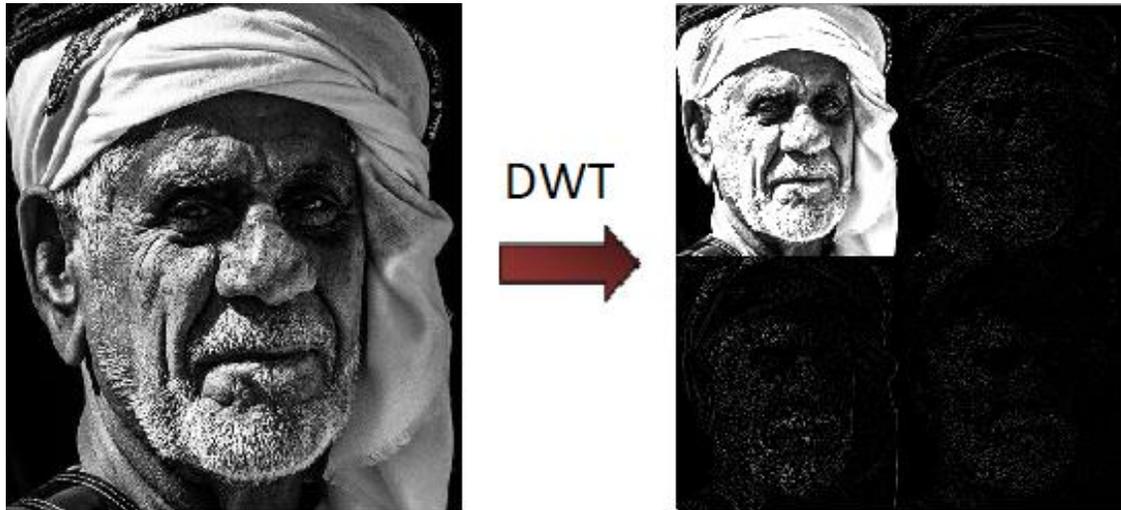


Figura 2.7. Ejemplo de la descomposición de la transformada *Wavelet*.

Como se observa, los resultados de la descomposición son la combinación de filtros, Pasa Bajo / Pasa Bajo da como resultado la aproximación de baja resolución de la imagen, mientras que las combinaciones Pasa Bajo / Pasa Alto, Pasa Alto / Pasa Bajo y Pasa Alto / Pasa Alto, proporcionando detalles horizontales, verticales y diagonales (Gao & Zheng, 2007). Para el proceso de reconstrucción simplemente se recombina las 4 sub imágenes (Pasa Bajo / Pasa Bajo, Pasa Bajo / Pasa Alto, Pasa Alto / Pasa Bajo y Pasa Alto / Pasa Alto) utilizando un proceso de undecimado (Pajares & De la Cruz, 2008).

2.7 Intérprete

Un intérprete es un programa que analiza y ejecuta de forma simultánea un programa (Luengo, 2002).

Un intérprete es un traductor que toma como entrada la especificación de un programa ejecutable y produce como salida el resultado de ejecutar la especificación. Entonces, no se tiene un resultado físico (código máquina) sino lógico (la ejecución) (Gómez & Salamanca, 2012). En la Figura 2.8, se puede visualizar un esquema general de un intérprete.

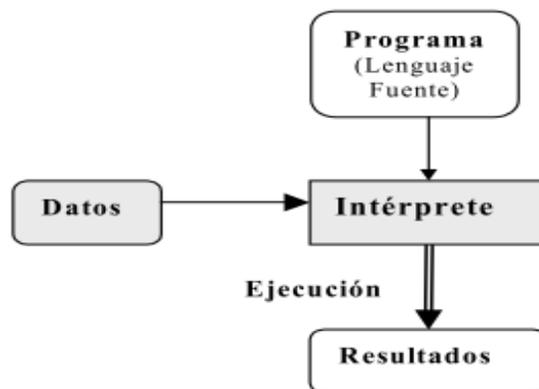


Figura 2.8. Esquema de un intérprete (Luengo, 2002).

2.7.1 Flex

De acuerdo con Paxson (1995), Aaby (2003), Béjar (2005) y Levine (2009), Flex es un una herramienta la cual permite generar analizadores léxicos, esto a partir de un conjunto de expresiones regulares. Flex busca concordancias en un fichero de entrada y ejecuta acciones asociadas a estas expresiones. Es un desarrollo realizado por GNU bajo licencia GPL. Los ficheros de entrada de Flex regularmente tienen la extensión “.l” y siguen el siguiente esquema:

```
%%  
patrón1 {acción1}  
patrón2 {acción2}  
...
```

Donde:

patrón: es una expresión regular

acción: es código en lenguaje de programación *C* con las acciones a ejecutar cuando se encuentre concordancia del patrón con el texto de entrada.

Flex recorre la entrada hasta que encuentra una concordancia y ejecuta el código asociado. El texto que no concuerda con ningún patrón lo copia tal cual a la salida.

2.7.2 Bison

De acuerdo con Aaby (2003), Béjar (2005), Levine (2009) y Donnelly & Stallman (2019), *Bison* es un generador de analizadores sintácticos que convierte una descripción para una gramática en un programa en lenguaje de programación *C*, que analiza esa gramática. Es un desarrollo realizado por *GNU* bajo licencia *GPL*. Usándolo junto a *Flex* esta herramienta permite construir compiladores de lenguajes. Un fichero de *Bison* normalmente tiene la extensión “.y” y describe una gramática. El ejecutable que se genera indica si un archivo de entrada dado pertenece o no al lenguaje generado por esa gramática. La forma general de una gramática de *Bison* es la siguiente:

```
{  
Declaraciones en C  
}  
Declaraciones de Bison  
%%  
Reglas gramaticales  
%%  
Código C adicional
```

Donde:

'%%', '{' y '}': son signos de puntuación para separar las secciones.

Declaraciones en C: pueden definir tipos y variables utilizadas en las acciones. También se puede utilizar *#include* para incluir archivos de cabecera.

Declaraciones de Bison: aquí se declaran los nombres de los símbolos terminales y no terminales.

Reglas gramaticales: son las producciones de la gramática, que además pueden llevar asociadas acciones, código en *C*, que se ejecutan cuando el analizador encuentra las reglas correspondientes.

Código C adicional: puede contener cualquier código *C* que desee utilizar.

2.8 Métricas de evaluación de la calidad de la imagen

Para validar los experimentos, es necesario llevar a cabo la evaluación de la calidad de las imágenes obtenidas, las métricas de calidad de imágenes realizan una comparación entre la imagen original y la obtenida mediante alguna técnica de SR. De acuerdo con Bhatt & Singh (2016) se muestran las siguientes métricas :

2.8.1 MSE

El Error Cuadrático Medio (*Mean Square Error*) mide el promedio de los errores al cuadrado, es decir, la diferencia entre la imagen de referencia y la obtenida.

$$MSE = \frac{\sum_{i=0}^{I-1} \sum_{j=0}^{J-1} (\hat{f}(x,y) - f(x,y))^2}{\sum_{i=0}^{I-1} \sum_{j=0}^{J-1} (f(x,y))^2} \quad (2.6)$$

Donde:

$\hat{f}(x,y)$ es la imagen original,

$f(x,y)$ es la reconstrucción,

I es el número de filas de los píxeles de las imágenes,

J es el número de columnas de los píxeles de las imágenes.

2.8.2 RMSE

Raíz del Error Cuadrático Medio (*Root Mean Squared Error*) amplifica y penaliza con mayor fuerza aquellos errores de mayor magnitud.

$$RMSE = \sqrt{MSE} \quad (2.7)$$

Donde:

MSE es el error cuadrático medio.

2.8.3 PSNR

La Proporción Máxima de Señal a Ruido (*Peak Signal to Noise Ratio*) se usa como medida cuantitativa de la calidad en la reconstrucción de imágenes. Si su valor es alto representa que el MSE entre la imagen de referencia y la obtenida es bajo, esto implica que ha sido mejorada adecuadamente.

$$PSNR = 10 \log \frac{MAX^2}{MSE} \quad (2.8)$$

Donde:

MAX es el valor máximo posible del píxel,

MSE es el error cuadrático medio.

2.8.4 SSIM

El Índice de Similitud Estructural es una medida que pretende cuantificar de forma numérica y automática la calidad visual de una imagen para un observador humano. Trabaja mediante la coincidencia de los patrones locales en la intensidad de los píxeles que han sido normalizados por luminiscencia o contraste.

$$SSIM = \frac{(2\mu_f\mu_{f'}+C_1)(2\sigma_{ff'}+C_2)}{(\mu_f^2+\mu_{f'}^2+C_1)(\sigma_f^2+\sigma_{f'}^2+C_2)} \quad (2.9)$$

Donde:

C_1 y C_2 son constantes,

σ_f y $\sigma_{f'}$ son las desviaciones estándar de las imágenes,

$\sigma_{ff'}$ es la covarianza.

2.9 Planteamiento de la solución

Diseñar e implementar un *framework*, con una interfaz de usuario ergonómica (menús, barras de herramientas e iconos de fácil acceso), que cuente con un intérprete para analizar y ejecutar los comandos proporcionados por el usuario (indicaciones en script), que permita el manejo de distintos formatos gráficos, aplicar y agregar nuevas técnicas (algoritmos) de Súper Resolución (extensiones), y que sea posible aplicar las métricas de evaluación para medir la calidad de las imágenes.

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García.

Adicional a los objetivos palteados en el Capítulo I, se integrarán al *framework* cuatro técnicas para el preprocesamiento de las imágenes, transformaciones de color, ecualización del histograma, operaciones morfológicas y transformaciones geométricas, las cuales de acuerdo con el estado del arte son empleadas antes o durante el proceso de SR.

2.10 Frameworks existentes

Superresolution Demo Esta es una herramienta de Súper Resolución basada en la proyección sobre conjuntos convexos (POCS). También puede comparar el resultado con la proyección bilineal (usando uno de los cuadros). Compatibilidad de *MATLAB*: creado con R2010b, compatible con cualquier lanzamiento. Compatibilidad de plataforma: *Windows, MacOS y Linux* (Cheng, 2019).

Super-resolution Software Es programa de *MATLAB* con interfaz gráfica de usuario que implementa varios métodos de Súper Resolución de imágenes desarrollados en el proyecto. Dicha aplicación está disponible para el uso de investigadores y empresas. Configuraciones probadas: *MATLAB* R2013b en *Windows 7 y Centos OS V.6.5* (Villena et al., 2015).

Super-resolution Toolbox Esta herramienta de *MATLAB* recopila 26 métodos de Súper Resolución de imagen única disponibles en la literatura. También contiene 6 conjuntos de datos de imágenes de teledetección simulados para probar el rendimiento de la Súper Resolución de los métodos (Fernandez et al., 2016).

Parallel Super-Resolution es un complemento *ImageJ* multiproceso para la súper resolución. La súper resolución es un problema de reconstrucción y fusión de imágenes, donde se obtiene una imagen de resolución mejorada a partir de varias imágenes de baja resolución deformadas geoméricamente. *Parallel Super-Resolution* es bajo Licencia Pública General de GNU , excepto para *Parallel Colt* (Parallel Super Resolution, 2013).

Regularized Multiframe Super-resolution Model esta función implementa el modelo de súper resolución de *multitrama* basado en la función de regularización controlada por difusión. Requiere: *Image Processing Toolbox y Signal Processing Toolbox*. Compatibilidad de *MATLAB*: creado con R2014b, compatible con cualquier lanzamiento. Compatibilidad de plataforma: *Windows, MacOS y Linux* (Maiseli, 2019).

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García.

Image Processing Toolbox proporciona un conjunto completo de algoritmos de referencia estándar y aplicaciones de flujo de trabajo para el procesamiento, análisis, visualización y desarrollo de algoritmos de imágenes. Puede realizar segmentación de imágenes, mejora de imágenes, reducción de ruido, transformaciones geométricas, registro de imágenes y procesamiento de imágenes en 3D (*Image Processing Toolbox*, 2019).

MATLAB combina un entorno de escritorio ajustado para análisis iterativo y procesos de diseño con un lenguaje de programación que expresa directamente las matemáticas de matriz y matriz. Incluye el *Live Editor* para crear scripts que combinan código, salida y texto formateado en un cuaderno ejecutable (*MATLAB*, 2019).

Signal Processing Toolbox proporciona funciones y aplicaciones para analizar, preprocesar y extraer características de señales muestreadas de manera uniforme y no uniforme. La caja de herramientas incluye herramientas para el diseño y análisis de filtros, remuestreo, suavizado, reducción de tendencias y estimación del espectro de potencia. También proporciona funcionalidad para extraer características como puntos de cambio y envolventes, encontrar picos y patrones de señal, cuantificar similitudes de señal y realizar mediciones como SNR y distorsión. También puede realizar análisis modal y de orden de señales de vibración (*Signal Processing Toolbox*, 2019).

2.11 Conclusión

A lo largo de este capítulo se presentó información correspondiente SR. así como las técnicas clásicas referentes a interpolación y *wavelet*, se mencionó que el proyecto se centra en la etapa de preprocesamiento dentro de las etapas que conforman el esquema de visión artificial, también se detallaron las métricas que permiten evaluar la calidad de las imágenes resultantes de acuerdo con el análisis realizado en el estado del arte, se recolectó información acerca de intérpretes y se seleccionaron las herramientas *Flex* y *Bison* para su creación. En comparación con los *frameworks* ya existentes una de las principales ventajas es el aspecto de contar con los códigos fuente que permiten ampliar la funcionalidad y la experimentación con nuevos algoritmos sin tener que depender de licencias y/o de terceros, también contar con un intérprete dentro del *framework* el cual facilita la funcionalidad del mismo, así como el trabajar con múltiples ventanas las cuales permiten visualizar los cambios realizados mediante las diversas técnicas aplicadas. En el capítulo 3 se detalla el proceso y desarrollo del *framework* así como sus componentes.

Capítulo 3 . Análisis, diseño e implementación del sistema

3.1 Introducción

En este capítulo se desarrolla el análisis de requerimientos, así como los diagramas que ejemplifican el diseño tanto los módulos y componentes requeridos para la implementación del *framework*. También se menciona el software con el que se trabajó para el cumplimiento de los objetivos, y se muestran las implementaciones realizadas referentes a los diversos componentes del *framework* como son los algoritmos clásicos de SR, las métricas para evaluar la calidad de las imágenes resultantes, técnicas para el preprocesamiento de imágenes, la funcionalidad para agregar nuevas técnicas, así como la creación del intérprete para la facilidad de uso de la herramienta creada.

3.2 Análisis de requerimientos

La Tabla 3.1 contiene la lista de los requerimientos necesarios para el eficiente funcionamiento del sistema.

Tabla 3.1. Requerimientos del sistema.

Identificador	Requerimiento	Descripción
R-01	Interfaz de usuario ergonómica.	Diseñar e implementar una interfaz de usuario ergonómica, que cuente con: menús, área para las ventanas, ventanas para el intérprete y el historial, y barras de herramientas.
R-02	Interfaz de Múltiples Documentos (<i>MDI</i>).	Diseñar e implementar la funcionalidad de Interfaz de Múltiples Documentos MDI, para poder trabajar con diferentes ventanas dentro del <i>framework</i> .
R-03	Intérprete (<i>Script</i>).	Implementar un intérprete utilizando una herramienta <i>LEX-YACC (Flex y Bison)</i> .

Continuación Tabla 3.1. Requerimientos del sistema.

Identificador	Requerimiento	Descripción
R-04	Lectura y escritura de diversos formatos gráficos.	Implementar las funciones de lectura y escritura (abrir, mostrar, guardar, copiar, pegar) de, al menos, los formatos gráficos: <i>BMP, JPG, PNG, TIFF</i> .
R-05	Algoritmos clásicos de Súper Resolución.	Implementar algoritmos clásicos de Súper Resolución: Interpolaciones: vecino más cercano, bilineal, bicúbica, <i>lanzcos</i> y <i>wavelet haar</i> .
R-06	Métricas para evaluar la calidad de las imágenes.	Implementar las métricas para evaluar la calidad de las imágenes: <i>MSE, RMSE, PSNR, SSIM</i> .
R-07	Nuevas técnicas de Súper Resolución.	Implementar la funcionalidad para agregar nuevas técnicas de Súper Resolución.
R-08	Funcionalidades adicionales (técnicas de preprocesamiento de imágenes y documentación de ayuda).	Implementar técnicas clásicas de Preprocesamiento de imágenes, por ejemplo: transformaciones de color, ecualización del histograma, operaciones morfológicas, transformaciones geométricas, etc. Desarrollar la documentación de ayuda: manual para el uso del sistema, manual para la programación del script (uso intérprete), manual para agregar nuevas técnicas, etc.

3.3 Software

A continuación se menciona y describe de forma breve el software utilizado para el desarrollo e implementación del *framework*.

3.3.1 Eclipse

Para determinar el *IDE* con el cual se trabajó, se realizó el análisis de diversas alternativas principalmente del ámbito de código abierto, descartando el uso de productos dependientes del pago de licencia, principalmente para no limitar el desarrollo del trabajo de investigación propuesto, por lo cual se seleccionó *Eclipse* como ambiente de desarrollo.

3.3.2 C/C++

Para llevar a cabo el desarrollo del sistema se utilizó el lenguaje C/C++, al considerarse las características básicas requeridas para los objetivos establecidos. Es un lenguaje potente para el procesamiento matemático y gráfico, es orientado a objetos lo cual brinda modularidad a las distintas etapas del procesamiento de imágenes y visualización de los resultados, permite la programación de bajo y alto nivel.

3.3.3 OpenCV

Para el manejo y procesamiento de imágenes se utilizó la biblioteca de código abierto para desarrollo de visión artificial *OpenCV*, ya que es un proyecto que posee un alto grado de madurez y constante actualización la cual provee una gran variedad de herramientas para procesamiento de imágenes y sus funciones son compatibles con el lenguaje de programación C/C++.

3.3.4 Flex

Para llevar a cabo el diseño e implementación del intérprete y la gramática se utilizó la herramienta Flex para la construcción del analizador léxico para procesar la salida y enviar los componentes léxicos al analizador sintáctico. *Flex* es de software libre y permite generar analizadores para usar con C++.

3.3.5 Bison

Para llevar a cabo el diseño e implementación del intérprete y la gramática se utilizó la herramienta *Bison* para implementar el analizador sintáctico y así especificar una acción una vez reconocida una instancia de las reglas gramaticales. *Bison* es una herramienta desarrollada por GNU bajo licencia GPL, usándolo junto a *Flex* esta herramienta permite construir compiladores de lenguajes.

3.3.6 Qt Creator

Para el desarrollo de la interfaz gráfica de usuario se seleccionó la herramienta *Qt Creator*, ya que es un IDE multiplataforma, en él se puede configurar fácilmente la librería *OpenCV*, es compatible con el lenguaje de programación C/C++ y es gratuito.

3.4 Diseño de la interfaz

Para llevar a cabo el desarrollo de la interfaz de usuario primero se realizó el análisis de los elementos necesarios para la creación y funcionamiento del *framework*, en la Figura 3.1 se puede observar que el sistema consta de siete módulos, cada uno con sus propios elementos.

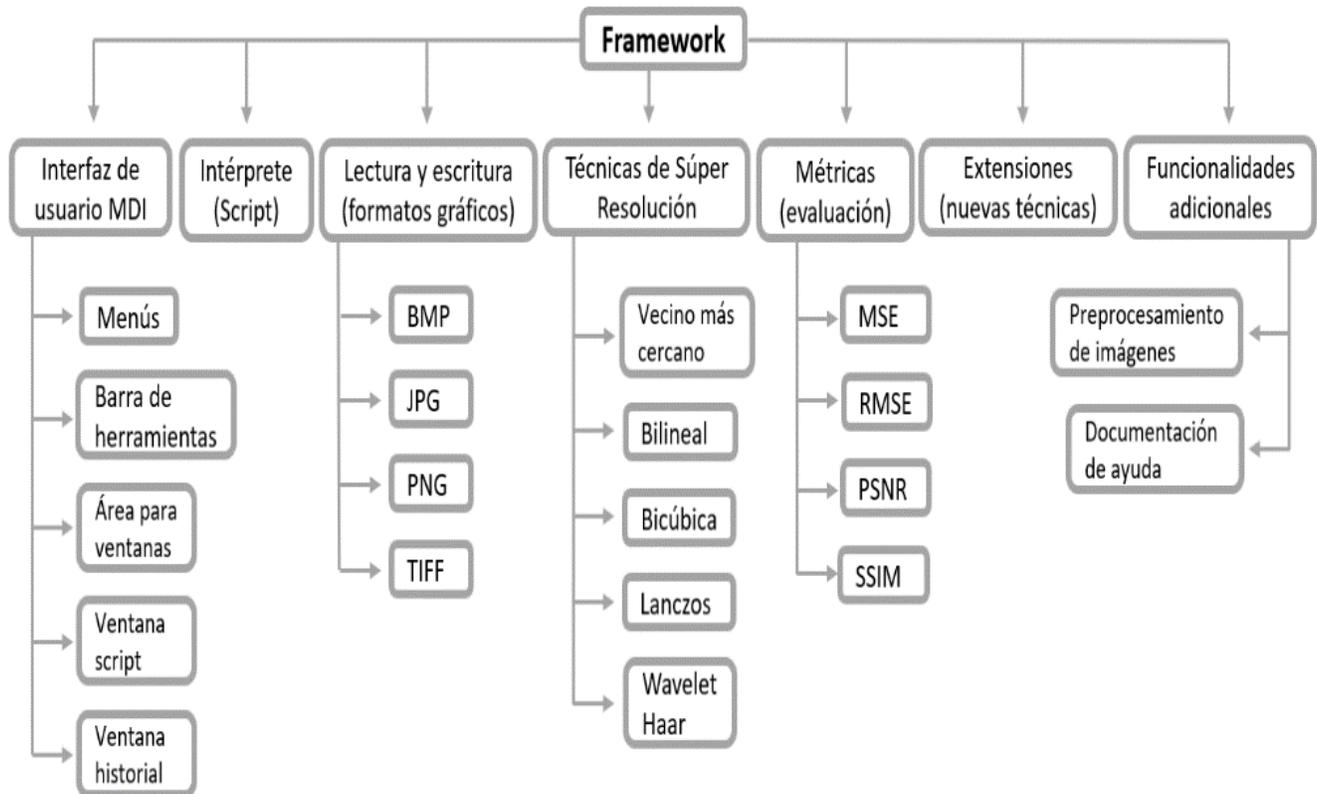


Figura 3.1. Módulos del sistema.

La Figura 3.2 muestra que la interfaz de usuario se compone de tres, el área de menús, la cual consta de diez menús, cada uno con sus submenús desplegables (la Figura 3.3), el área para ventanas, la cual permite trabajar mediante múltiples ventanas tanto de las imágenes a trabajar como de los *scripts*, también contiene una ventana específica para ir actualizando el historial de las acciones realizadas. Por último, el área para la barra de herramientas la cual consta de diez iconos de fácil acceso para las acciones elementales.

