

Centro Nacional de Investigación y Desarrollo Tecnológico

Subdirección Académica

Departamento de Ciencias Computacionales

TESIS DE MAESTRÍA EN CIENCIAS

**Detección de Ruido Impulsivo o Gaussiano en Imágenes
Monocromáticas Mediante Redes Neuronales Artificiales
Pulso-Acopladas**

presentada por

Ing. Estela Ortiz Rangel

como requisito para la obtención del grado de

Maestra en Ciencias de la Computación

Director de tesis

Dr. Manuel Mejía Lavalle

Cuernavaca, Morelos a 08 de diciembre del 2016

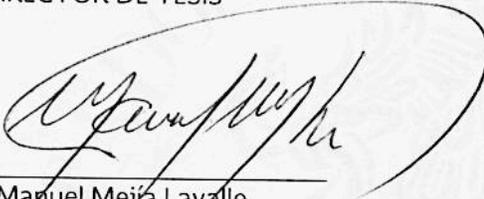
OFICIO No. DCC/296/2016

Asunto: Aceptación de documento de tesis

C. DR. GERARDO V. GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial de la **Ing. Estela Ortiz Rangel**, con número de control M14CE066, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis profesional titulado "**Detección de Ruido Impulsivo o Gaussiano en Imágenes Monocromáticas mediante Redes Neuronales Pulso-Acopladas**" y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS



Dr. Manuel Mejía Lavalle
Doctor en Ciencias
Computacionales
8342472

REVISOR 1



Dr. Dante Mújica Vargas
Doctor en Comunicaciones y
Electrónica
09131756

REVISOR 2



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

C.p. M.T.I. María Elena Gómez Torres - Jefa del Departamento de Servicios Escolares.
Estudiante
Expediente

NACS/Imz

Cuernavaca, Mor., 20 de diciembre de 2016
OFICIO No. SAC/366/2016

Asunto: Autorización de impresión de tesis

ING. ESTELA ORTIZ RANGEL
CANDIDATA AL GRADO DE MAESTRA EN CIENCIAS
DE LA COMPUTACIÓN
PRESENTE

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"Detección de Ruido Impulsivo o Gaussiano en Imágenes Monocromáticas mediante Redes Neuronales Pulso-Acopladas"**, 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.

ATENTAMENTE
"CONOCIMIENTO Y TECNOLOGÍA AL SERVICIO DE MÉXICO"



DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO



SEP TecNM
CENTRO NACIONAL
DE INVESTIGACIÓN
Y DESARROLLO
TECNOLÓGICO
SUBDIRECCIÓN
ACADÉMICA

C.p. M.T.I. María Elena Gómez Torres.- Jefa del Departamento de Servicios Escolares.
Expediente

GVGR/mcr

Dedicatoria

Dedico este nuevo logro a mis padres, en especial a mi madre que me impulsó desde el inicio a plasmar, perseguir y ser constante en todas mis metas. Agradezco a mis dos familias por ser un apoyo y un ejemplo de buenos valores, cariño y fe.

Te dedico este trabajo a ti Jonathan porque me has apoyado con tu buen humor en los buenos y malos tiempos, porque con tu paciencia me hago paciente y en tu cariño he hallado mi mayor motivación. Gracias por estar siempre presente, con tu tiempo, con una palabra, un abrazo, una sonrisa y por tu esfuerzo cotidiano.

Dedico esta tesis a mis compañeros en quienes encontré buenos amigos, porque nos hemos acompañado en este camino y gracias a ustedes disfruté cada momento, especialmente a aquellos que han puesto en pausa el trayecto para que un día logren alcanzar sus metas.

Agradecimiento

En primer lugar agradezco a CONACYT y a CENIDET por haber hecho posible la culminación de este trabajo de investigación. A todos los colaboradores del centro por su valiosísima aportación al desarrollo de cada uno de los investigadores tanto en el aspecto académico como material y cultural.

Un agradecimiento especial al CIC y al IPN por abrir sus puertas a investigadores que como yo, buscan ampliar sus experiencias formativas, fue una grata experiencia regresar al alma mater y especialmente gracias al Dr. Humberto Sossa por compartir su experiencia y saber conmigo.

Doy las gracias a los profesores de CENIDET que contribuyeron a mi formación como Maestra en Ciencias, especialmente al Dr. Manuel Mejía Lavalle por guiar acertadamente el desarrollo de esta tesis, por su tiempo y su dedicación, por incentivarme y enseñarme. Así mismo agradezco a mis revisores el Dr. Raúl Pinto y el Dr. Dante Mújica por sus enseñanzas, paciencia, consejos y críticas.

Deseo dar un agradecimiento especial a los compañeros que aportaron a la realización de esta tesis por haber puesto los cimientos de la investigación: Selene y Bismarck; a Marilú por ser mi compañera en este viaje y a mis compañeros de IA por amenizar el trayecto.

Resumen

En este trabajo se presenta un esquema de comparación entre técnicas tradicionales y Redes Neuronales tipo Pulso-Acopladas para llevar a cabo la detección y el filtrado de ruido impulsivo y gaussiano en imágenes digitales.

El ruido es un elemento no deseado que contamina las imágenes digitales, proviene principalmente de defectos en los dispositivos de captura, transmisión y almacenamiento. La Visión Artificial ha creado modelos de ruido y técnicas de filtrado que permiten minimizar su efecto, en el presente trabajo se tratan los ruidos impulsivo y aditivo gaussiano y se estudia cómo disminuir su efecto mediante las Redes Neuronales Pulso-Acopladas.