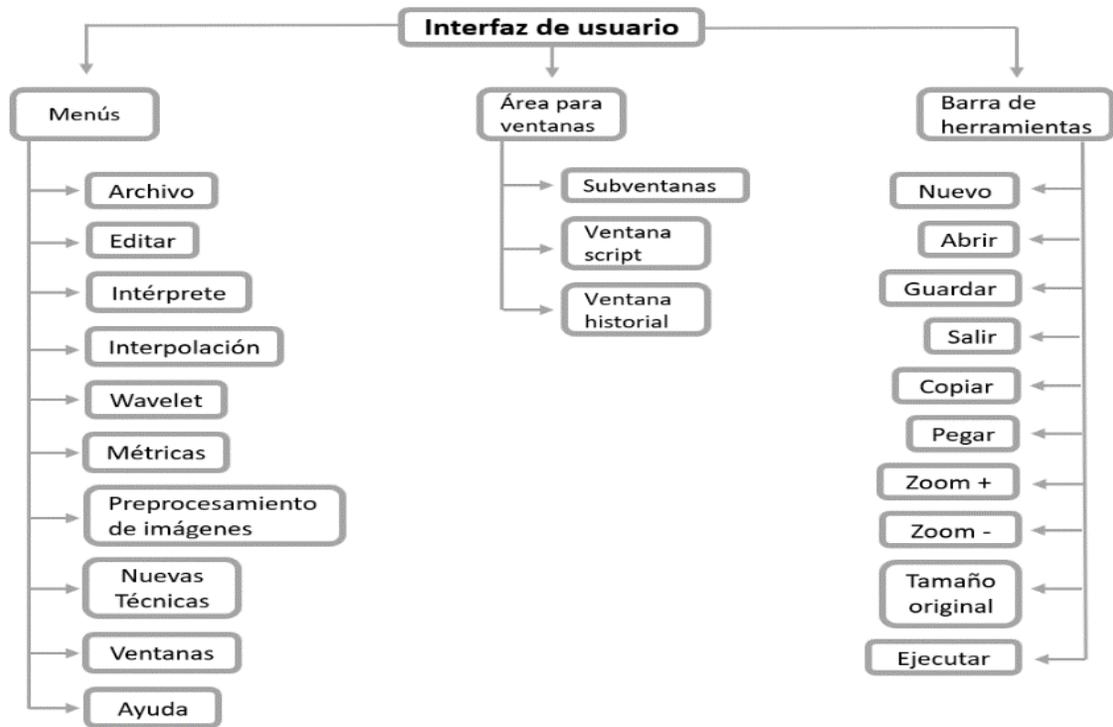


Figura 3.2. Componentes de la Interfaz.

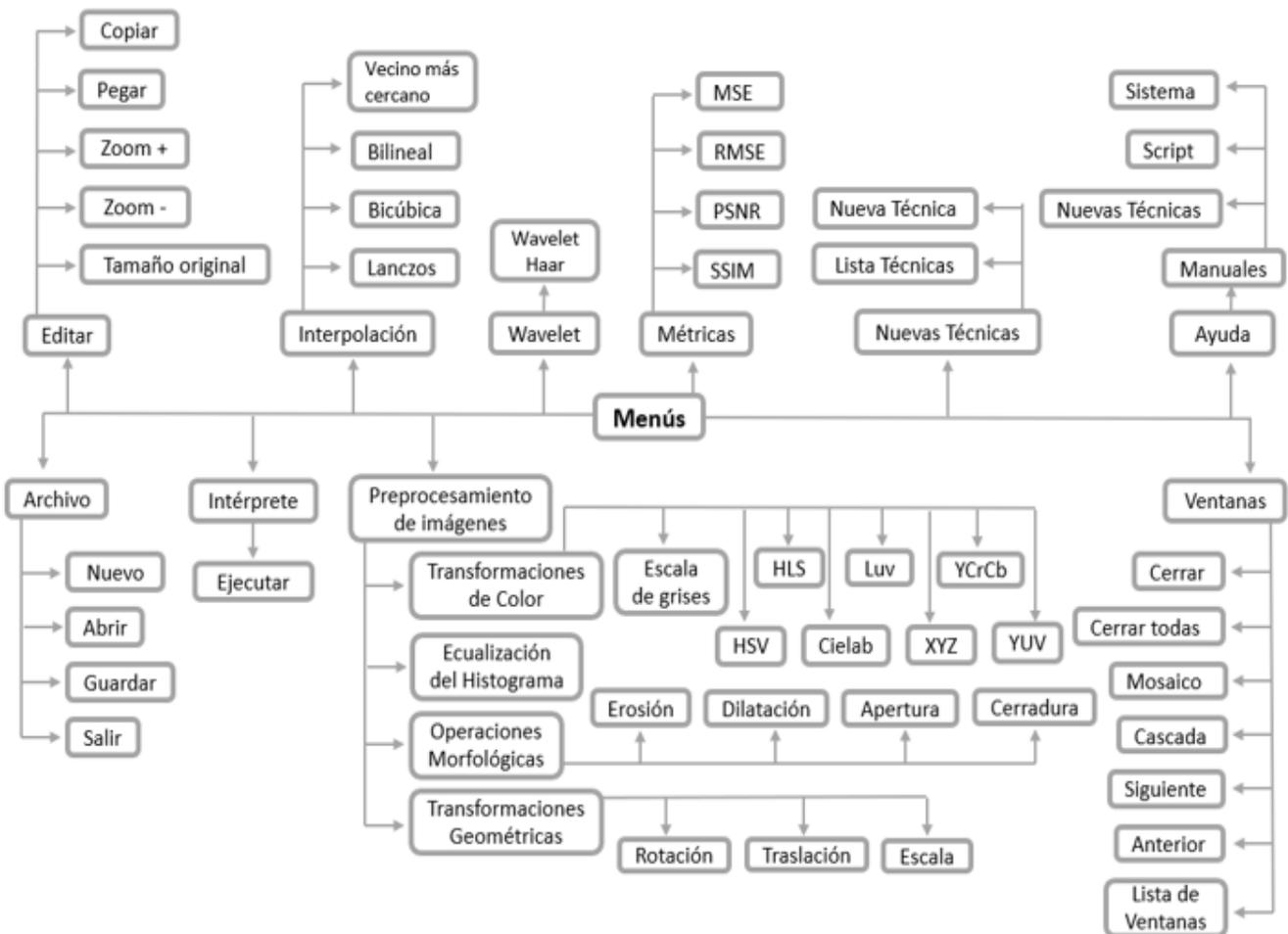


Figura 3.3. Menús y submenús.

3.5 Desarrollo de los algoritmos clásicos de Súper Resolución

La Figura 3.4 muestra los cinco algoritmos clásicos de Súper Resolución referentes al requerimiento R-05, los cuales son categorizados de acuerdo a su tipo, interpolación o *wavelet*.

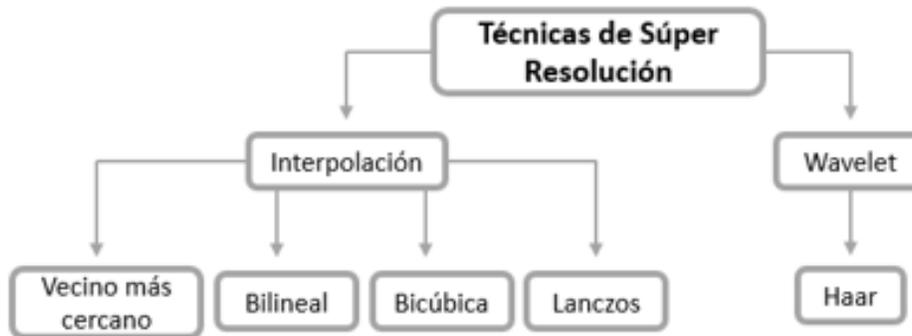


Figura 3.4. R-05 Algoritmos clásicos de Súper Resolución.

Para generar los algoritmos clásicos de Súper Resolución se implementó un programa en C++ con la función *resize* de *OpenCV*, a continuación, se muestra la sintaxis de dicha función, así como los resultados obtenidos para cada técnica de SR.

La función de *OpenCV* *resize* permite cambiar el tamaño de una imagen (*OpenCV*, 2018).

Sintaxis:

C++: `void resize(InputArray src, OutputArray dst, Size dsize, double fx=0, double fy=0, int interpolation=INTER_LINEAR)`

Parámetros:

src imagen de entrada.

dst imagen de salida.

dsize tamaño de imagen de salida.

fx factor de escala a lo largo del eje horizontal.

fy factor de escala a lo largo del eje vertical.

interpolation método de interpolación.

INTER_NEAREST: interpolación del vecino más cercano.

INTER_LINEAR: interpolación bilineal.

INTER_CUBIC: interpolación bicúbica.

INTER_LANCZOS4: interpolación *Lanczos*.

En la Figura 3.5 se observa la imagen de *lenna* del banco de imágenes *Set14*, de la cual se toma una región de interés (25x25) la cual se interpoló mediante las cuatro técnicas con un factor de escala igual a cuatro (100x100).

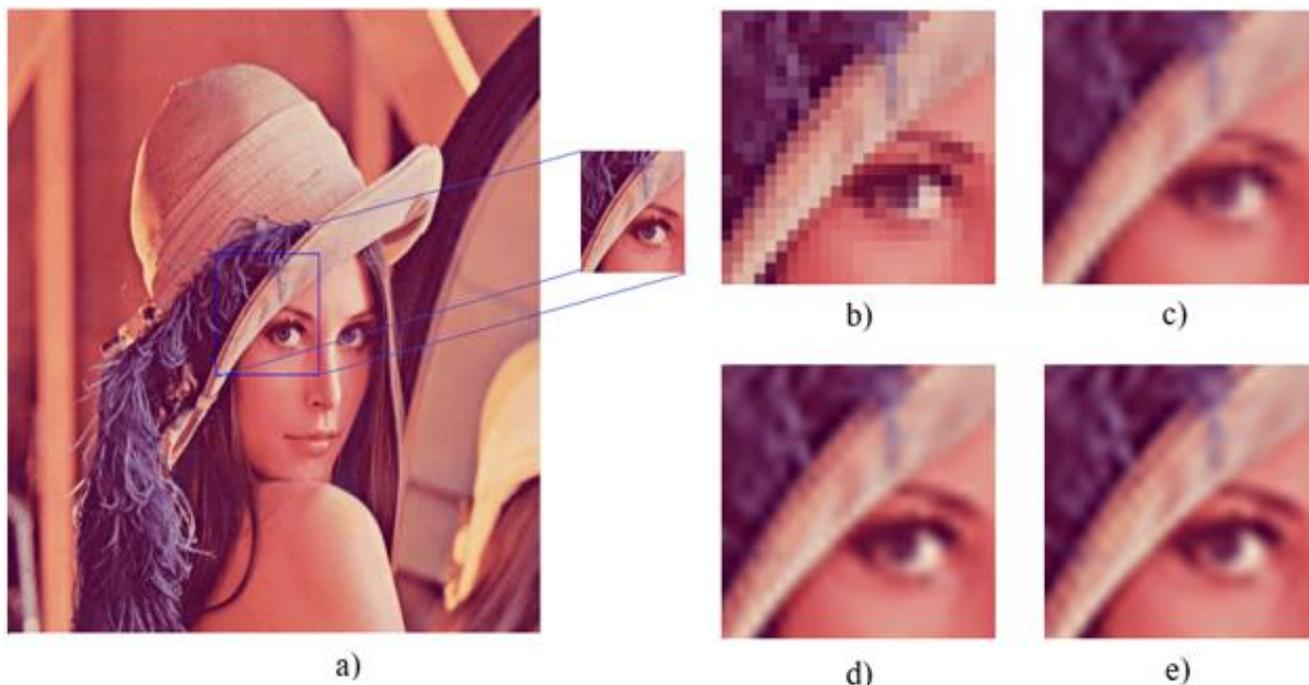


Figura 3.5. Resultados de los algoritmos de interpolación.

- a) Imagen en baja resolución y región de interés b)
Interpolación del vecino más cercano c) Interpolación
bilineal d) Interpolación bicúbica e) Interpolación *Lanczos*.

También se realizó la implementación de la transformada *Wavelet*, a través de la utilización de filtros de pasa banda de descomposición y reconstrucción, mediante los coeficientes de descomposición (0.707, 0.707) de frecuencias altas y bajas de la transformada *Wavelet* para *Haar*.

La Figura 3.6 muestra la imagen de *lenna*¹ del banco de imágenes *Set14*, a la cual se le aplicó la transformación a escala de grises, y posteriormente, se realizó la descomposición de la imagen; después se le aplicó una interpolación bicúbica con un factor de escala igual a cuatro y, por último, se reconstruyó obteniendo así una imagen en alta resolución.

¹ La imagen se llama “lenna”, así la nombraron en la revista en que apareció; la mujer que aparece en la imagen se llama “Lena Sjooblom” (nombre de soltera). Así que es correcto llamar a la imagen “lenna”.



Figura 3.6. Resultados del algoritmo de la wavelet haar.

a) Imagen original b) Descomposición *wavelet* c) SR mediante *wavelet haar*.

3.6 Diseño e implementación del intérprete

Para llevar a cabo el desarrollo de la gramática, primero se determinaron las palabras reservadas y su sintaxis, las cuales sirven como comandos en el intérprete, después, se definieron las acciones que realiza cada una, posteriormente, fueron llevadas a *Flex* para lo cual se declararon las expresiones regulares, así como las reglas léxicas. En *Bison* fueron declarados los símbolos terminales y las reglas de producción. En la Tabla 3.2 se muestran las palabras reservadas que se definieron para que el usuario pueda llevar a cabo la interacción con el *framework* mediante las indicaciones declaradas en un script (.txt) :

Tabla 3.2. Gramática para uso del intérprete en el *framework*.

Nombre	Sintaxis gramática	Acción(es)	Ejemplo comandos
Nuevo script	<i>NUEVO : Palabra_Nuevo;</i>	Crear nueva ventana para ejecutar scripts.	<i>Nuevo</i>
Abrir ventana para seleccionar archivo(s)	<i>ABRIR_ARCHIVO : Palabra_Abrir_Archivo:</i>	Abrir ventana para que el usuario seleccione la imagen o el script (Imagen: BMP, JPG, PNG, TIFF. Script: TXT).	<i>AbrirArchivo</i>
Abrir archivo ruta	<i>ABRIR : Palabra_Abrir Blanco Ruta;</i>	Validar la ruta, nombre y extensión de la imagen o script, de ser correcta abrir la imagen o script, de lo contrario mostrar error y la ruta (Imagen: BMP, JPG, PNG, TIFF. Script: TXT).	<i>Abrir C:/lenna.png</i>
Técnica de Súper Resolución	<i>SR: Tecnica Blanco TipoTecnica Blanco EscalaX Blanco NombreArchivo;</i>	Aplicar técnica de SR a una imagen (<i>Interpolación, Wavelet. NN, BL, BC, Lanczos, Haar</i>).	<i>Interpolacion Lanczos 2X lenna.png</i>
Métricas para evaluar imágenes	<i>MÉTRICAS: Palabra_Metricas Blanco NombreArchivo Blanco Ruta</i>	Mostrar tabla con los resultados obtenidos de las métricas de evaluación de la calidad entre dos imágenes (<i>Métricas</i> (MSE, RMSE, PSNR y SSIM)).	<i>Metricas lenna.png_Lanczos_2X C:/lenna_HR.png</i>

Continuación Tabla 3.2. Gramática para uso del intérprete en el *framework*.

Nombre	Sintaxis	Acción(es)	Ejemplo comandos
Copiar imagen	<i>COPIAR: Palabra_Copiar Blanco NombreArchivo;</i>	Copiar una imagen en el portapapeles.	<i>Copiar lenna.png</i>
Pegar imagen	<i>PEGAR: Palabra_Pegar;</i>	Pegar la imagen copiada en el portapapeles.	<i>Pegar</i>
Transformaciones de color	<i>COLOR : Color Blanco NombreArchivo;</i>	Aplicar una transformación de color a una imagen (Transformaciones de color: EscalaDeGrisés, HSV, HSL, Cielab, Luv, XYZ, YCrCb, YUV).	<i>HSV lenna.png</i>
Ecuación del histograma	<i>ECUALIZAR_HISTOGRAMA : Palabra_EcuaciónHistograma Blanco NombreArchivo;</i>	Aplicar la ecualización del histograma a una imagen.	<i>EcuaciónHistograma lenna.png</i>
Operaciones morfológicas	<i>OPERACION_MORFOLOGICA : Morfológica Blanco NombreArchivo Blanco ElementoEstructurante;</i>	Aplicar una operación morfológica a una imagen con cierto valor estructurante (<i>Erosión, Dilatación, Apertura, Cerradura</i>).	<i>Erosion lenna.png 5</i>
Abrir manual	<i>MANUAL: Palabra_Manual Blanco Manual;</i>	Mostrar el manual, para su lectura (<i>Sistema, Script, Técnica</i>)	<i>Manual Sistema</i>

Continuación Tabla 3.2. Gramática para uso del intérprete en el *framework*.

Nombre	Sintaxis	Acción(es)	Ejemplo comandos
Rotar imagen	<i>ROTAR : Palabra_Rotar Blanco NombreArchivo Blanco Grados;</i>	Rotar una imagen ciertos grados.	<i>Rotar lenna.png 90</i>
Trasladar imagen	<i>TRASLADAR : Palabra_Trasladar Blanco NombreArchivo Blanco Columns Coma Filas;</i>	Trasladar una imagen cierto número de columnas y filas.	<i>Trasladar lenna.png 10,10</i>
Escalar imagen	<i>ESCALAR : P_Escalar Blanco NombreArchivo Blanco Porcentaje;</i>	Escalar un cierto porcentaje una imagen.	<i>Escalar lenna.png 50</i>
Guardar archivo seleccionando ruta	<i>GUARDAR_ARCHIVO : Palabra_Guardar_Archivo Blanco NombreArchivo;</i>	Abrir ventana para que el usuario seleccione la carpeta para guardar la imagen o <i>script</i> y le coloque nombre y extensión.	<i>GuardarArchivo lenna.png</i>
Guardar archivo especificando ruta	<i>GUARDAR : Palabra_Guardar Blanco NombreArchivo Blanco Ruta</i>	Guardar imagen o <i>script</i> (nombre y ruta establecidos).	<i>Guardar lenna.png C:/lenna2.png</i>
Cerrar el sistema	<i>SALIR : Palabra_Salir</i>	Finalizar la ejecución del <i>framework</i> .	<i>Salir</i>

En el Anexo A se puede consultar con mayor detalle las expresiones regulares y las reglas léxicas que componen al analizador léxico así como los símbolos terminales y las reglas gramaticales del analizador sintáctico creado para la implementación del intérprete.

3.7 Implementación de las métricas de evaluación

La Figura 3.7 muestra las cuatro métricas para evaluar la calidad de las imágenes referentes al requerimiento R-06.



Figura 3.7. R-06 Métricas para evaluar la calidad de las imágenes.

Se llevó a cabo la implementación en *C++* y *OpenCV* de las métricas para evaluar la calidad de las imágenes resultantes en SR, a continuación se muestran los resultados obtenidos (Tabla 3.3) para las 4 interpolaciones realizadas, la Figura 3.8 muestra a *lenna* del *Set14* en baja resolución así como las imágenes resultantes de aplicar la interpolación y la imagen de referencia, *lenna* en alta resolución.



Figura 3.8. Imágenes a evaluar mediante las métricas.

- a) Imagen en baja resolución
- b) Vecino más cercano
- c) Bilineal
- d) Bicúbica
- e) *Lanczos*
- f) Imagen en alta resolución.

Tabla 3.3. Resultados de las métricas para evaluar las imágenes resultantes de la Figura 3.8.

Imagen	MSE	RMSE	PSNR	SSIM
b) Vecino más cercano	140.148	11.8384	26.6649	0.659128
c) Bilineal	110.601	10.5167	27.6932	0.681658
d) Bicúbica	92.0757	9.59561	28.4894	0.699268
e) Lanczos	88.1907	9.39099	28.6766	0.703022

De acuerdo con los resultados obtenidos, se observa que la interpolación *Lanczos* es la que mejor se ajusta a la imagen de alta resolución para este ejemplo.

3.8 Desarrollo de técnicas de preprocesamiento

Para generar las técnicas de preprocesamiento se llevó a cabo la implementación de programas en C++ y OpenCV, a continuación, se muestran los resultados obtenidos para cada técnica. En la Figura 3.9 se muestra la imagen *butterfly* del banco de imágenes *Set5* así como las imágenes resultantes que se obtienen al aplicarle las transformaciones de color, escala de grises, *HSV*, *Cielab*, *HSL*, *Luv*, *XYZ*, *YCrCb*, y *YUV*.

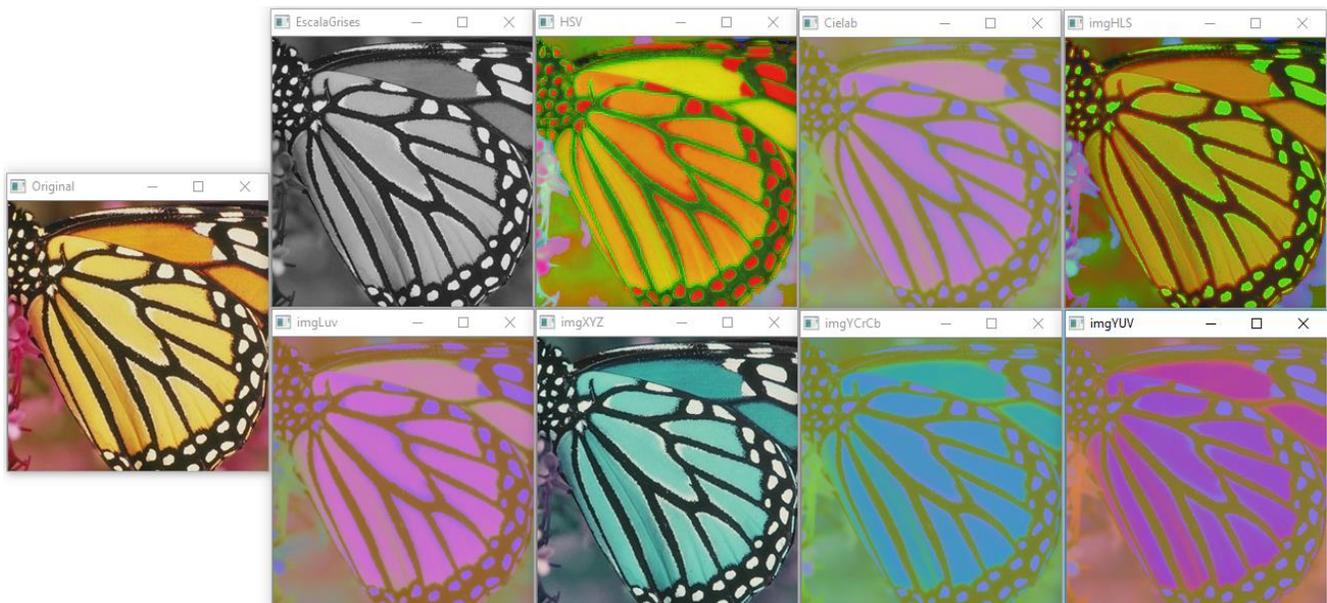


Figura 3.9. Transformaciones de color.

La Figura 3.10 muestra la imagen *102061* tomada del banco de imágenes *BSD100* a la cual se le aplicó una transformación a escala de grises, así como el resultado de aplicar la técnica de equalización del histograma, en la cual se observa una mejor distribución de los valores dentro del rango de colores que contiene la imagen.

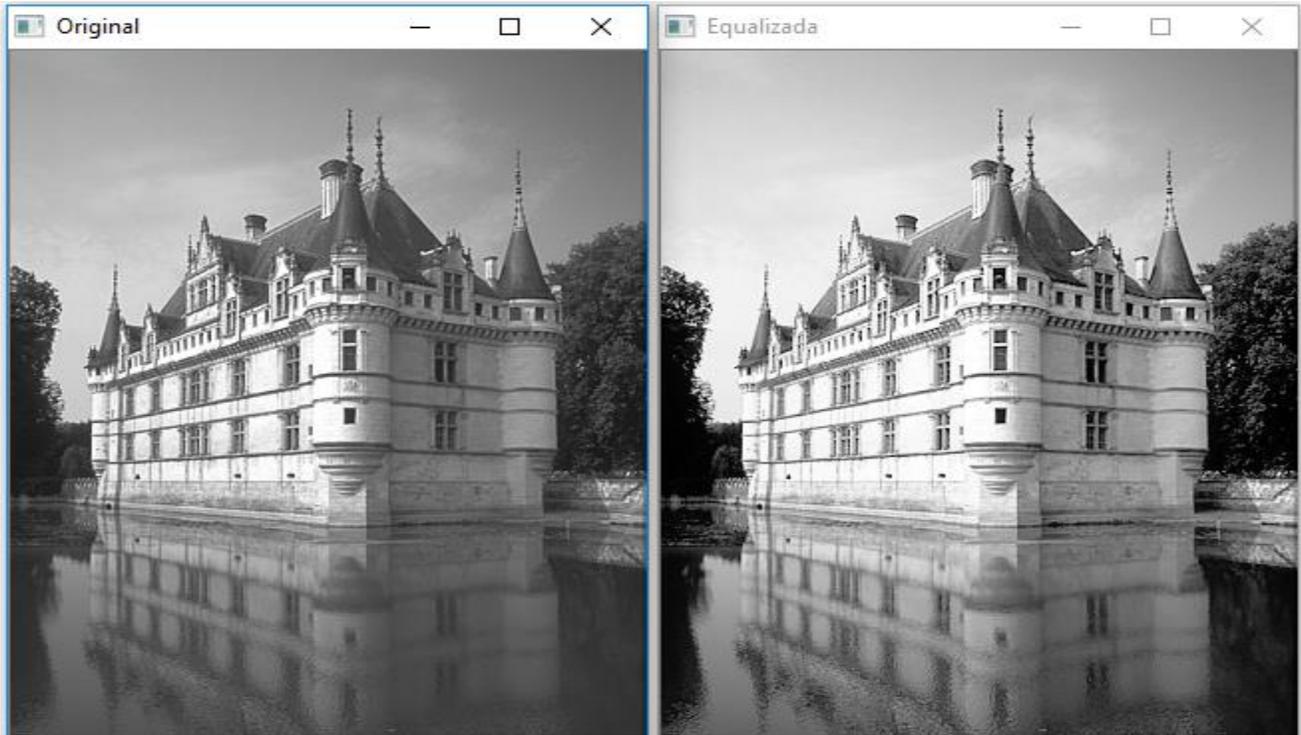


Figura 3.10. Ecuación del histograma.

En la Figura 3.11 se observa la imagen *img_020* tomada del banco de imágenes *SunHays80*, así como las imágenes resultantes de aplicar las cuatro operaciones morfológicas, erosión, dilatación, apertura y cerradura, para las cuales se declaró un elemento estructurante igual a cinco.

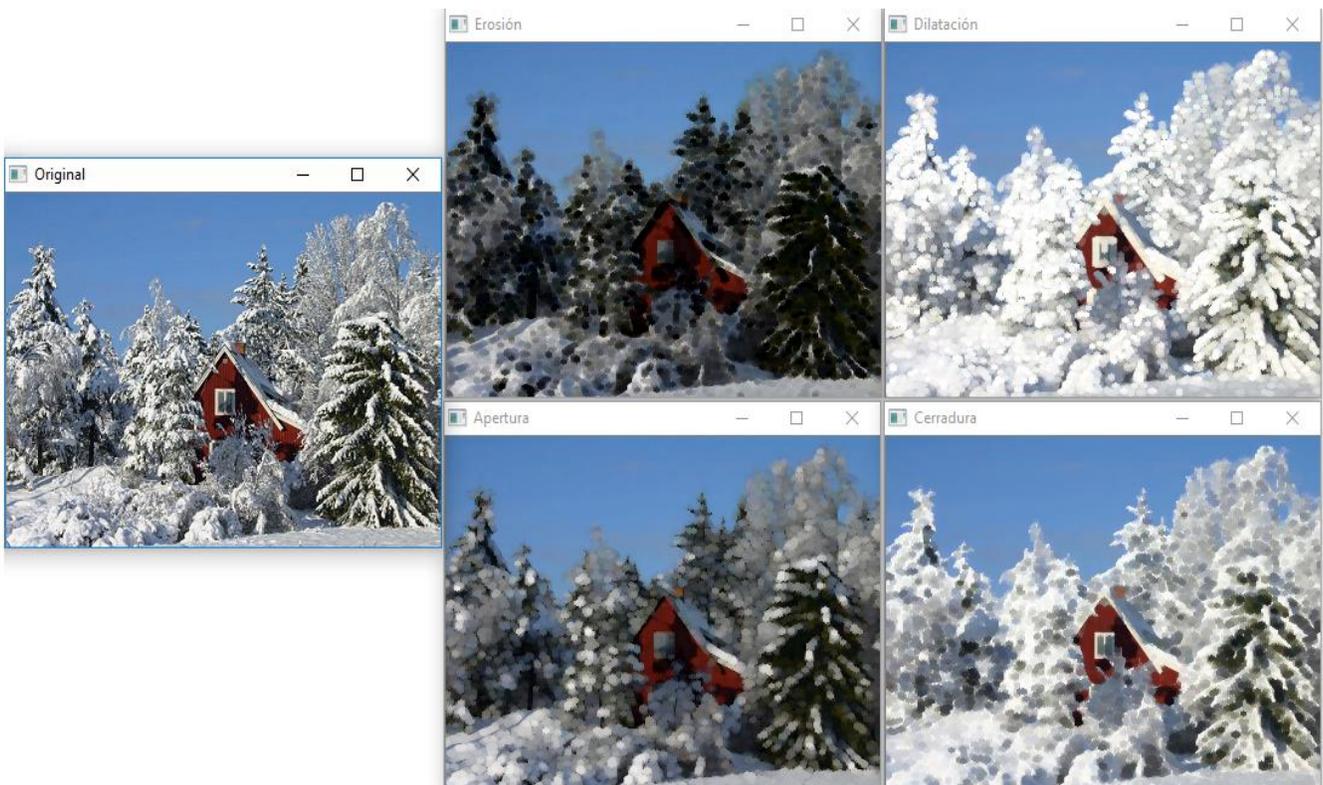


Figura 3.11. Operaciones morfológicas.

La Figura 3.12 permite visualizar la imagen *img_009* del banco de imágenes *Urban100* y la ejecución de las transformaciones geométricas, se muestra la imagen trasladada 25 columnas y 50 filas, así como la imagen rotada 90° y la imagen escalada al 80% de su tamaño original.

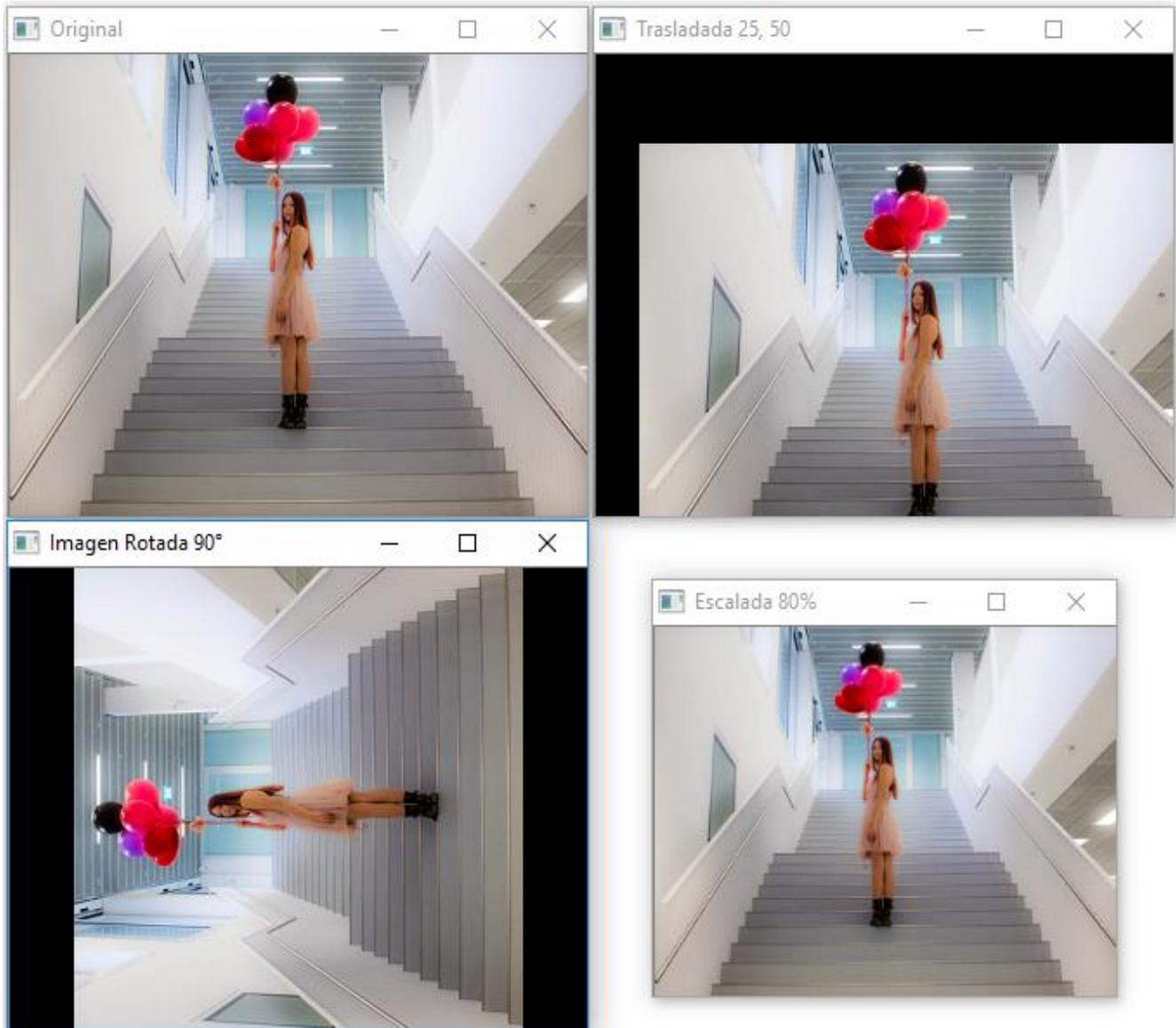


Figura 3.12. Transformaciones geométricas.

3.9 Conclusión

En este capítulo se llevó a cabo el análisis de requerimientos, estableciendo ocho requerimientos necesarios para la implementación y funcionamiento del *framework*, se realizó el diseño de la interfaz de usuario delimitando sus componentes, módulos, menús y submenús. Se mencionó el software mediante el cual se realizaron los desarrollos e implementaciones de cinco algoritmos clásicos de SR, cuatro métricas para evaluar la calidad de las imágenes resultantes, técnicas para el preprocesamiento de imágenes, la funcionalidad para agregar nuevas técnicas, así como la creación del intérprete el cual facilita el uso del *framework*. El siguiente capítulo consta de las experimentaciones realizadas para la validación del funcionamiento *framework*, así como el análisis de los resultados obtenidos respecto a lo comprometido al inicio de la tesis.

Capítulo 4 . Experimentación y resultados

4.1 Introducción

En este capítulo se muestran las pruebas realizadas para la validación del funcionamiento del *framework*, así como las instrucciones utilizadas para la aplicación de las técnicas mencionadas en el capítulo 3. También incluye las secciones para el análisis de los resultados, se muestran los compromisos alcanzados, otros logros así como una conclusión referente al capítulo.

4.2 Banco de imágenes

De acuerdo con la revisión del estado del arte (Agustsson & Timofte, 2017), (Sharma et al., 2017), (Yang et al., 2018), para llevar a cabo las pruebas del *framework*, se realizó la descarga de 5 bancos de imágenes *Set5*, *Set14*, *BSD100*, *Urban100* y *Sun-Hays80*, obteniendo un total de 299 imágenes en formato *png* con diferentes dimensiones en alta resolución, cada una con su correspondiente imagen de baja resolución (ver Anexo B).

4.3 Implementación de pruebas

A continuación se muestran las diez pruebas realizadas para validar el cumplimiento de los diferentes requerimientos y su adecuado funcionamiento dentro del *framework*.

4.3.1 Prueba 01

Nombre: Elementos de la interfaz de usuario.

Objetivo: Esta prueba se enfoca en validar el cumplimiento y correcto funcionamiento del requerimiento R-01 Interfaz de usuario ergonómica, la Figura 4.1 muestra la interfaz de usuario, la cual cuenta con: diez menús, área para las subventanas, ventanas para el intérprete y el historial, y barras de herramientas, los cuales se ejemplificaron en la Figura 3.2.

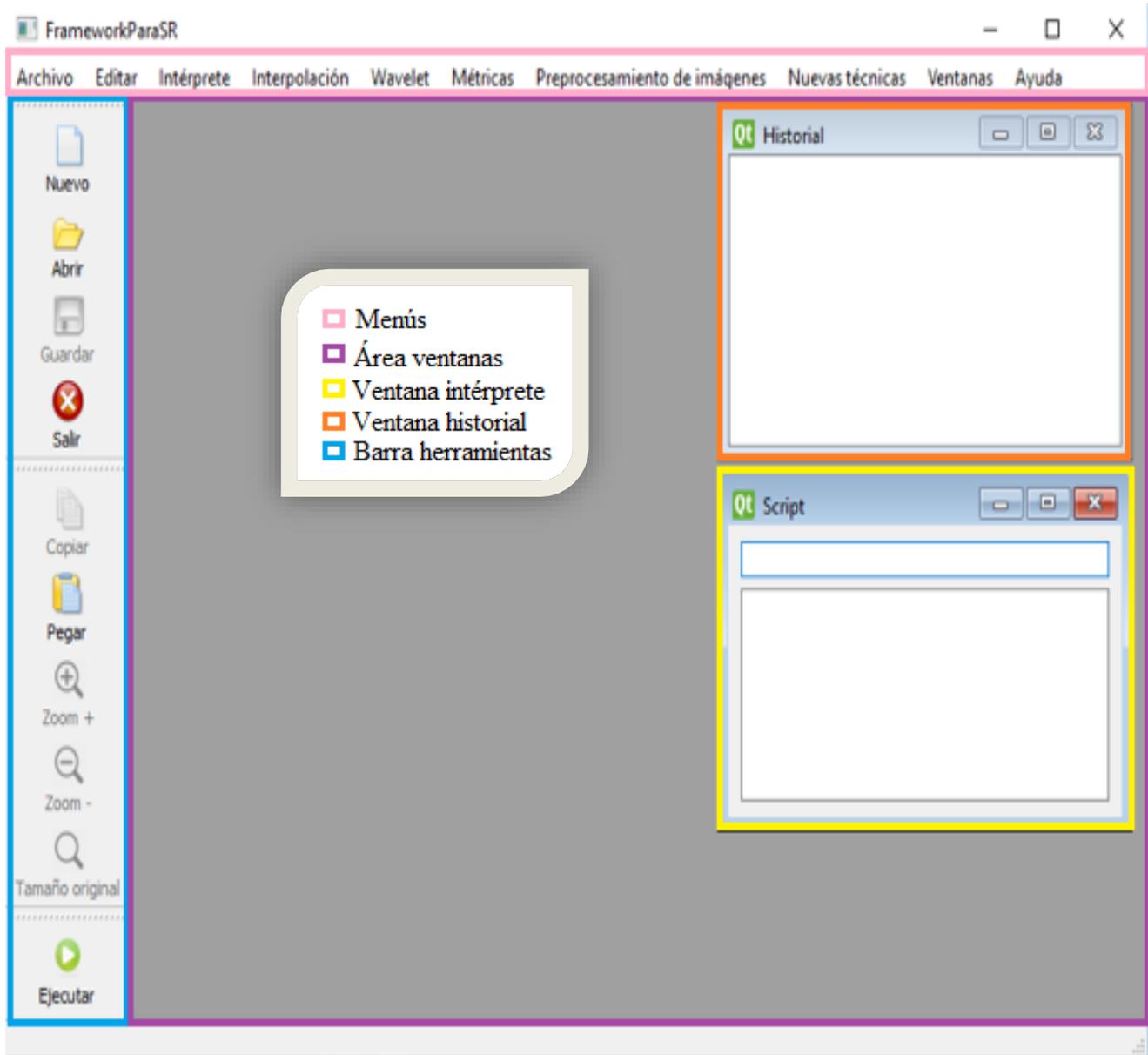


Figura 4.1. Prueba 01: Elementos de la interfaz de usuario.

4.3.2 Prueba 02

Nombre: Menús y submenús de la interfaz de usuario.

Objetivo: Esta prueba permite validar que el *framework* cuenta con los diez menús y submenús establecidos en el análisis de requerimientos (Figura 3.3), así como el correcto funcionamiento de los mismos, en la Figura 4.2 se pueden observar los menús y submenús que componen el sistema.

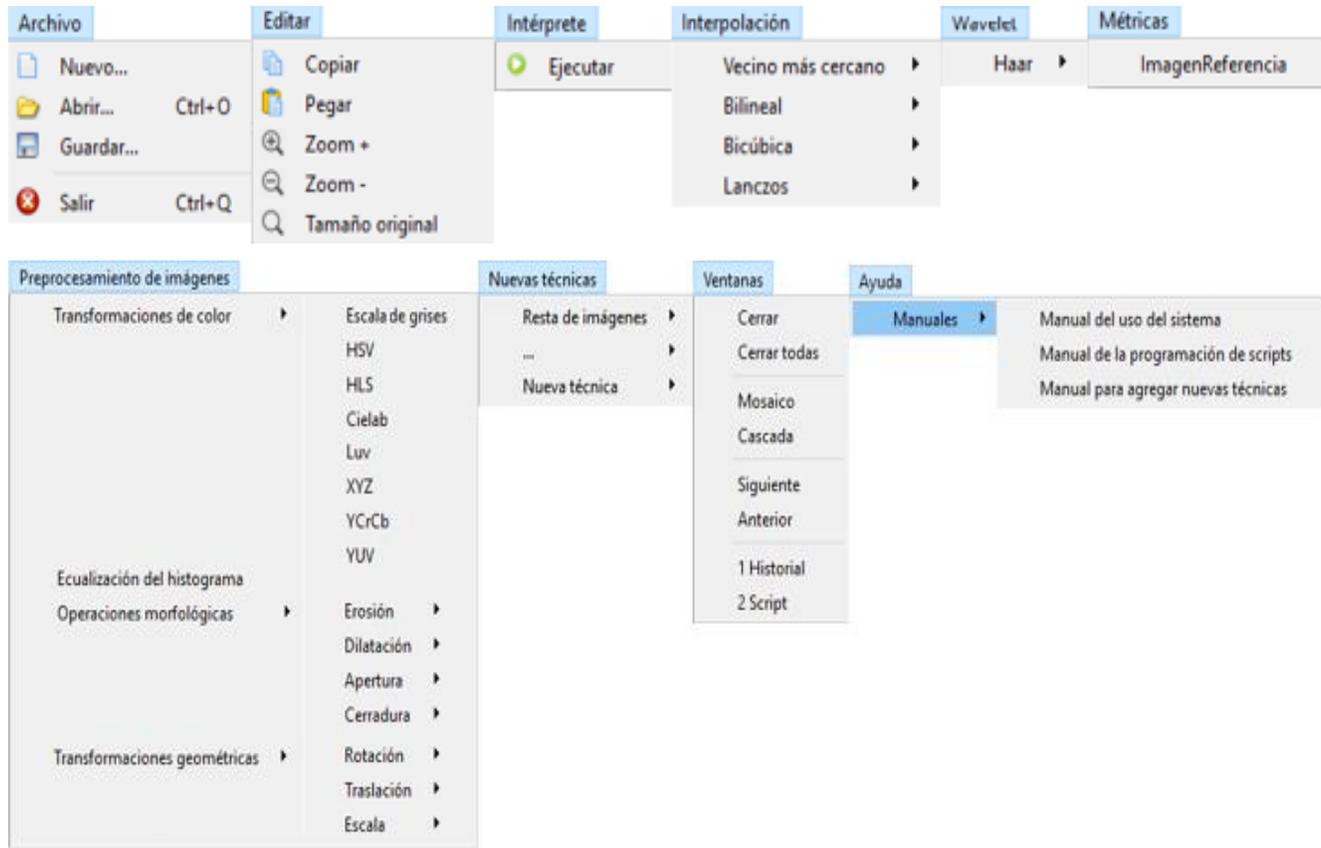


Figura 4.2. Prueba 02: Menús y submenús de la interfaz de usuario.

4.3.3 Prueba 03

Nombre: Lectura y escritura de diversos formatos gráficos.

Objetivo: Esta prueba permite validar el cumplimiento y correcto funcionamiento del requerimiento R-04 Lectura y escritura de diversos formatos gráficos. Es posible abrir archivos y visualizarlos dentro del *framework*, así como realizar copias de los archivos abiertos y guardar dichos archivos, ya sea modificados o con diferentes formatos gráficos.

Secuencia: Para abrir un archivo es necesario seleccionar el menú *Archivo* después *Abrir* (o bien hacer *clic* directamente en la barra de herramientas en el icono de *Abrir*) dirigirse a la carpeta donde se encuentra el archivo que se desea abrir, seleccionar uno o más archivos y dar *clic* en *Abrir*.

Para guardar un archivo es necesario seleccionar la ventana que lo contiene, después seleccionar el menú *Abrir* y a continuación *Guardar* (o bien hacer *clic* directamente en la barra de herramientas en el icono de *Guardar*), seleccionar la ruta en donde se desea almacenar el archivo, colocar el nombre del archivo y su extensión (en caso de que se requiera modificar) y por último hacer *clic* en *Guardar*.

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García.

Para copiar un archivo se debe seleccionar la ventana que lo contiene, después seleccionar el menú *Editar* y la opción de *Copiar* (o bien hacer *clic* directamente en la barra de herramientas en el icono de *Copiar*). Para pegar el archivo copiado se debe seleccionar el menú *Editar* y la opción de *Pegar* (o bien hacer *clic* directamente en la barra de herramientas en el icono de *Pegar*).

Resultados: Al abrir, se mostrarán los archivos seleccionados en nuevas subventanas dentro del *framework*. Al guardar, el archivo será almacenado en la ruta seleccionada, con el nombre y extensión que el usuario determine. Al copiar y pegar, se copiará la imagen seleccionada y se almacenará en el portapapeles hasta que se decida pegarla, al realizar dicha acción se creará una nueva ventana con la imagen copiada y al nombre de esa ventana se le agregará la palabra *Copia_* seguida por el nombre de la imagen copiada. Para llevar a cabo esta prueba se utilizaron las imágenes *baboon*, *comic*, *lenna* y *monarch* del banco de imágenes *Set14*, a dichas imágenes se les realizó el cambio de extensión obteniendo las imágenes *baboon_HR.png*, *comic_HR.tiff*, *lenna_HR.jpg* y *monarch_HR.bmp*. En las siguientes figuras (4.3, 4.4, 4.5, 4.6, 4.7 y 4.8) se observan la funciones de lectura y escritura (para abrir, mostrar, guardar, copiar y pegar) de los formatos gráficos formatos *bmp*, *jpg*, *png* y *tiff*. La Figura 4.3 ejemplifica el proceso para seleccionar y abrir un archivo, en este caso se seleccionaron cinco archivos, cuatro son imágenes en los formatos *bmp*, *jpg*, *png*, *tiff*, y el script en formato *txt*.

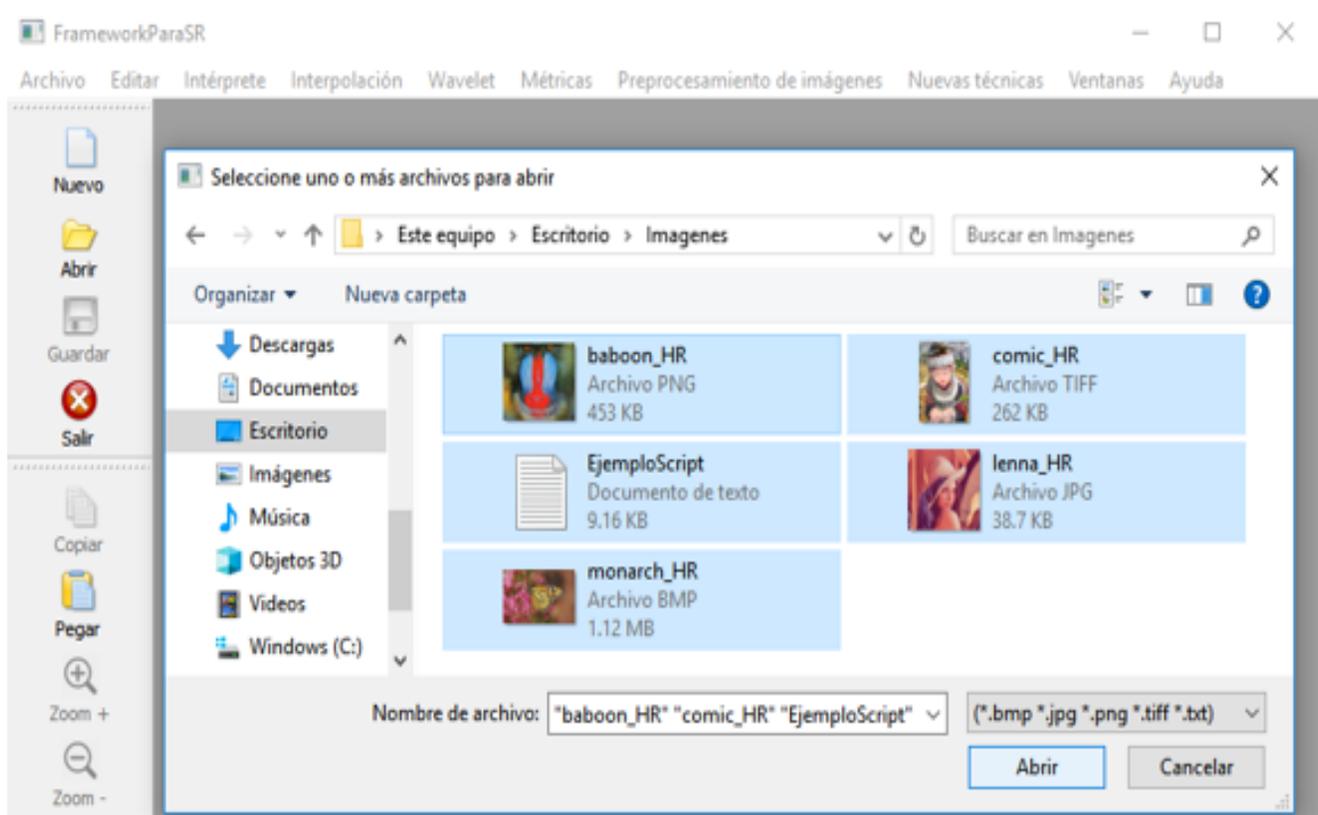


Figura 4.3. Prueba 03: Abrir imágenes y script.

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García. En la Figura 4.4 se observa cómo el sistema abre y muestra los archivos seleccionados, en esta imagen también es posible validar el requerimiento R-02 Interfaz de Múltiples Documentos (*MDI*), ya que cada archivo se muestra en una nueva subventana, las acciones que se realicen serán aplicadas en la ventana seleccionada.

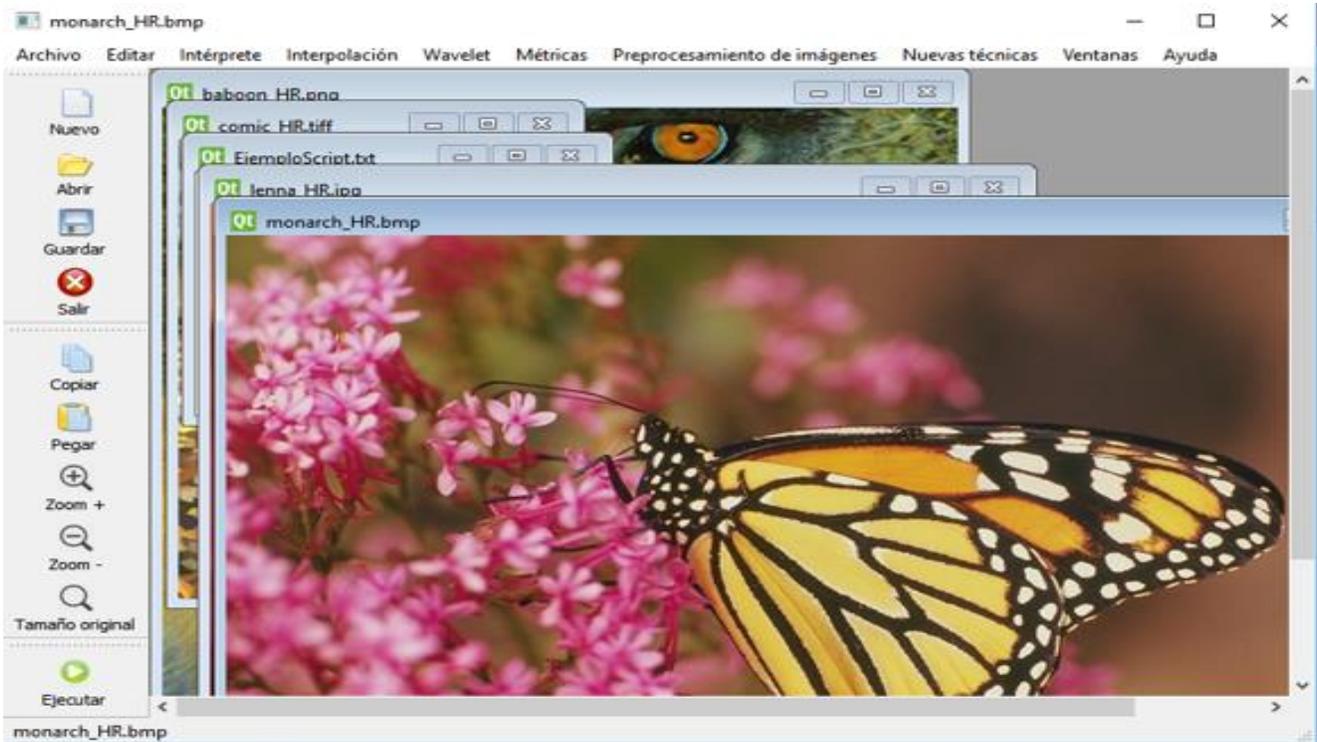


Figura 4.4. Prueba 03: Mostrar imágenes y script.

La Figuras 4.5, ejemplifica el proceso de guardar un *script* (es un archivo *.txt* el cual contiene indicaciones para ser ejecutadas por el intérprete) con el nombre *EjemploScriptModificado.txt*.

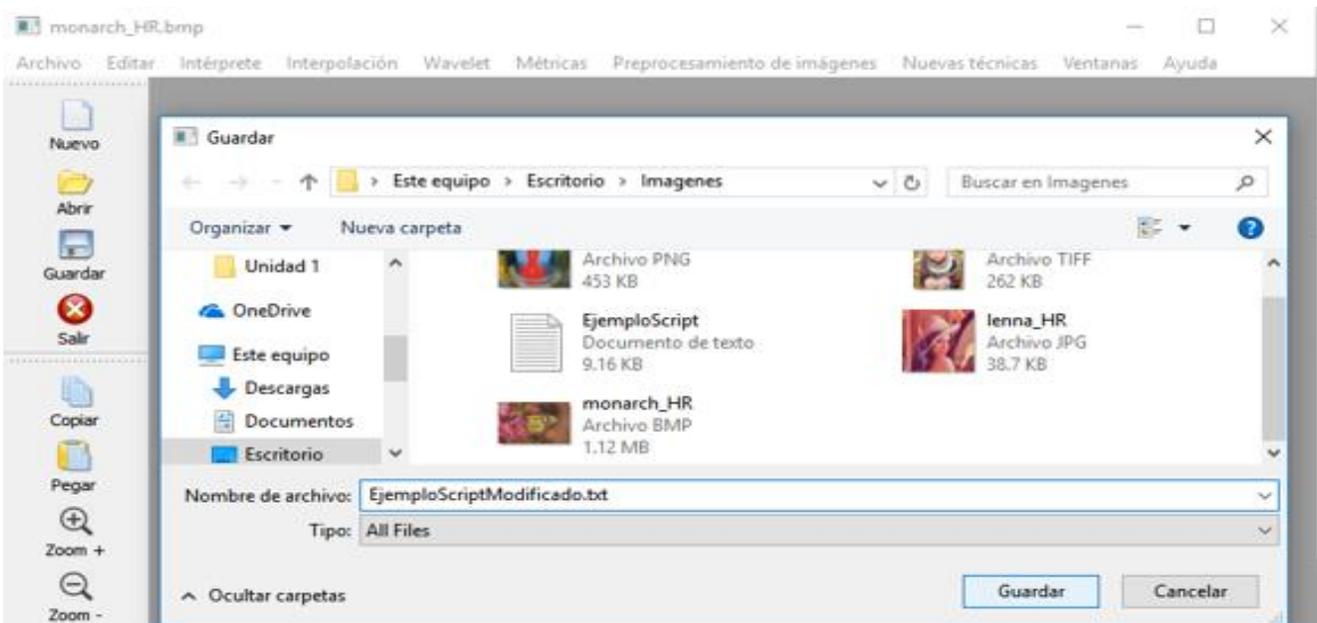


Figura 4.5. Prueba 03: Guardar script.

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García.

Las Figuras 4.6 y 4.7 ejemplifican el proceso de guardar una imagen en formato *.tiff* con el nombre de *Copia_comic_HR.tiff*, y cómo se visualizan los archivos almacenados en la ruta seleccionada.

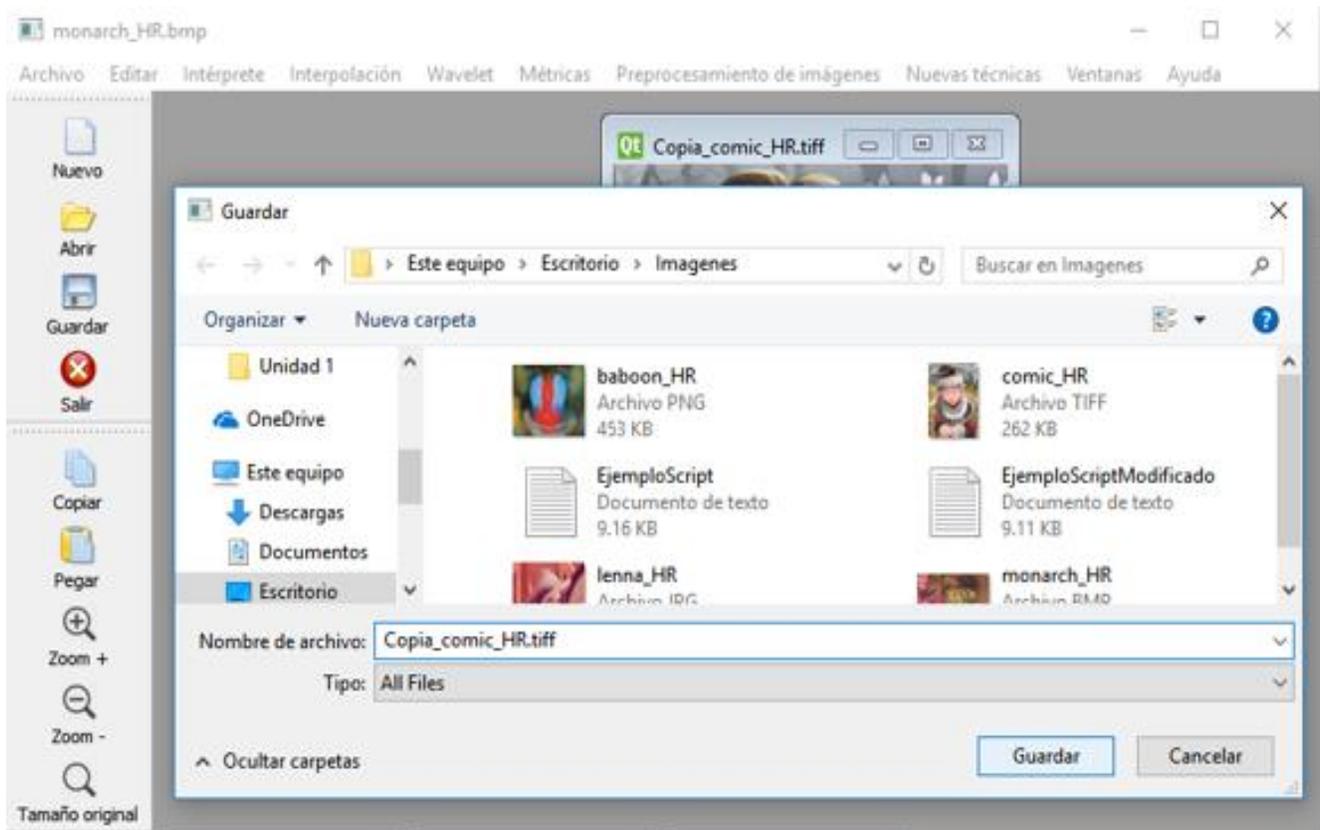


Figura 4.6. Prueba 03: Guardar imagen.

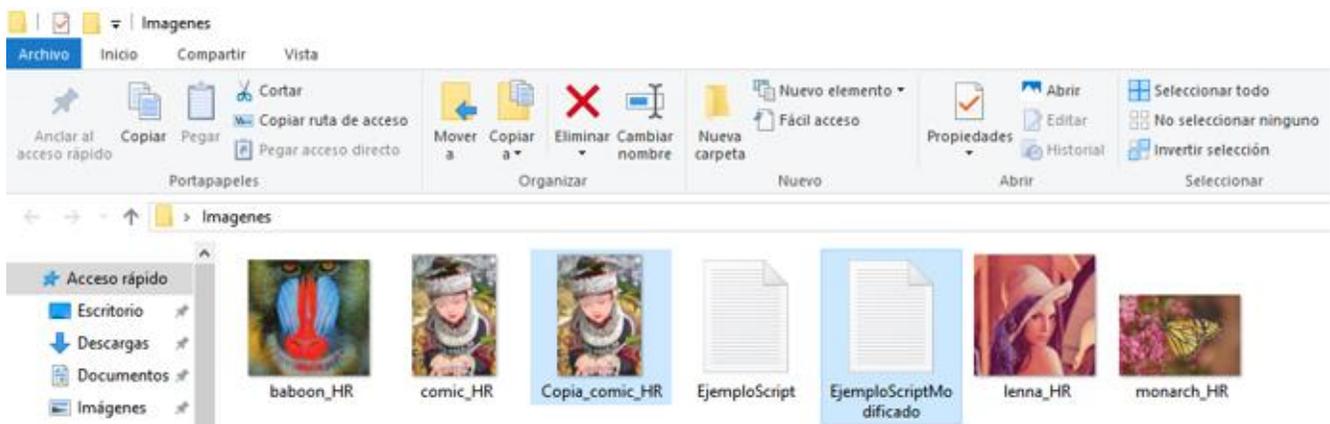


Figura 4.7. Prueba 03: Guardar imagen y script.

En la Figura 4.8 se puede observar el resultado de copiar y pegar una imagen, se crea una nueva ventana con la copia de la imagen seleccionada y en el título de la ventana, se muestra al inicio, la palabra *Copia_*.



Figura 4.8. Prueba 03: Copiar y pegar una imagen.

4.3.4 Prueba 04

Nombre: Algoritmos clásicos de Súper Resolución.

Objetivo: Esta prueba se enfoca en validar el cumplimiento del requerimiento R-05 Algoritmos clásicos de Súper Resolución, verificando el correcto funcionamiento de las cinco técnicas de SR agregadas al *framework* (las interpolaciones vecino más cercano, bilineal, bicúbica, *lanz*os así como la *wavelet haar*).

Secuencia: Para aplicar una técnica de SR es necesario seleccionar una imagen y después hacer *clic* en el menú *Interpolación* o en *Wavelet* y seleccionar algún submenú: *Vecino más cercano*, *Bilineal*, *Bicúbica*, *Lanzos* o *Haar*, después introducir el valor de la *Escala* (en un elemento de los números naturales (\mathbb{N})) a la que se desea aumentar la imagen y por último pulsar la tecla *Enter*.

Resultados: Se creará una nueva ventana con la imagen aumentada a la escala establecida, en el título contendrá el nombre de la imagen original así como la técnica y la escala aplicadas.

Para llevar a cabo esta prueba se utilizaron las imágenes *img_017*, *img_077*, *img_033*, *img_039* y *man* de los bancos de imágenes *Urban100*, *SunHays80* y *Set14*, a dichas imágenes se les aplicó las cinco técnicas de SR, las interpolaciones Vecino más cercano, Bilineal, Bicúbica y *Lanzos* así como la *Wavelet Haar*, todas con un factor de escala igual a 4. A continuación, se observan los resultados de aplicar alguna

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García. técnica a las imágenes mencionadas, la Figura 4.9 muestra la imagen obtenida al aplicar la interpolación del vecino más cercano 4X a la imagen *img_017*.

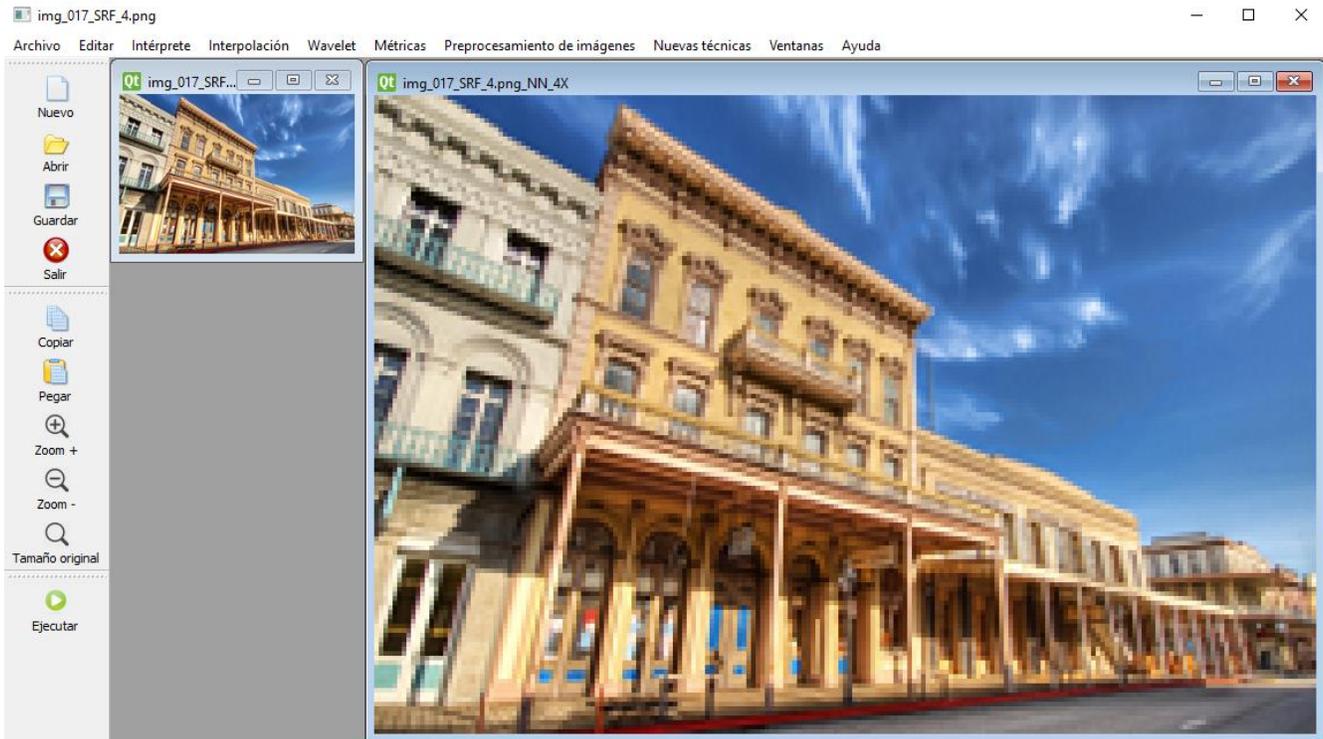


Figura 4.9. Prueba 04: Interpolación del vecino más cercano 4X.

La Figura 4.10 muestra la imagen obtenida al aplicar la interpolación bilineal 4X a la imagen *img_077*.

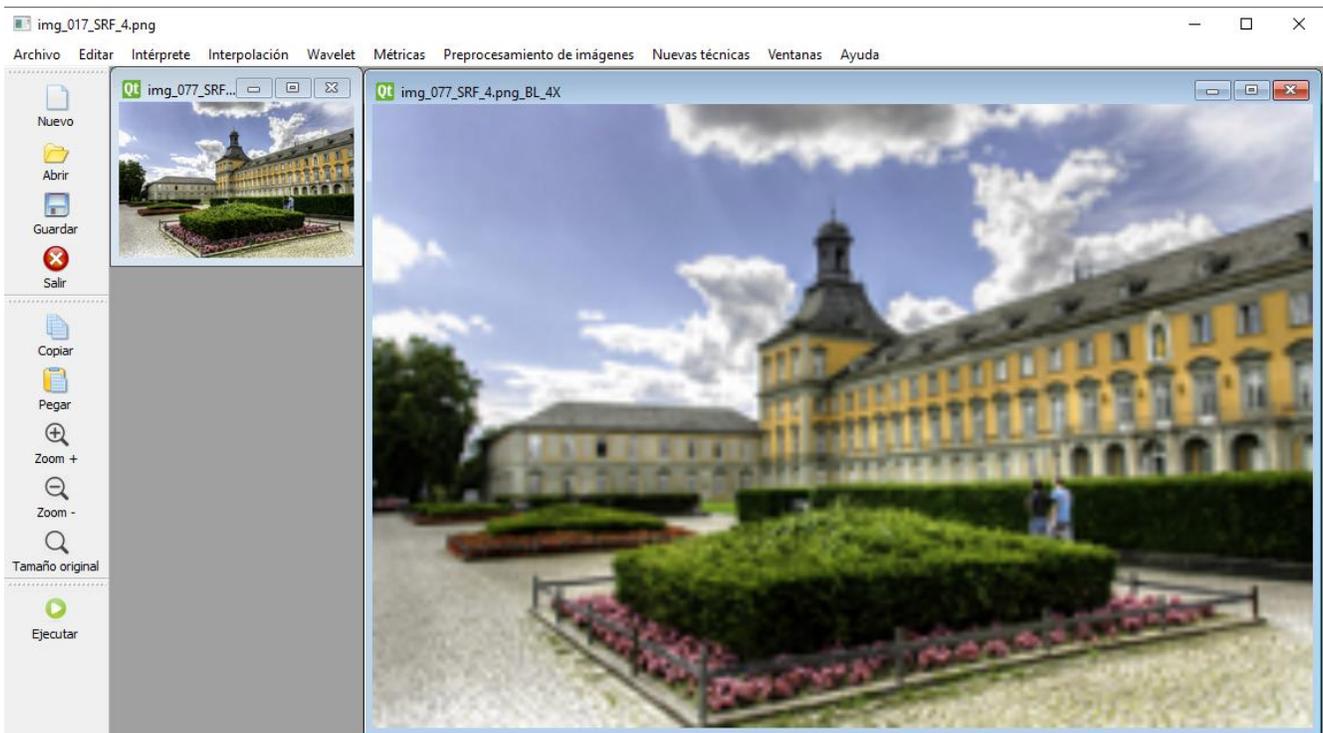


Figura 4.10. Prueba 04: Interpolación bilineal 4X.

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García.

En la Figura 4.11 se observa la imagen obtenida al aplicar la interpolación bicúbica 4X a la imagen *img_033*.

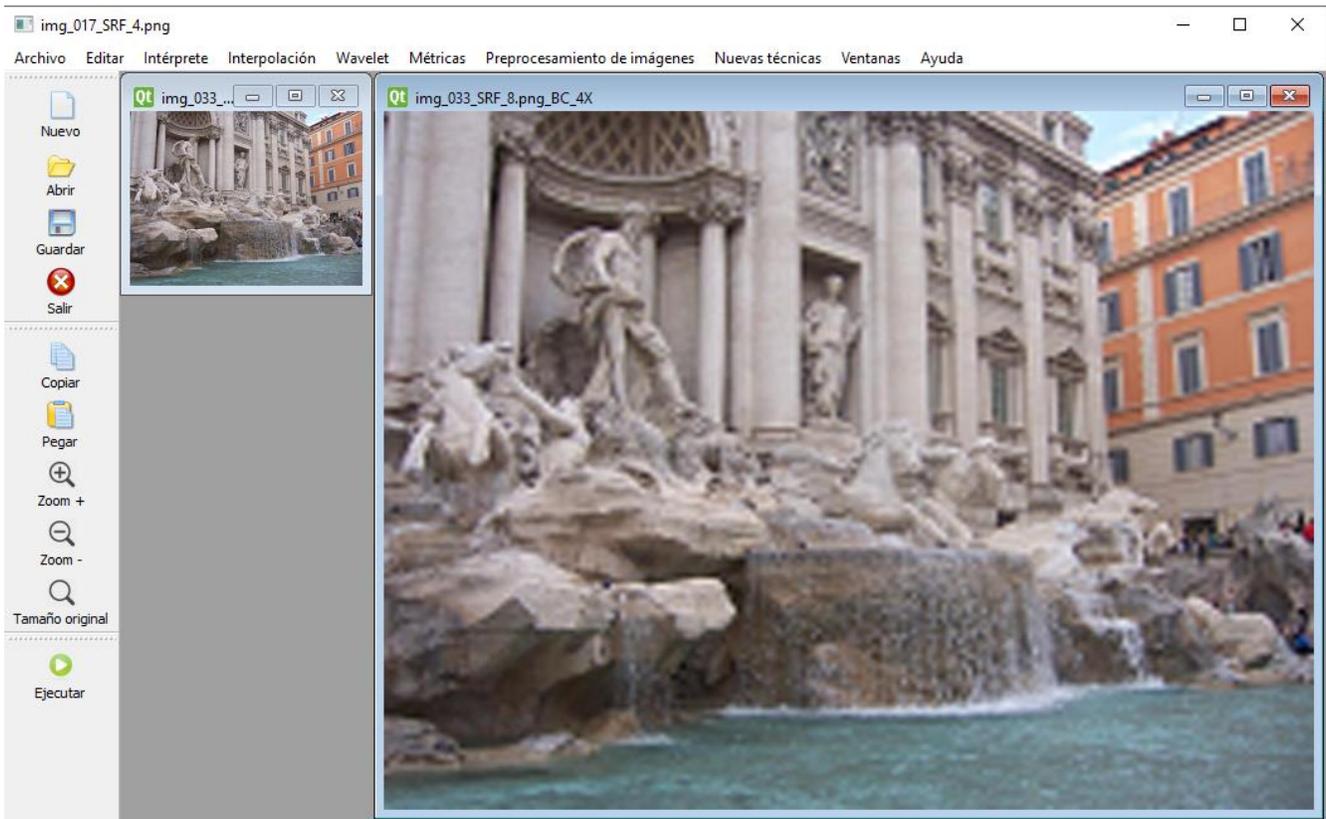


Figura 4.11. Prueba 04: Interpolación bicúbica 4X.

La Figura 4.12 muestra la imagen obtenida al aplicar la interpolación *lanczos* 4X a la imagen *img_039*.

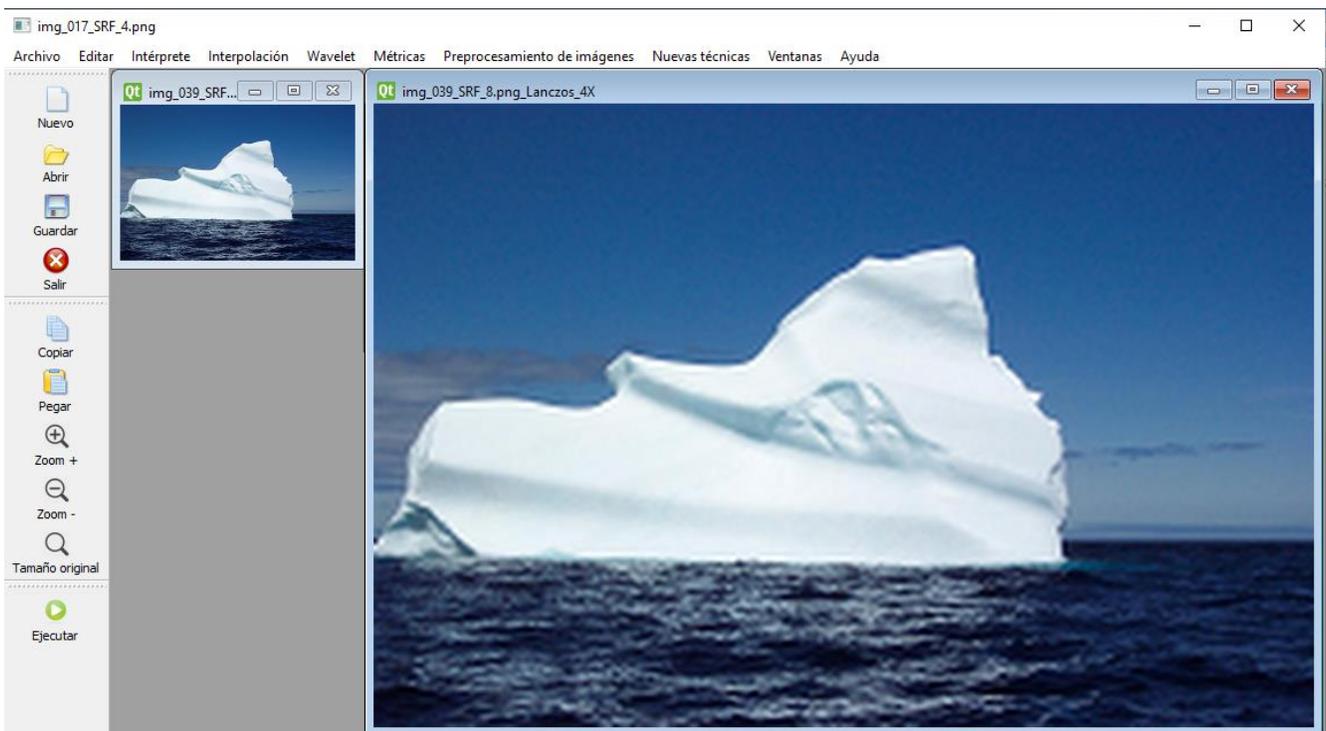


Figura 4.12. Prueba 04: Interpolación *Lanczos* 4X.

En la Figura 4.13 se muestra la descomposición de la *wavelet haar* 4X de la imagen *man*.

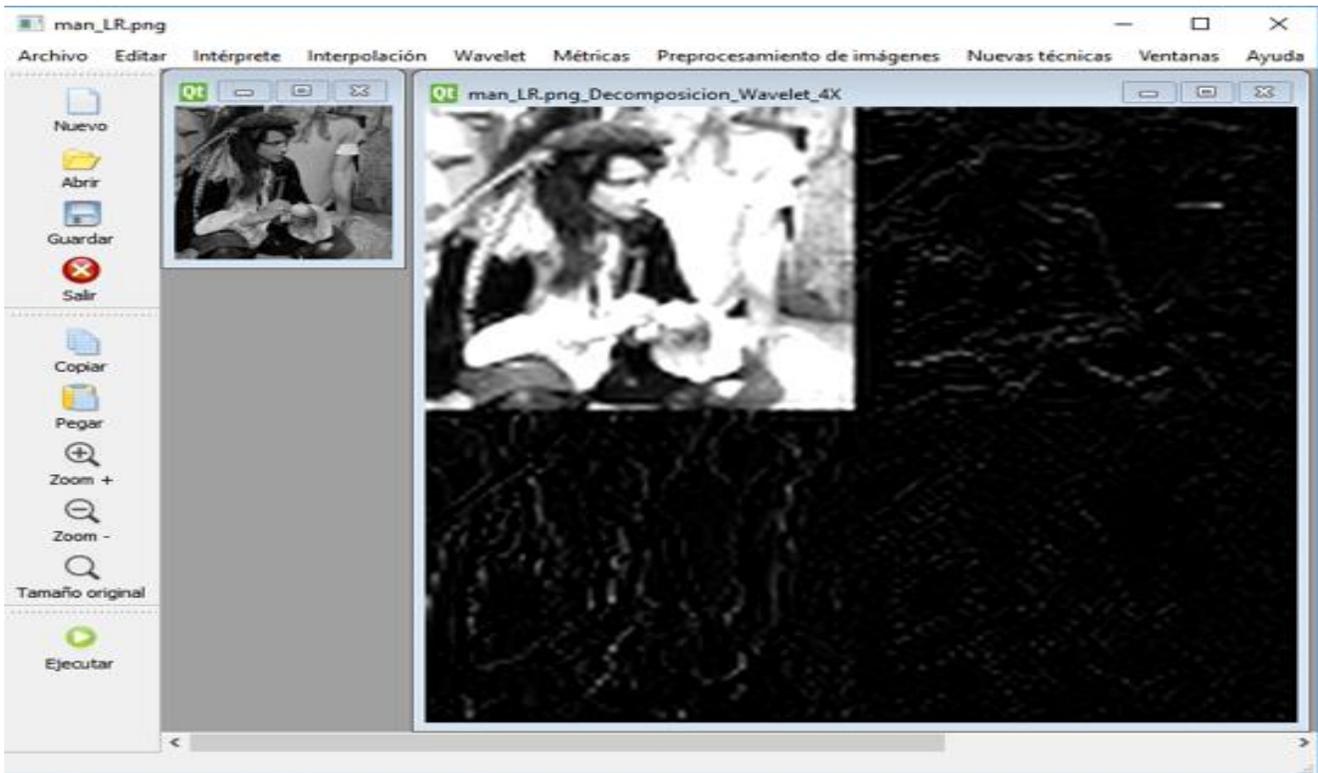


Figura 4.13. Prueba 04: Descomposición Wavelet haar 4X.

En la Figura 4.14 se observa la imagen obtenida al aplicar la *wavelet haar* 4X a la imagen *man*.

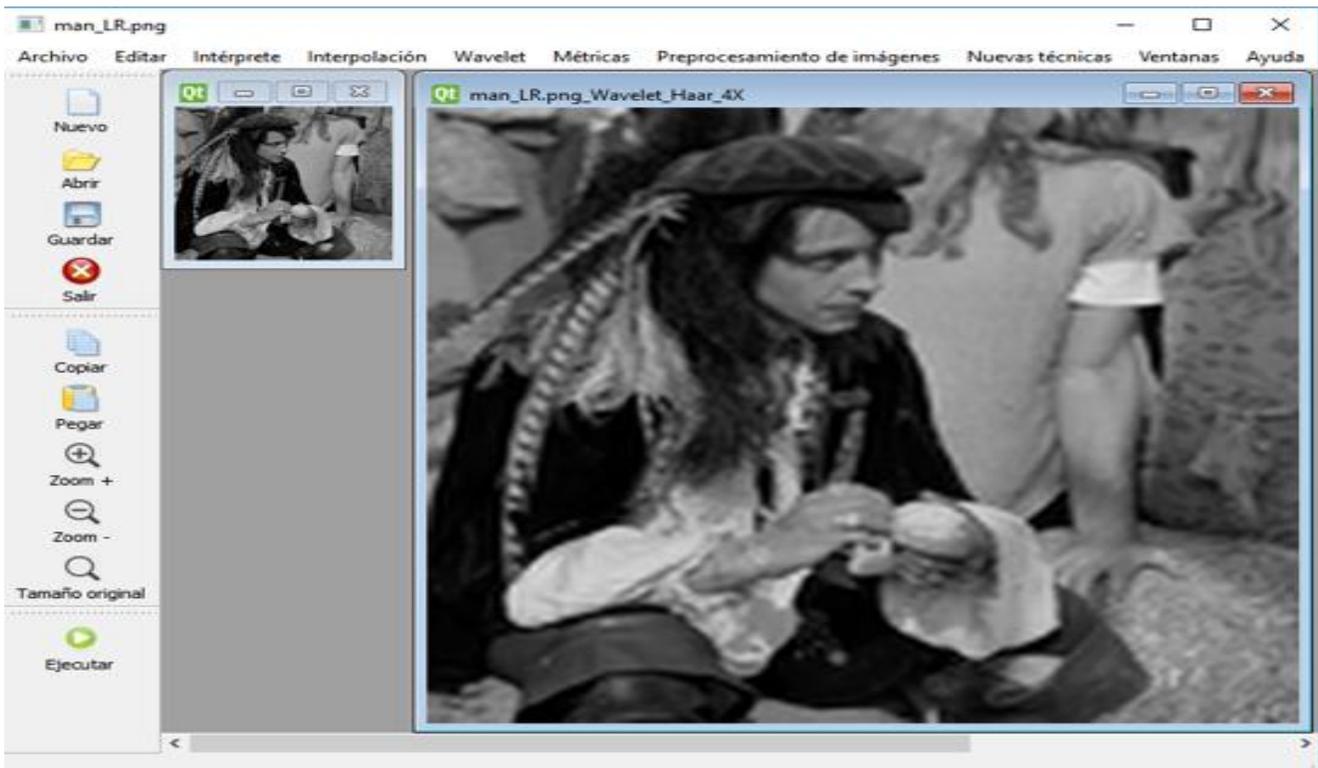


Figura 4.14. Prueba 04: Wavelet haar 4X.

4.3.5 Prueba 05

Nombre: Métricas para evaluar la calidad de las imágenes.

Objetivo: En esta prueba se valida la implementación y correcto funcionamiento de las cuatro métricas (Error Cuadrático Medio *MSE*, Raíz del Error Cuadrático Medio *RMSE*, Proporción Máxima de Señal a Ruido *PSNR* e Índice de Similitud Estructural *SSIM*) para evaluar la calidad de las imágenes obtenidas después de aplicar alguna técnica de SR.

Secuencia: Para aplicar las métricas de evaluación es necesario seleccionar una imagen y después hacer *clic* en el menú *Métricas* y después *clic* en *ImagenReferencia*, a continuación, seleccionar la imagen de referencia y hacer *clic* en *Abrir*.

Resultados: Se crea una nueva ventana con la imagen de referencia, así como una ventana con una tabla la cual consta de siete columnas, la primera contiene el nombre de la imagen a *Evaluar*, la segunda la ruta y nombre de la imagen de *Referencia*, en la tercera la *Técnica* que se le aplicó a la imagen original, así como el resultado obtenido de las cuatro métricas que evalúan la calidad de las imágenes *MSE*, *RMSE*, *PSNR* y *SSIM*.

En la Figura 4.15 se muestra la imagen de referencia así como la imagen a evaluar y la tabla que muestra los valores obtenidos para cada métrica de evaluación.

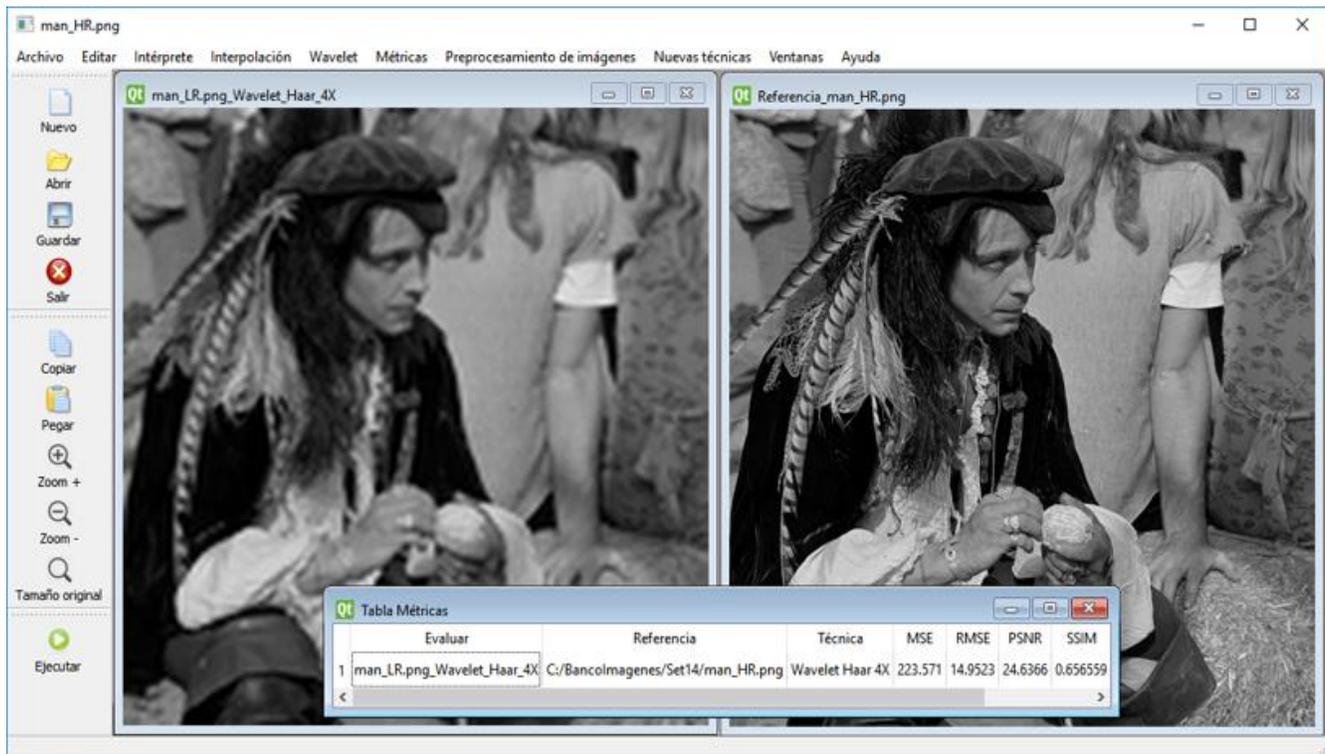


Figura 4.15. Prueba 05: Métricas de evaluación.

La Tabla 4.1. muestra los resultados de las cuatro métricas (Error Cuadrático Medio *MSE*, Raíz del Error Cuadrático Medio *RMSE*, Proporción Máxima de Señal a Ruido *PSNR* e Índice de Similitud Estructural *SSIM*) para evaluar la calidad de las imágenes obtenidas al aplicar las cinco técnicas de SR (*Interpolación Vecino más cercano NN*, *Bilineal BL*, *Bicúbica BC*, *Lanczos* y *Wavelet Haar*) con una escala igual a 4, de cinco imágenes *img_017*, *img_077*, *img_033*, *img_039* y *man*.

Tabla 4.1. Resultados de aplicar las métricas a las imágenes mostradas en la prueba 04.

Evaluar	Referencia	Técnica	MSE	RMSE	PSNR	SSIM
<i>img_017_LR.png</i>	<i>Urban100/img_017_HR.png</i>	<i>NN 4X</i>	497.911	22.3139	21.1593	0.641736
		<i>BL 4X</i>	443.289	21.0544	21.6639	0.662826
		<i>BC 4X</i>	396.163	19.9039	22.1521	0.688001
		<i>Lanczos 4X</i>	387.518	19.6855	22.2479	0.692582
		<i>W.Haar 4X</i>	991.205	31.4834	18.1692	0.636783
<i>img_077_LR.png</i>	<i>Urban100/img_077_HR.png</i>	<i>NN 4X</i>	628.078	25.0615	20.1507	0.554528
		<i>BL 4X</i>	595.713	24.4072	20.3804	0.560742
		<i>BC 4X</i>	561.232	23.6903	20.6394	0.582297
		<i>Lanczos 4X</i>	554.666	23.5513	20.6905	0.58633
		<i>W.Haar 4X</i>	544.001	23.3238	20.7748	0.590039
<i>img_033_LR.png</i>	<i>SunHays80/img_033_HR.png</i>	<i>NN 4X</i>	353.734	18.8078	22.644	0.525209
		<i>BL 4X</i>	320.068	17.8905	23.0784	0.542914
		<i>BC 4X</i>	290.372	17.0403	23.5013	0.576541
		<i>Lanczos 4X</i>	285.438	16.8949	23.5757	0.582827
		<i>W.Haar 4X</i>	283.185	16.8281	23.6101	0.574824
<i>img_039_LR.png</i>	<i>SunHays80/img_039_HR.png</i>	<i>NN 4X</i>	250.003	15.8115	24.1513	0.748501
		<i>BL 4X</i>	225.916	15.0305	24.5913	0.761063
		<i>BC 4X</i>	211.362	14.5383	24.8805	0.773206
		<i>Lanczos 4X</i>	208.907	14.4536	24.9313	0.775551
		<i>W.Haar 4X</i>	343.394	18.5309	22.7729	0.731562
<i>man_LR.png</i>	<i>Set14/man_HR.png</i>	<i>NN 4X</i>	307.804	17.5443	23.2481	0.593077
		<i>BL 4X</i>	263.727	16.2397	23.9193	0.616291
		<i>BC 4X</i>	229.699	15.1558	24.5192	0.650398
		<i>Lanczos 4X</i>	223.513	14.9504	24.6378	0.656628
		<i>W.Haar 4X</i>	223.571	14.9523	24.6366	0.656559

4.3.6 Prueba 06

Nombre: Nuevas técnicas de Súper Resolución.

Objetivo: En esta prueba se valida la implementación y el funcionamiento para que el usuario pueda agregar nuevas técnicas al *framework*.

Secuencia: Para llevar a cabo dicha funcionalidad, el usuario deberá modificar el archivo *AgregarTecnicas.txt* (dicho archivo se localiza generalmente en la ruta *C:\Program Files\FrameworkParaSR\AgregarTecnicas*, dichas carpetas se crearon al instalar el *framework*) en el cual se debe agregar el nombre de la nueva técnica, la ruta del ejecutable, el número de parámetros que recibe el programa, el tipo de dato para cada parámetro, colocar si el programa será silencioso (no mostrará ninguna ventana, únicamente se ejecutará) o no silencioso (mostrará las ventanas de la ejecución realizada) y por último especificar si la ejecución del subprocesso finalizará al cerrar el *framework* o no.

Resultados: Al agregar la nueva técnica en el archivo *AgregarTecnicas.txt* se visualizarán dentro del *framework*, en el menú *Nuevas técnicas*.

En la Figura 4.16 se muestra un ejemplo del archivo *AgregarTecnicas.txt*, así como la forma en que se visualiza la lista de nuevas técnicas en el *framework*.

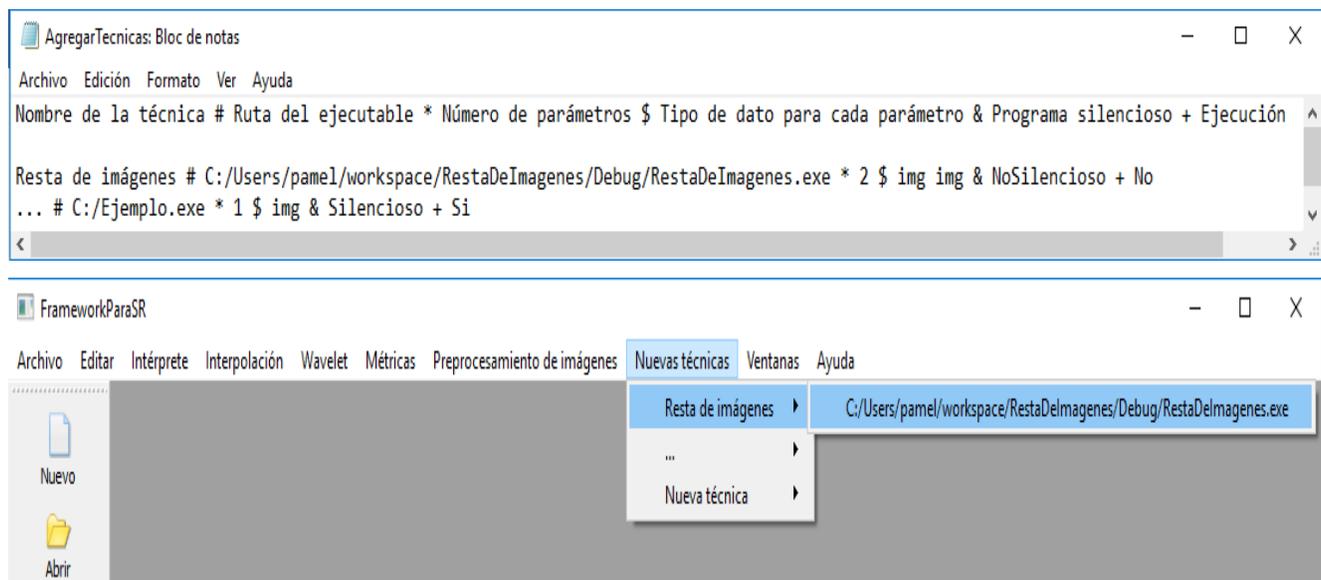


Figura 4.16. Prueba 06: Nuevas técnicas de Súper Resolución.

La Figura 4.17 muestra la ejecución de la nueva técnica agregada denominada Resta de imágenes.



Figura 4.17. Resta de imágenes.

4.3.7 Prueba 07

Nombre: Funcionalidades adicionales (técnicas de preprocesamiento de imágenes y documentación de ayuda).

Objetivo: Para dar evidencia del potencial del *framework* al extender la funcionalidad con algoritmos de Visión Artificial, como son las técnicas de preprocesamiento de las imágenes las cuales incluyen ocho transformaciones de color, ecualización del histograma, cuatro operaciones morfológicas y tres transformaciones geométricas, así como la documentación de ayuda la cual consta de tres manuales, uno para el uso del sistema, otro para la programación de *scripts* y el último para agregar nuevas técnicas.

Secuencia: Para aplicar las transformaciones de color, se debe seleccionar una imagen y después, hacer *clic* en el menú *Procesamiento de imágenes*, a continuación hacer *clic* en *Transformaciones de color* y seleccionar alguna de las opciones *Escala de grises*, *HSV*, *HSL*, *Cielab*, *Luv*, *XYZ*, *YCrCb* o *YUV*.

Para realizar una ecualización del histograma es necesario seleccionar la imagen y después pulsar el menú *Procesamiento de imágenes*, a continuación hacer *clic* en *Ecualización del histograma*.

Para realizar alguna operación morfológica es necesario seleccionar la imagen y a continuación dar *clic* en el menú *Procesamiento de imágenes*, después seleccionar *Operaciones morfológicas* y seleccionar alguna opción *Erosión*, *Dilatación*, *Apertura* o *Cerradura*, a continuación debe ingresar el valor del *Elemento Estructurante* y por último presionar la tecla *Enter*.

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García. Para aplicar las transformaciones geométricas, se debe seleccionar una imagen y después, hacer *clic* en el menú *Procesamiento de imágenes*, a continuación hacer *clic* en *Transformaciones geométricas* y después seleccionar el tipo de transformación a realizar. Si selecciona *Rotación* debe introducir los *Grados a rotar* y por último presionar la tecla *Enter*. Si selecciona *Traslación* debe introducir las *Columnas a trasladar* y presionar la tecla *Enter*, después introducir las *Filas a trasladar* y por último presionar la tecla *Enter*. Si selecciona *Escala* debe introducir el *Porcentaje a escalar* y por último presionar la tecla *Enter*.

Para hacer uso de los manuales de ayuda, se debe seleccionar el menú *Ayuda*, hacer *clic* en *Manuales* y a continuación hacer *clic* en el menú a consultar, el cual puede ser *Manual del uso del sistema*, *Manual de la programación de scripts* o *Manual para agregar nuevas técnicas*.

Resultados: Al aplicar alguna de las técnicas para el preprocesamiento de las imágenes, se crea una nueva ventana con la imagen resultante y en el título se visualiza la técnica aplicada (y sus parámetros si es el caso).

En la Figura 4.18 se observan las imágenes resultantes de aplicar las ocho transformaciones de color a la imagen *img_043* del banco de imágenes *SunHays80*.

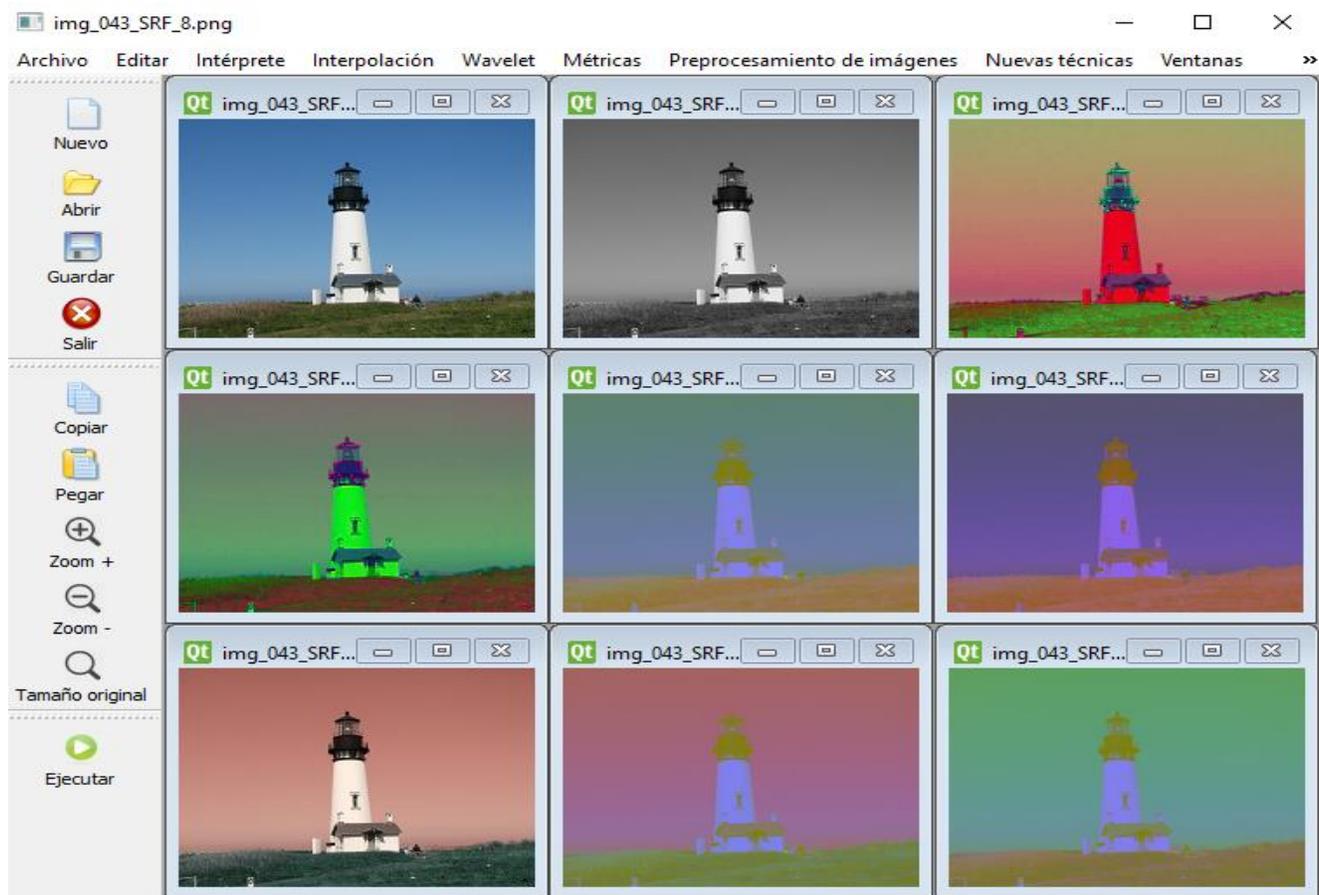


Figura 4.18. Prueba 07: Transformaciones de color.

La Figura 4.19 muestra la imagen *man* del banco de imágenes *Set14*, así como el resultado de aplicarle la ecualización del histograma.



Figura 4.19. Prueba 07: Ecualización del histograma.

En la Figura 4.20 se muestran las imágenes resultantes de aplicar las cuatro operaciones morfológicas con un elemento estructurante igual a cinco a la imagen *bird* del banco de imágenes *Set5*, como muestra de las funciones complementarias.

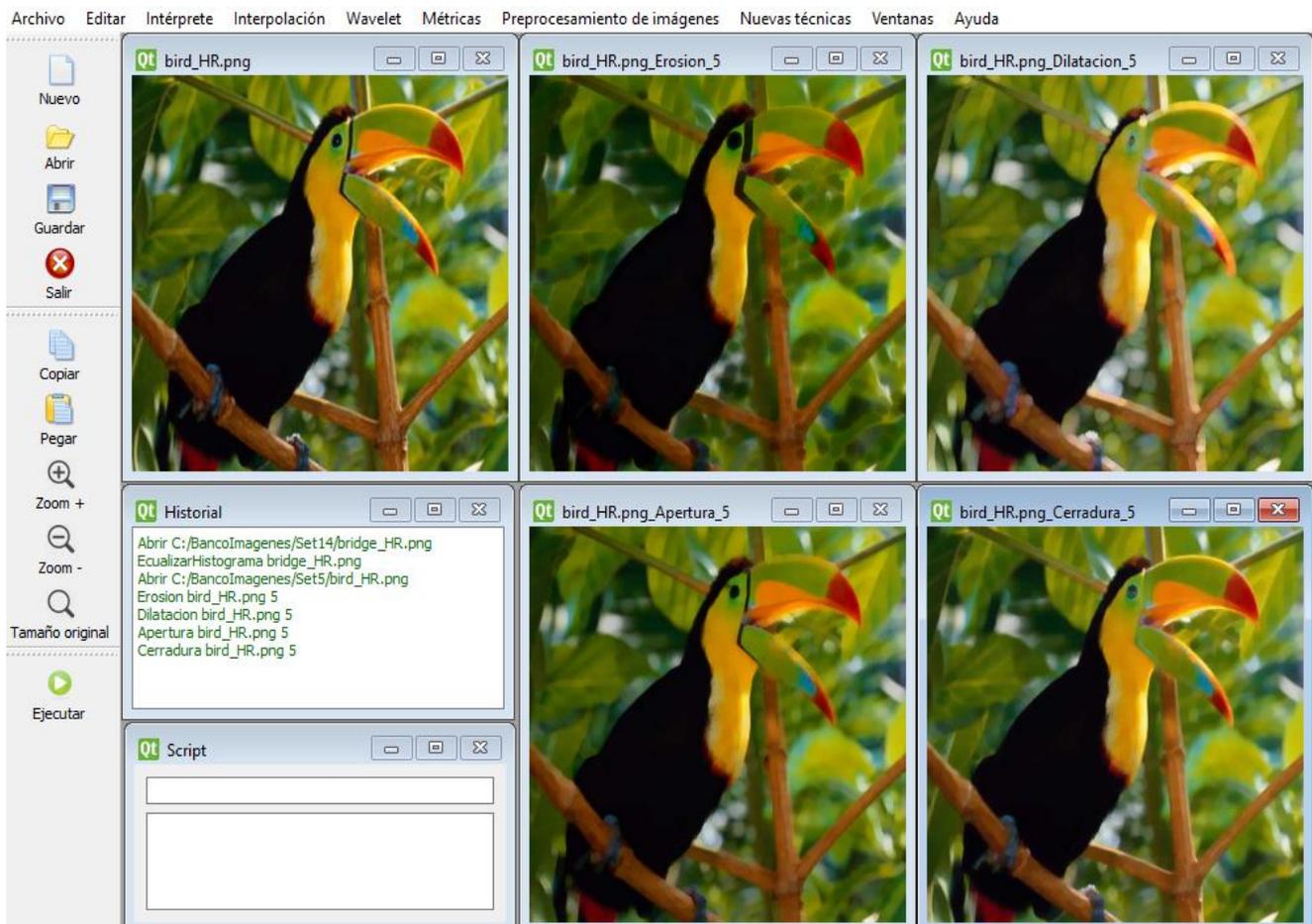


Figura 4.20. Prueba 07: Operaciones morfológicas.

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García. La Figura 4.21 muestra la imagen 108005 del banco de imágenes *BSD100*, a la cual se le aplicaron las tres transformaciones geométricas, se rotó cuarenta y cinco grados, se trasladó 20 filas y 20 columnas y se escaló al ochenta por ciento de su tamaño original.

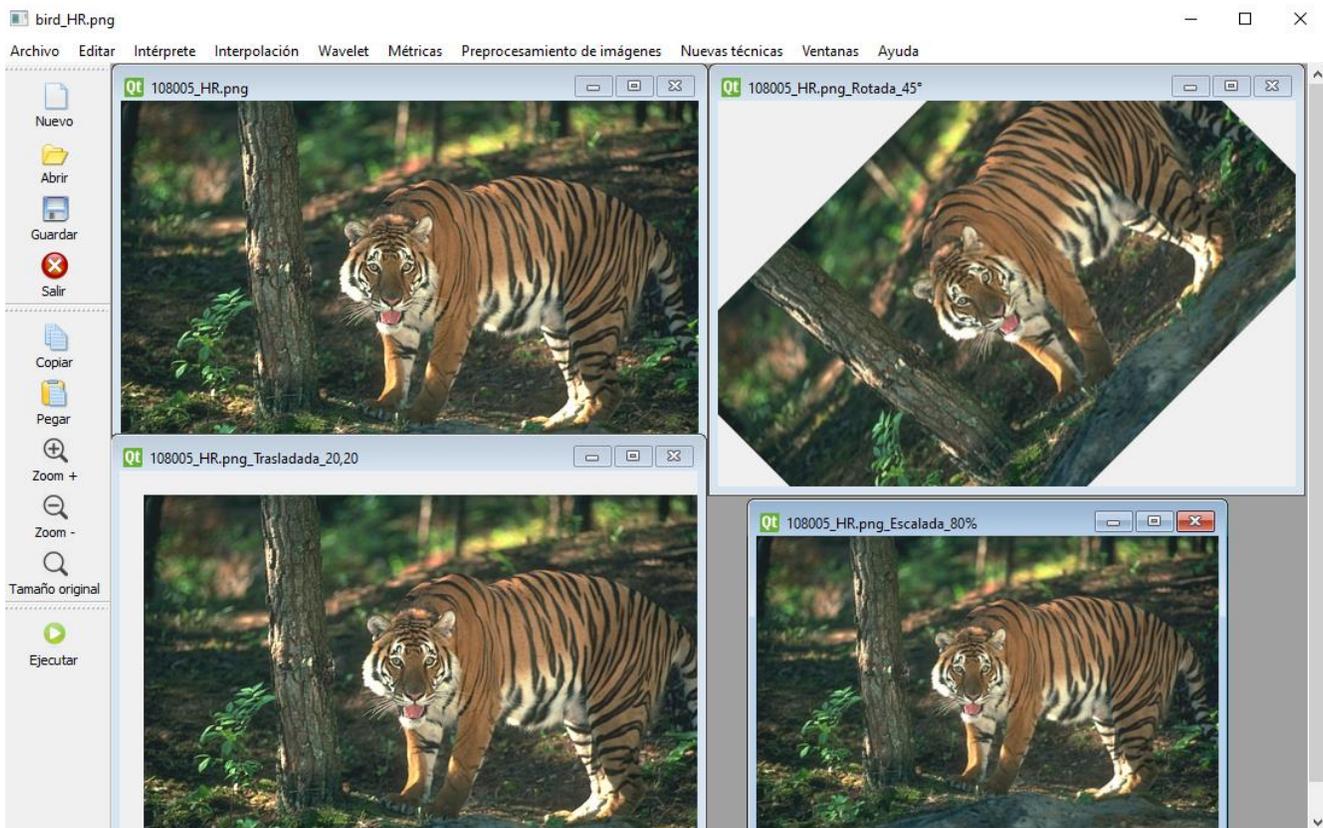


Figura 4.21. Prueba 07: Transformaciones geométricas.

Por último, la Figura 4.22 permite la visualización del menú de ayuda, el cual contiene los tres manuales así como la muestra de acceder alguno de ellos.

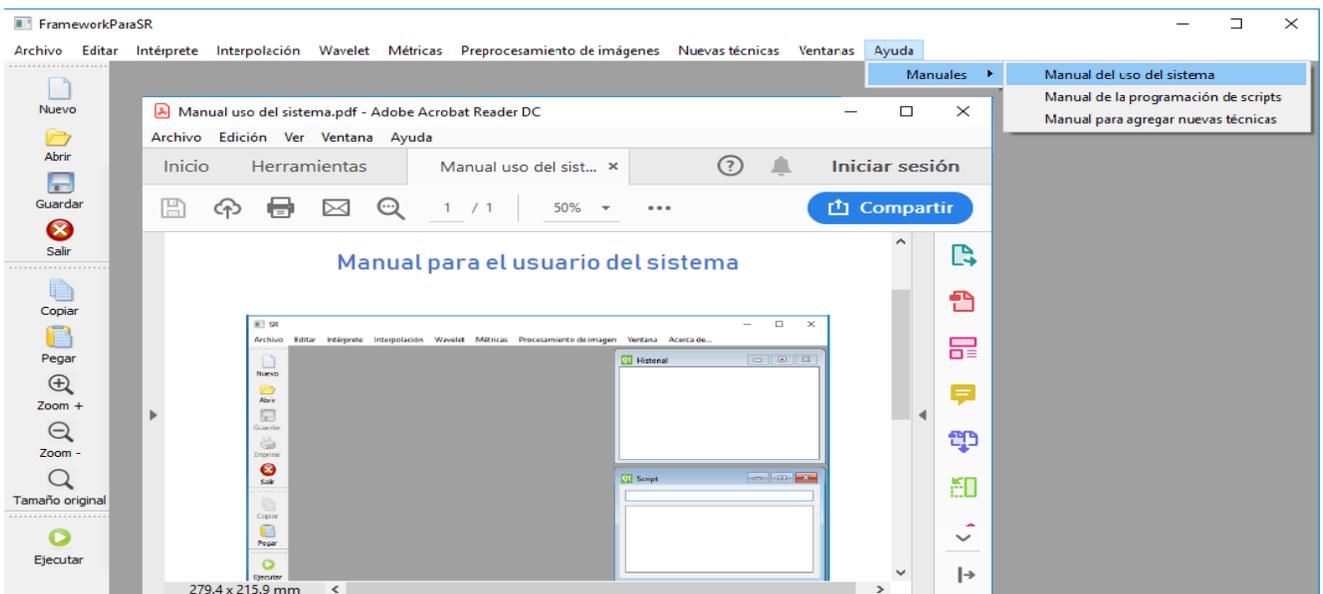


Figura 4.22. Prueba 07: Menú de ayuda, manual del uso del sistema.

4.3.8 Prueba 08

Nombre: Ventana del historial.

Objetivo: Esta prueba permite validar el funcionamiento y actualización de la ventana que va almacenando el historial, tanto de las acciones realizadas desde los menús y la barra de herramientas, así como desde los comandos ejecutados mediante el intérprete.

Resultados: La ventana de historial se va actualizando con las indicaciones de las acciones realizadas tanto desde los menús del *framework* así como desde la ejecución del intérprete.

La Figura 4.23 ejemplifica la ventana de historial con las técnicas aplicadas a varias imágenes.

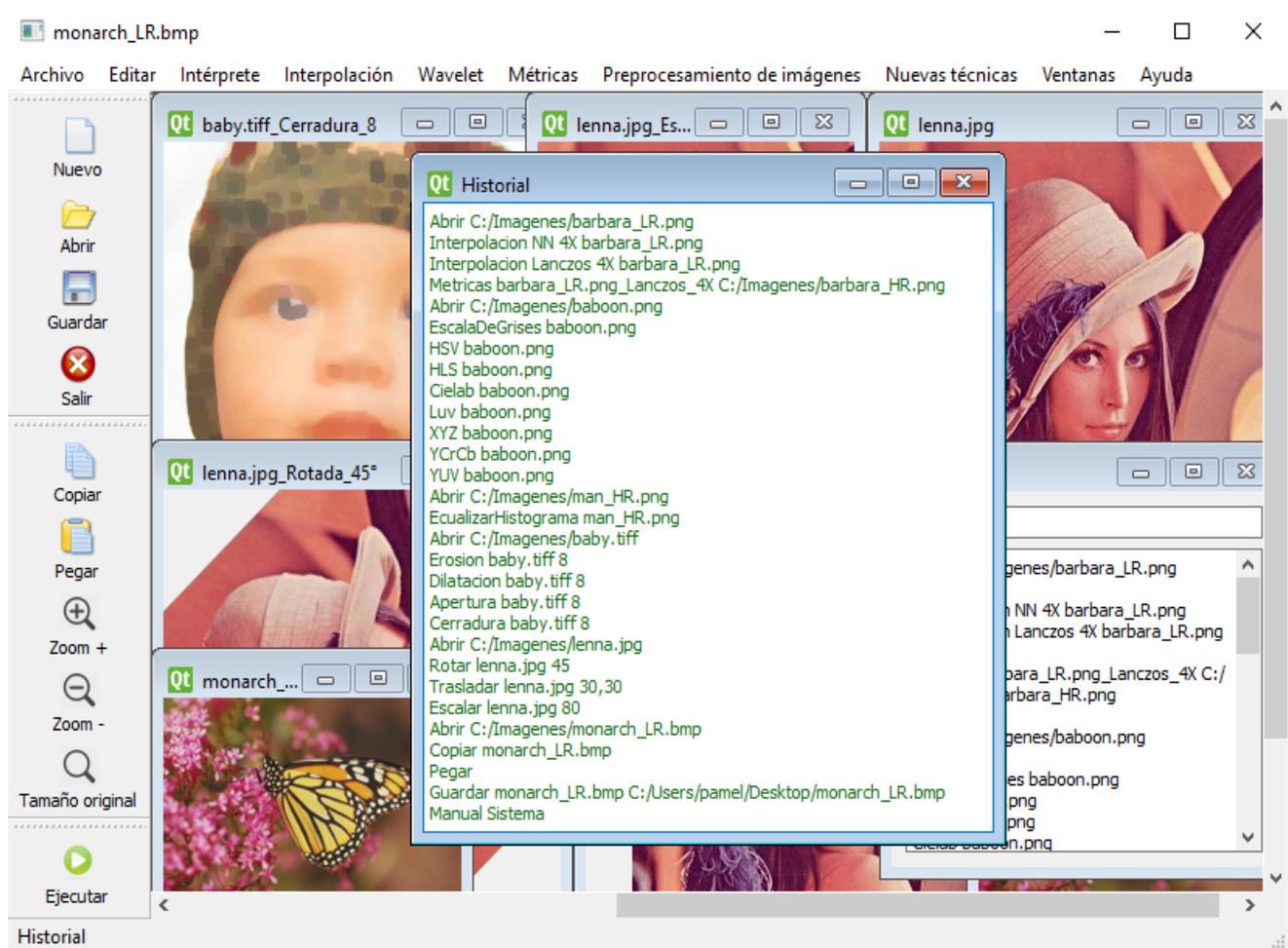


Figura 4.23. Prueba 08: Ventana de historial.

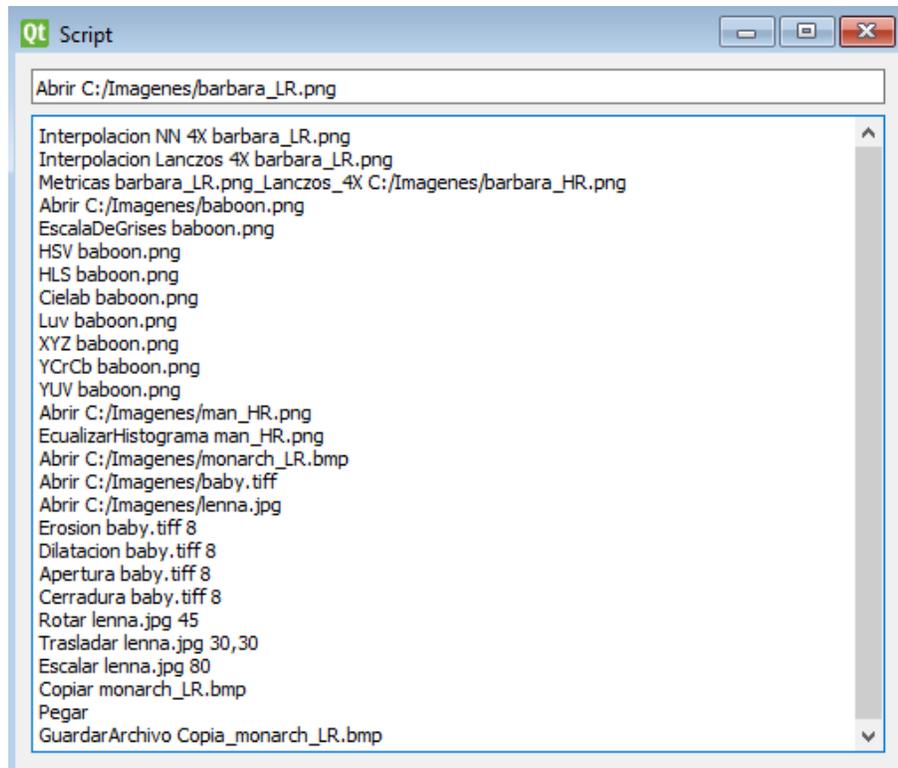
4.3.9 Prueba 09

Nombre: Funcionalidad del intérprete.

Objetivo: Validar que el *framework* cuenta con una ventana para la ejecución de los *scripts*. Y que comandos sean validados mediante el intérprete desarrollado a través de los analizadores léxico y sintáctico.

Secuencia: Ingresar el código dentro de la ventana del script, si el código es colocado en el primer recuadro entonces se debe hacer *clik* sobre la tecla *Enter*, si el código se coloca en el segundo recuadro es necesario hacer *clik* en el icono de *Ejecutar* el cual se localiza en la barra de herramientas, o bien hacer *clik* en el menú *Intérprete* y a continuación *clik* en *Ejecutar*.

Resultados: Los comandos ingresados serán analizados por el intérprete y de ser válidos se realizarán las acciones asociadas a cada comando, para algunos errores se emitirá un mensaje en una subventana. La Figura 4.24 muestra un ejemplo de algunos comandos válidos para el *script* y las acciones que se realizan mediante la ejecución de los mismos.



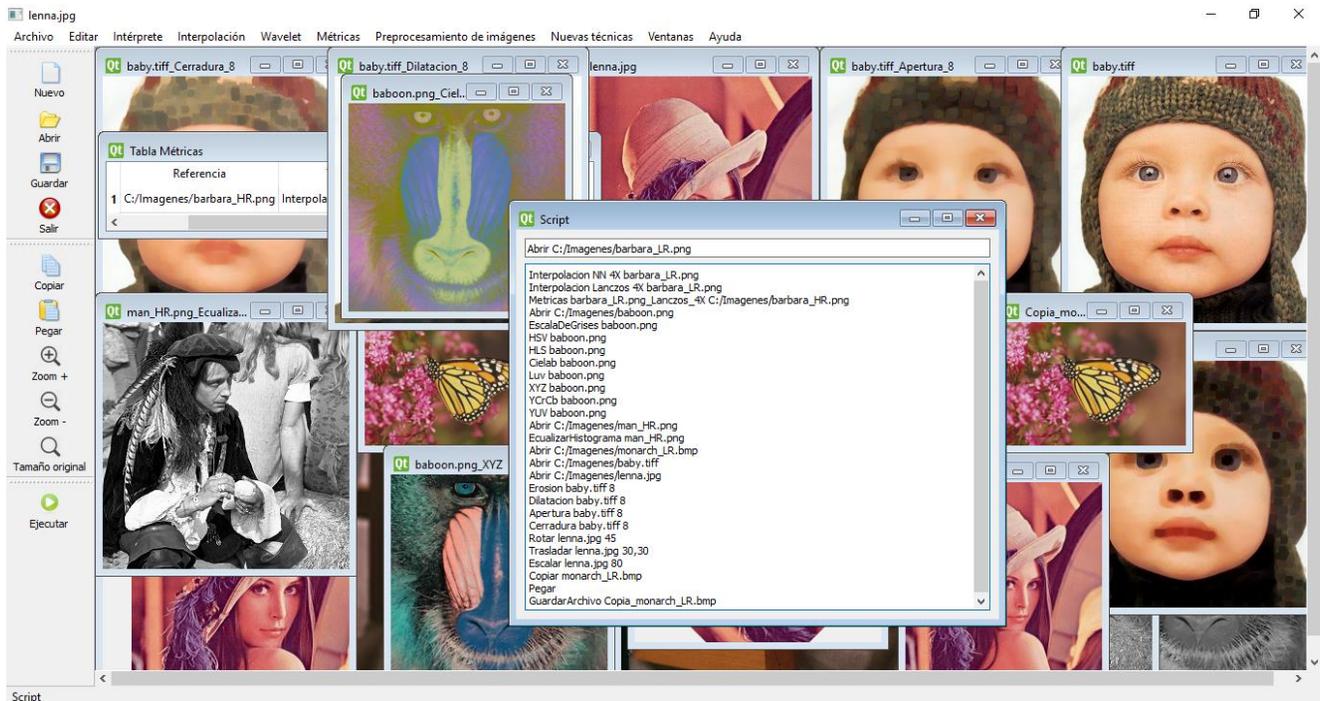


Figura 4.24. Prueba 09: Funcionalidad del intérprete.

4.3.10 Prueba 10

Nombre: Lista de subventanas.

Objetivo: Permite validar el menú *Ventanas*, el cual consta de los siguientes submenús, *Cerrar* el cual cierra la subventana que tiene el foco, *Cerrar todas* cierra todas las subventanas, *Mosaico* ordena las subventanas en forma de mosaico, *Cascada* ordena las subventanas en forma de cascada, *Siguiente* y *Anterior* direccionan a la subventana que tuvo el foco antes o después de la ventana activa, y la lista de ventanas se actualiza conforme se van creando nuevas subventanas,

Secuencia: Para cerrar una ventana, seleccionar el menú *Ventanas* y hacer *clic* en el submenú *Cerrar*.
Para cerrar todas las ventanas, seleccionar el menú *Ventanas* y hacer *clic* en el submenú *Cerrar todas*.
Para ordenar las ventanas en mosaico, seleccionar el menú *Ventanas* y hacer *clic* en el submenú *Mosaico*.
Para ordenar las ventanas en cascada, seleccionar el menú *Ventanas* y hacer *clic* en el submenú *Cascada*.
Para seleccionar la ventana siguiente, seleccionar el menú *Ventanas* y hacer *clic* en el submenú *Siguiente*.
Para seleccionar la ventana anterior, seleccionar el menú *Ventanas* y hacer *clic* en el submenú *Anterior*.
Para seleccionar una ventana en específico o ver la lista de ventanas abiertas, seleccionar el menú *Ventanas* y seleccionar el nombre de la ventana en la lista de las subventanas.

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García.

Resultados: Al seleccionar *Cerrar*, se cierra la subventana que tiene el foco.

Al seleccionar *Cerrar todas*, se cierran todas las subventanas.

Al seleccionar *Mosaico*, se ordenan todas las subventanas en forma de mosaico.

Al seleccionar *Cascada*, se ordenan las subventanas en forma de cascada.

Al seleccionar *Siguiente* o *Anterior*, direcciona a la subventana que tuvo el foco antes o después de la ventana activa.

La lista de ventanas se actualiza conforme se van creando nuevas subventanas y al hacer *clic* en el nombre de alguna ventana se coloca el foco en dicha ventana.

La Figura 4.25 ejemplifica el resultado de seleccionar la opción de ordenamiento de las ventanas en forma de *Mosaico*.

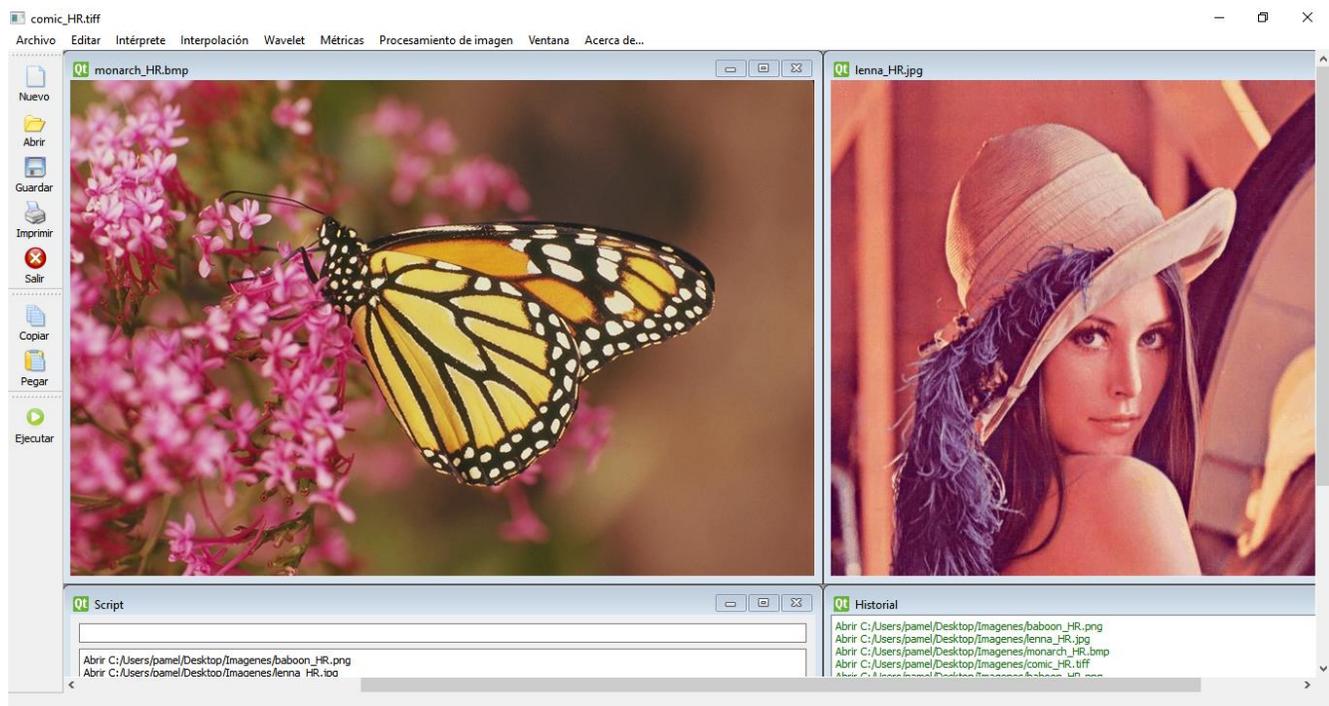


Figura 4.25. Prueba 10: Mosaico.

La Figura 4.26 ejemplifica el resultado de seleccionar la opción de ordenamiento de las ventanas en forma de *Cascada*. Y la Figura 4.27 muestra la lista de subventanas que se crea dentro del menú *Ventanas*.

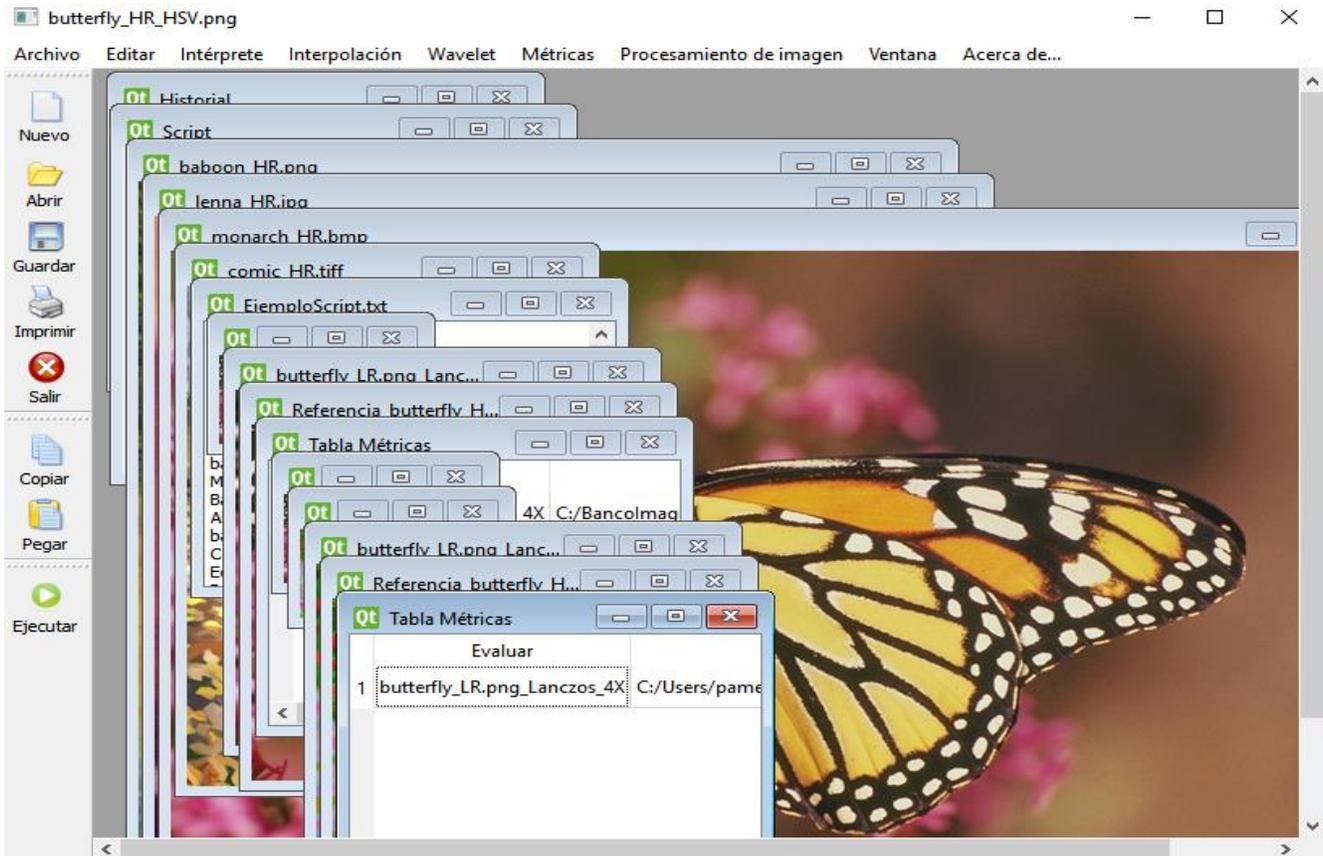


Figura 4.26. Prueba 10: Cascada.

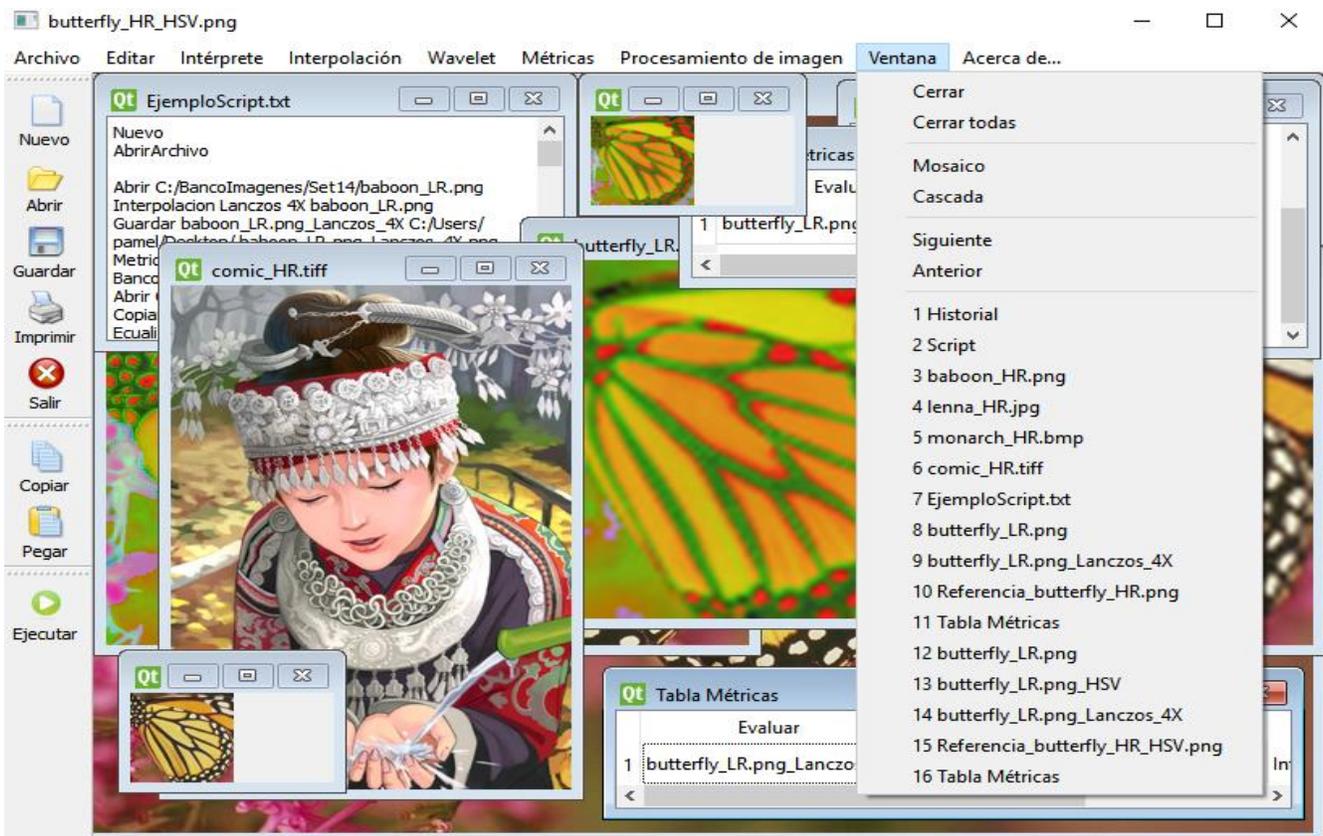


Figura 4.27. Prueba 10: Lista de subventanas.

4.4 Compromisos alcanzados y otros logros

De acuerdo con la información mostrada en este capítulo se observa que cada uno de los objetivos alcances y limitaciones planteados en el Capítulo I, fueron transformados en diferentes requerimientos del sistema; y una vez implementado cada requerimiento se realizaron las pruebas para validar el correcto funcionamiento de los mismos, la Tabla 4.2 muestra la relación de cada requerimiento con sus correspondientes pruebas realizadas.

Tabla 4.2. Requerimientos y pruebas.

Requerimiento Prueba	Interfaz de usuario	Interfaz de Múltiples Documentos (MDI)	Intérprete (Script)	Lectura y escritura (formatos)	Algoritmos clásicos de SR	Métricas para evaluar	Agregar nuevas técnicas	Funcionalidades adicionales
Elementos de la interfaz de usuario	✓	✓	✓					
Menús y submenús de la interfaz	✓	✓	✓	✓	✓	✓	✓	✓
Lectura y escritura de diversos formatos	✓	✓	✓	✓	✓	✓	✓	✓
Algoritmos clásicos de SR				✓	✓			✓
Métricas para evaluar las imágenes				✓	✓	✓		✓
Agregar nuevas técnicas	✓			✓			✓	
Funcionalidades adicionales	✓	✓		✓	✓	✓		✓
Ventana del historial	✓	✓	✓	✓	✓	✓		✓
Funcionalidad del intérprete			✓	✓	✓	✓		✓
Lista de subventanas	✓	✓						

Capítulo 5 . Conclusiones

5.1 Conclusiones

De acuerdo con la información recabada en este documento se observa que se cumplió con los objetivos y alcances establecidos en el Capítulo 1 de esta tesis, los cuales permitieron el diseño e implementación del *framework* para la manipulación de imágenes con el enfoque de Súper Resolución.

Para llevar a cabo la elaboración del *framework* se realizó el desarrollo en lenguaje *C/C++* mediante el *IDE* de *Eclipse* y la biblioteca de desarrollo *OpenCV*, la interfaz de usuario se llevó a cabo mediante la herramienta *QT Creator*, y el intérprete fue implementado a través de *Flex* y *Bison*. Para generar las pruebas de la funcionalidad del *framework* se seleccionaron y descargaron cinco bancos de imágenes: *Set5*, *Set14*, *BSD100*, *Urban100* y *Sun-Hays80*, en formato *PNG* con diferentes dimensiones en alta resolución, cada una con su correspondiente imagen de baja resolución.

El *framework* facilita al usuario la ejecución de cinco algoritmos clásicos de Súper Resolución: *interpolación del vecino más cercano*, *bilineal*, *bicúbica* y *Lanczos*, así como *wavelet haar*. Permite realizar funciones de lectura y escritura de los formatos gráficos: *BMP*, *JPG*, *PNG*, *TIFF*, así como de archivos *.txt* en los cuales se pueden almacenar comandos para después ser ejecutados por el intérprete. Así como la evaluación de las imágenes mediante cuatro métricas *MSE*, *RMSE*, *PSNR* y *SSIM*. Y agregar nuevas técnicas como submenús dentro del *framework*. Un aporte importante es que el *framework* cuenta con un intérprete el cual da flexibilidad y posibilidades de programar tareas.

Adicionalmente, para dar evidencia del potencial del *framework* al extender la funcionalidad con algoritmos de Visión Artificial como *Transformaciones de color* (*Escala de grises*, *HSV*, *HSL*, *Cielab*, *Luv*, *XYZ*, *YCrCb* o *YUV*), *Ecualización del histograma*, *Operaciones morfológicas* (*Erosión*, *Dilatación*, *Apertura* o *Cerradura*) y *Transformaciones geométricas* (*Rotación*, *Traslación* o *Escala*). También cuenta con tres manuales de ayuda, *Manual del uso del sistema*, *Manual de la programación de scripts* y *Manual para agregar nuevas técnicas*.

5.2 Trabajos futuros

Durante el proceso de la investigación y el desarrollo del *framework* se detectaron algunos puntos de interés que se enlistan a continuación, los cuales se pueden tomar en cuenta para trabajos futuros:

- Incrementar la gramática del intérprete, para que se puedan utilizar ciclos, de forma que permita trabajar con todo un banco de imágenes de forma automática así como aplicar técnicas al conjunto de imágenes y obtener los resultados de las métricas en una única tabla.
- Implementar, mediante *plugins*, la agregación de nuevas técnicas, para aumentar la funcionalidad del *framework* y que las técnicas agregadas sean compatibles con el funcionamiento de las técnicas ya establecidas en el sistema.
- Extender la funcionalidad del intérprete para lograr obtener un lenguaje de programación de alto nivel.
- Implementar la funcionalidad para obtener reportes que permitan analizar los resultados obtenidos de las imágenes procesadas de acuerdo con las técnicas seleccionadas y las métricas aplicadas.
- Aumentar la cantidad de algoritmos correspondientes a técnicas de Súper Resolución.
- Implementar la funcionalidad para exportar y graficar los datos proporcionados por la métrica de evaluación.

Referencias

Aaby, A. (2003). *Compiler Construction using Flex and Bison*. Retrieved from http://webdiis.unizar.es/asignaturas/LGA/material_2004_2005/Intro_Flex_Bison.pdf

Agustsson, E., & Timofte, R. (2017). NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2017–July*, 1122–1131. <https://doi.org/10.1109/CVPRW.2017.150>

Álvarez, V., Ponomaryov, V., Shkvarko, Y., & Reyes, R. (2017). Image Super-Resolution via Block Extraction and Sparse Representation. *IEEE Latin America Transactions*, 15(10), 1977–1982. <https://doi.org/10.1109/TLA.2017.8071243>

Béjar, R. (2005). *Introducción a Flex y Bison*. Retrieved from http://webdiis.unizar.es/asignaturas/LGA/material_2004_2005/Intro_Flex_Bison.pdf

Bhatt, U., & Singh, A. (2016). Performance Analysis of Wavelet Filter Bank for an Image Super Resolution Algorithm. *2016 1st India International Conference on Information Processing (IICIP)*, 3–6. <https://doi.org/10.1109/IICIP.2016.7975346>

Blanes, J., & Gorricho, J. (2015). Técnicas de Evaluación de la Calidad de la Imagen. Tendencias y métricas basadas en Bordes, 12. Retrieved from <https://upcommons.upc.edu/bitstream/handle/2117/9805/Image.pdf?sequence=3&isAllowed=y>

Castro, E., Nakano, M., Sánchez, G., & Pérez, H. (2017). Improvement of Image Super-resolution Algorithms using Iterative Back Projection. *IEEE Latin America Transactions*, 15(11), 2214–2219. <https://doi.org/10.1109/TLA.2017.8070429>

Cheng, S. (2019). Demostración de Súper Resolución. *MATLAB Central File Exchange*. Retrieved from <https://www.mathworks.com/matlabcentral/fileexchange/30488-superresolution-demo>

Dong, C., Loy, C. C., & He, K. (2016). Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2), 295–307. Retrieved from <https://ieeexplore.ieee.org/document/7115171>

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García.

Donnelly, C., & Stallman, R. (2019). *Bison, versión 3.4*. Retrieved from <https://www.gnu.org/software/bison/manual/bison.pdf>

Dumic, E., Grgic, S., & Grgic, M. (2007). The Use of Wavelets in Image Interpolation : Possibilities and Limitations. *RADIOENGINEERING*, 16(4), 101–109. Retrieved from https://www.researchgate.net/publication/26511783_The_Use_of_Wavelets_in_Image_Interpolation_Possibilities_and_Limitations

Fernandez, R., Latorre P. & Pla, F. (2016) Super-resolution Toolbox. Single-Image Super-Resolution In Remote Sensing: A Practical Overview. *International Journal of Remote Sensing*. Retrieved from <https://www.vision.uji.es/srtoolbox/index.html>

Galindo, J., & Camps, J. (2008). Diseño e implementación de un marco de trabajo (framework) de presentación para aplicaciones JEE. *Universitat Oberta de Catalunya*, 173. Retrieved from <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/876/1/00765tfc.pdf>

Gao, Z., & Zheng, Y. F. (2007). Quality Constrained Compression Using DWT Based Image Quality Metric, 1–25. Retrieved from <http://www2.ece.ohio-state.edu/~zheng/publications/Quality-constrained-compression.pdf>

Gómez, R., & Salamanca, J. (2012). *Intérprete para un Lenguaje de Programación Orientado a Objetos, con mecanismos de optimización y modificación dinámica de código*. (Tesis de licenciatura). Pontificia Universidad Católica del Perú, Facultad de Ciencias e Ingeniería. Retrieved from <http://tesis.pucp.edu.pe/repositorio/handle/123456789/1503>

Gomeztagle, F. (2010). *Algoritmos para Superresolución en Imágenes y Video* (Tesis doctoral). Instituto Politecnico Nacional, Ingeniería Mecánica y Eléctrica. Retrieved from <https://tesis.ipn.mx/bitstream/handle/123456789/9448/53.pdf?sequence=1&isAllowed=y>

Gonzalez, R., & Woods, R. (2002). *Digital Image Processing (2nd Edition)*. New Jersey. Retrieved from http://web.ipac.caltech.edu/staff/fmasci/home/astro_refs/Digital_Image_Processing_2ndEd.pdf

Gutiérrez, D. (2010). *Frameworks y Componentes* (Documentos de trabajo). *Universidad de los Andes Venezuela*. Mérida. Retrieved from http://www.codecompiling.net/files/slides/IS_clase_10_frameworks_componentes.pdf

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García.

Image Processing Toolbox. (2019). Perform image processing, visualization, and analysis. *The MathWorks, Inc.* Retrieved from https://www.mathworks.com/products/image.html?s_tid=FX_PR_info

Levine, J. (2009). *Flex & Bison: Text Processing Tools*. Sebastopol, CA: O'Reilly Media. Retrieved from http://web.iitd.ac.in/~sumeet/flex__bison.pdf

Luengo, M. (2002). *Desarrollo y evaluación de técnicas de construcción de Procesadores de Lenguaje para Máquinas Abstractas Orientadas a Objetos* (Tesis doctoral). Universidad de Oviedo, Depto. de Informática. Retrieved from <http://di002.edv.uniovi.es/~cueva/investigacion/tesis/Candi.pdf>

Maiseli, B. (2019). Modelo de superresolución multitráma regularizado. *MATLAB Central File Exchange*. Retrieved from <https://www.mathworks.com/matlabcentral/fileexchange/49143-regularized-multiframe-super-resolution-model>

MATLAB. (2019). *The MathWorks, Inc.* Retrieved from https://www.mathworks.com/products/matlab.html?s_tid=FX_PR_info

Mejía, J. (2005). *Apuntes de Procesamiento Digital de Imágenes* (Documentos de trabajo). *Universidad Autónoma de San Luis Potosí*. San Luis Potosí, México. Retrieved from <https://vdocuments.site/36581524-apuntes-de-procesamiento-digital-de-imagenes-1-borrador-jose-ramon-mejia-vilet.html>

Morera, L., Pinto, R., Ochoa, H., & Vergara, O. (2018). Auto-regularized Gradients of Adaptive Interpolation for MRI Super-Resolution. *Journal of Signal Processing Systems*, 14. <https://doi.org/10.1007/s11265-018-1408-1>

Nakamura, J. (2006). *Image Sensors and Signal Processing for Digital Still Cameras*. Group, Taylor & Francis. Retrieved from [http://last.hit.bme.hu/download/firtha/video/Sensors/Junichi Nakamura Image sensors and signal processing for digital still cameras 2006.pdf](http://last.hit.bme.hu/download/firtha/video/Sensors/Junichi%20Nakamura%20Image%20sensors%20and%20signal%20processing%20for%20digital%20still%20cameras%202006.pdf)

OpenCV. (2018). OpenCV 2.4.13.7 documentation. *Geometric Image Transformations*. Retrieved from https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html

Parallel Super Resolution (2013). Retrieved from <https://sites.google.com/site/piotrwendykier/software/parallelsuperresolution>

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García.

Park, S. C., Park, M. K., & Kang, M. G. (2003). Super-resolution image reconstruction: A technical overview. *IEEE Signal Processing Magazine*, 20(3), 21–36. <https://doi.org/10.1109/MSP.2003.1203207>

Paxson, V. (1995). *Flex, versión 2.5*. Retrieved from <http://es.tldp.org/Manuales-LuCAS/FLEX/flex-es-2.5.pdf>

Platero, C. (2009). *Introducción a la Visión Artificial* (Documentos de trabajo). Universidad Politécnica de Madrid, Departamento de Electrónica, Automática e Informática Industrial, Madrid, España. Retrieved from http://www.elai.upm.es/webantigua/spain/Asignaturas/MIP_VisionArtificial/ApuntesVA/cap1IntroVA.pdf

Pérez, J. (2018). *Implementación de algoritmos de Súper Resolución, en hardware dedicado, para imágenes de resonancia magnética* (Tesis de maestría). Centro Nacional de Investigación y Desarrollo Tecnológico, Departamento de Ciencias Computacionales. Cuernavaca, México.

Rodríguez, D. (2012). *Diseño e implementación de un Framework de Presentación* (Tesis de licenciatura). Universitat Oberta de Catalunya (UOC), Estudios de Informática y Multimedia. Retrieved from <http://openaccess.uoc.edu/webapps/bitstream/10609/18968/9/drodriguezsiTFC0113memoria.pdf>

Sánchez, A. (2008). *Super - Resolution : Multi - Frame Registration and Interpolation Using Optimal Projections on Functional Spaces* (Tesis doctoral). Universidad Nacional de Educación a Distancia (España). Escuela Técnica Superior de Ingeniería Informática. Retrieved from <http://e-spacio.uned.es/fez/view/tesisuned:IngInf-Asanchez>

Sánchez, L. (2018). *Desarrollo e implementación de algoritmos de Súper Resolución empleando Wavelets* (Tesis de maestría (en proceso)). Centro Nacional de Investigación y Desarrollo Tecnológico, Departamento de Ciencias Computacionales. Cuernavaca, México.

Sharma, M., Chaudhury, S., & Lall, B. (2017). Deep learning based frameworks for image super-resolution and noise-resilient super-resolution. *2017 International Joint Conference on Neural Networks (IJCNN), 2017–May*, 744–751. <https://doi.org/10.1109/IJCNN.2017.7965926>

Signal Processing Toolbox. (2019). *The MathWorks, Inc.* Retrieved from https://www.mathworks.com/products/signal.html?s_tid=FX_PR_info

Desarrollo de un *FrameWork* para la experimentación con algoritmos de Súper Resolución. Martha Pamela Cejudo García.

Souverville, S. (2016). *Diseño de un algoritmo de súper resolución en imágenes digitales utilizando la teoría de lógica difusa* (Tesis de maestría). Instituto Politécnico Nacional. Retrieved from https://tesis.ipn.mx/bitstream/handle/123456789/23127/Tesis_Samuel_Souverville_Orozco_A140423.pdf?sequence=1&isAllowed=y

Tinoco, C. (2014). *Implementación de un algoritmo basado en el dominio de las frecuencias y procesamiento en paralelo para la generación de una imagen médica en super resolución* (Tesis de licenciatura). Universidad Nacional de Ingeniería, Facultad de Ingeniería Eléctrica y Electrónica. Retrieved from <http://cybertesis.uni.edu.pe/handle/uni/2543>

Villena S., Vega M., Babacan D., Molina R. & Katsaggelos A. (2015). *Super-resolution Software* Retrieved from <https://decsai.ugr.es/pi/superresolution/software.html>

Wang, X., Dong, C., & Loy, C. C. (2018). Recovering Realistic Texture in Image Super-resolution by Deep Spatial Feature Transform. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 10. Retrieved from <https://arxiv.org/abs/1804.02815>

Yang, M., Sorkine-hornung, A., Sorkine-hornung, O., Huang, T. S., Lin, L., Zhang, K., ... Wang, Y. (2018). NTIRE 2018 Challenge on Single Image Super-Resolution : Methods and Results. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Retrieved from http://openaccess.thecvf.com/content_cvpr_2018_workshops/w13/html/Timofte_NTIRE_2018_Challenge_CVPR_2018_paper.html

Yuan, Y., Liu, S., Zhang, J., Zhang, Y., Dong, C., & Lin, L. (2018). Unsupervised Image Super-Resolution using Cycle-in-Cycle Generative Adversarial Networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 814–823. Retrieved from <https://arxiv.org/abs/1809.00437>

Zhang, Z., & Sze, V. (2017). FAST : A Framework to Accelerate Super-Resolution Processing on Compressed Videos. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 1015–1024. <https://doi.org/10.1109/CVPRW.2017.138>

Anexos

Anexo A. Intérprete

A.1 Expresiones Regulares

En la Tabla A.1 se muestran las expresiones regulares declaradas en el analizador léxico.

Tabla A.0.1. Expresiones regulares.

Digito	[0-9]
Numero	{Digito}+
Letra	[a-zA-Z]
Palabra	{Letra}+
Disco	[A-Z]{1}
DosPuntos	[:]{1}
Diagonal	[/]{1}
Simbolo	"_""-""%""("""")""&""\$""#""!""°""=""""" ""+""-""
Carpeta	((Letra){Digito){Simbolo})+
Punto	"."+
Palabras	((Letra){Digito){Simbolo})+
Nombre	{Palabras}{Punto}{Palabras}

A.2 Reglas léxicas

La Tabla A.2 contiene las reglas léxicas establecidas en el analizador léxico.

Tabla A.0.2. Reglas léxicas.

"Nuevo"	{return P_Nuevo;}
"Abrir"	{return P_Abrir;}
"Metricas"	{return P_Metricas;}
"Copiar"	{return P_Copiar;}
"Interpolacion" "Wavelet"	{printf("tecnica: %s\n", yytext); return Tecnica;}
"NN" "BL" "BC" "Lanczos" "Haar"	{printf("tipoTecnica: %s\n", yytext); return TipoTecnica;}
" "	{return Blanco;}
{Numero}	{printf("valor: %s\n", yytext); return Valor;}
"X"	{return Constante;}
"Erosion" "Dilatacion" "Apertura" "Cerradura"	{printf("morfolologica: %s\n", yytext); return Morfolologica;}

A.3 Símbolos terminales

A continuación se listan los símbolos terminales declarados en el analizador sintáctico.

%token P_Nuevo	%token Palabra	%token P_Rotar
%token P_Abrir	%token Palabras	%token P_Trasladar
%token P_Metricas	%token Punto	%token Coma
%token P_Copiar	%token Tecnica	%token P_Escalar
%token P_Pegar	%token P_EcualizarHistograma	%token Blanco
%token Dígito	%token Morfologica	%token P_Guardar
%token Letra	%token P_Manual	%token P_Salir
%token Numero	%token Manual	

A.4 Reglas gramaticales

En la Tabla A.3 se muestran las reglas gramaticales declaradas en el analizador sintáctico.

Tabla A.0.3. Reglas gramaticales.

ABRIR :	P_Abrir Blanco Ruta {printf("openFile");};
SR :	Tecnica Blanco TipoTecnica Blanco Valor Constante Blanco NombreArchivo {printf("SR");};
METRICAS :	P_Metricas Blanco NombreArchivo Blanco Ruta {printf("metricas");};
COPIAR :	P_Copiar Blanco NombreArchivo {printf("copy");};
PEGAR :	P_Pegar {printf("paste");};
COLOR :	Color Blanco NombreArchivo {printf("color");};
GUARDAR :	P_Guardar Blanco NombreArchivo Blanco Ruta {printf("save");};
MANUAL :	P_Manual Blanco Manual {printf("manual");};
SALIR :	P_Salir {printf("exit");};

Anexo B. Banco de imágenes

B.1 Set5

Contiene 5 imágenes (*baby, bird, butterfly, head y woman*) en formato PNG con diferentes dimensiones en alta resolución, cada una con su correspondiente imagen de baja resolución. Sitio de descarga: <https://github.com/jbhuang0604/SelfExSR>. La Figura B.1 muestra un conjunto del *dataset*.



Figura B.1. Banco de imágenes Set5.

B.2 Set14

Contiene 14 imágenes (*baboon, barbara, bridge, coastguard, comic, face, flowers, foreman, lenna, man, monarch, pepper, ppt3 y zebra*) en formato PNG con diferentes dimensiones en alta resolución, cada una con su correspondiente imagen de baja resolución. Sitio de descarga: <https://github.com/jbhuang0604/SelfExSR>. La Figura B.2 muestra un conjunto del *dataset*.



Figura B.2. Banco de imágenes Set14.

B.3 BSD100

Contiene 100 imágenes en formato PNG con diferentes dimensiones en alta resolución, cada una con su imagen de baja resolución. Sitio de descarga: <https://github.com/jbhuang0604/SelfExSR>. La Figura B.3 muestra un conjunto del *dataset*.



Figura B.3. Muestra del banco de imágenes BSD100.

B.4 Urban100

Contiene 100 imágenes en formato PNG con diferentes dimensiones en alta resolución, cada una con su imagen de baja resolución. Sitio de descarga: <https://github.com/jbhuang0604/SelfExSR>. La Figura B.4 muestra un conjunto del *dataset*.

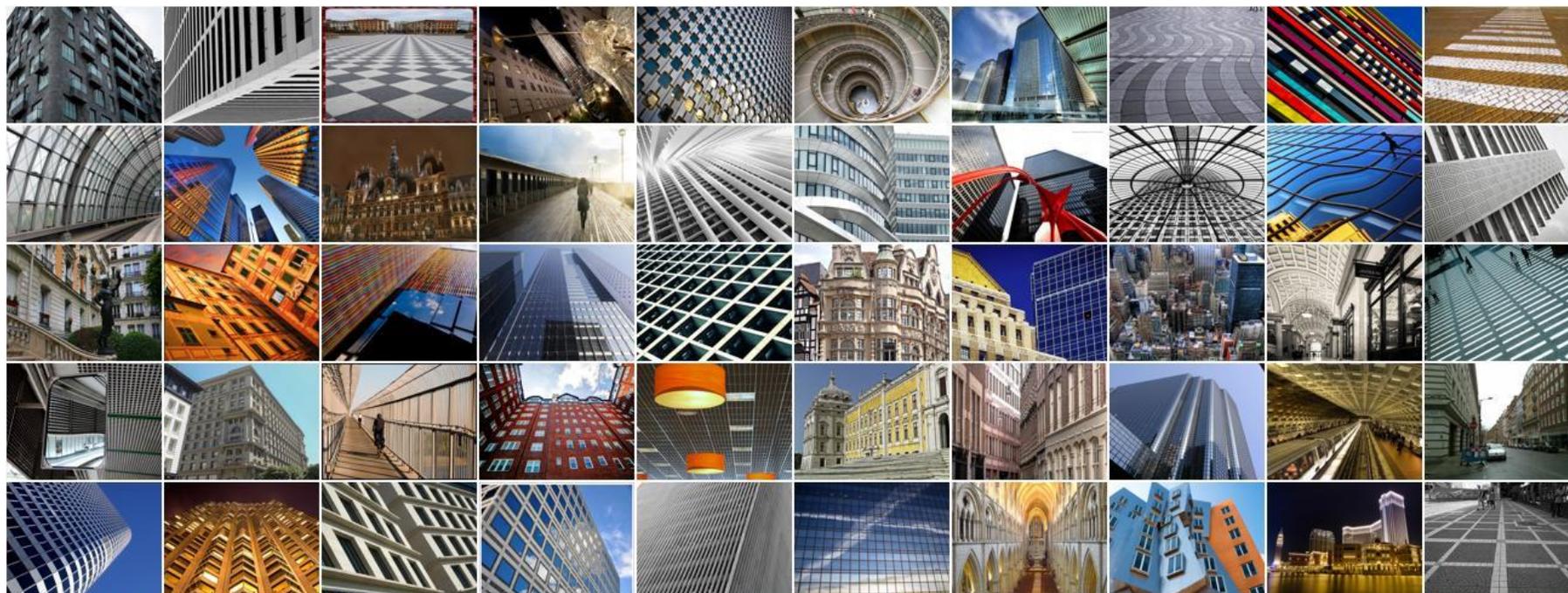


Figura B.4. Muestra del banco de imágenes Urban100.

B.5 Sun-Hays80

Contiene 80 imágenes en formato PNG con diferentes dimensiones en alta resolución, cada una con su imagen de baja resolución. Sitio de descarga: <https://github.com/jbhuang0604/SelfExSR>. La Figura B.5 muestra un conjunto del *dataset*.



Figura B.5. Banco de imágenes Sun-Hays80.



**TECNOLÓGICO
NACIONAL DE MÉXICO**

cenidet[®]
Centro Nacional de Investigación
y Desarrollo Tecnológico