Algunas técnicas para contrarrestar el ruido son efectivas, como el filtro de mediana y el filtro promedio, no obstante conforme aumenta el nivel de ruido las técnicas para eliminarlo se hacen más complejas y la preservación de los detalles más difícil, por lo que se requiere de nuevas técnicas de detección y reducción del ruido como los filtros diseñados en el dominio de la frecuencia y los filtros difusos, entre otros.

Las Redes Neuronales tipo Pulso-Acopladas emulan a las neuronas de la corteza visual de los mamíferos, su modelo computacional implica que a cada neurona corresponde un pixel de la imagen, la capa de salida es una matriz binaria de neuronas cuya activación depende de un umbral dinámico que varía con cada iteración. Este tipo de redes se ha utilizado para realizar segmentación de imágenes, formación de descriptores y recientemente en la detección de ruido, principalmente impulsivo y gaussiano.

Por medio de este trabajo se ha llegado al cumplimiento de el objetivo general de implementar y evaluar las Redes Neuronales Pulso-Acopladas para la detección de pixeles ruidosos en las imágenes digitales, así mismo se han alcanzado los objetivos específicos con sus respectivos alcances y limitaciones al implementarse tres variantes de algoritmos de filtrado basados en las Redes Neuronales-Pulso Acopladas y al compararse con tres técnicas de filtrado tradicionales.

Adicionalmente dentro de las aportaciones de la investigación se encuentra la búsqueda de mejoras en los algoritmos de filtrado basadas en Redes Neuronales Pulso-Acopladas, por un lado al variar los parámetros de las redes neuronales para lograr una mejor detección del ruido, y por otro lado al proponer modificaciones en las técnicas complementarias de filtrado.

Abstract

By this written work the thesis of the Master of Science in Artificial Intelligence corresponding to the detection and noise filtering into digital images using traditional techniques and Pulse-Coupled Neural Networks research is completed.

Noise is an unwanted element that contaminates digital images, comes mainly from defects in capture, transmission and storage devices. Machine Vision has created noise models and filtering techniques that minimize its effect, in this paper the impulsive and Gaussian additive noises and how to reduce its effect through Pulse-Coupled Neural Networks are studied.

Some techniques to counteract noise are effective as the median filter and average filter, however with increasing the noise level techniques to eliminate it become more complex and the preservation of the details most difficult, so which it requires new techniques such as filters designed in the frequency domain, fuzzy filters and others.

Pulse-Coupled Neural Networks emulate the neurons of the visual cortex of mammals, their computer model implies that each neuron corresponds to a pixel of the image, the output layer is a binary matrix of neurons whose activation depends on a dynamic threshold that varies with each iteration. This type of network has been used to perform image segmentation, formation of descriptors and recently in detecting noise.

By this work has come to meeting the overall objective of implementing and evaluating Pulse-Coupled Neural Networks to detect noisy pixels in digital images, have achieved the specific objectives with their respective scope and limitations, further into the contributions of the research is seeking improvements in filtering algorithms based on Pulse-Coupled Neural Networks.

Additionally into the main contributions of the investigation are, in first place, the network's parameters modification to produce best noise detection and in second place by improving variations in the filtering algorithms.

Índice general

Índice de figuras	iii
Índice de tablas	viii
Capítulo 1 Introducción	1
1.1 Motivación	1
1.2 Descripción del problema	2
1.3 Objetivo general	4
1.4 Objetivos específicos	4
1.5 Alcances y limitaciones del proyecto	4
1.6 Justificación y beneficios	5
1.7 Metodología de solución	5
1.8 Organización de la tesis	6
Capítulo 2 Estado del Arte	7
2.1 Antecedentes institucionales	7
2.2 Trabajos relacionados	8
2.3 Discusión del estado del arte	13
Capítulo 3 Marco Teórico	16
3.1 Visión Artificial	16
3.2 Procesamiento Digital de Imágenes	16
3.3 Redes Neuronales Pulsantes	23
3.4 Detectores de impulsos	28
3.5 Discusión	29
Capítulo 4 Implementación	30
4.1 Requerimientos del sistema	30
4.2 Casos de uso del sistema	31
4.3 Diseño de interfaz de la plataforma de experimentación	34
4.4 Codificación	35
4.4 Discusión	77
Capítulo 5 Experimentación y Resultados	78
5.1 Detalles y objetivos de la experimentación	78
5.2 Filtrado tradicional y basado en PCNN para tratar ruido impulsivo	79
5.3 Filtrado tradicional y basado en PCNN para tratar ruido gaussiano	87
5.4 Filtrado de ruido impulsivo y gaussiano mezclados	94
Capítulo 6 Conclusión	97
6.1 Objetivos logrados	97

6.2 Productos	99
6.3 Aportaciones	102
6.4 Lecciones aprendidas	102
6.5 Conclusiones.....	103
6.6 Trabajo futuro.....	105
Bibliografía	106

Índice de figuras

Figura 1.1 Aplicación comercial de Visión Artificial [Sabia.tic.udc.es, 2016].....	1
Figura 1.2 Ubicación de la solución del problema.....	3
Figura 1.3 Metodología de solución	6
Figura 2.1 Plataforma de pruebas para segmentación con PCNN [Cárdenas, 2015]..	7
Figura 2.2 Metodología extracción de características con PCNN [Zárate, 2015]	8
Figura 2.3 PSNR para filtros de mediana con Cameraman [Sreenivasulu, 2014]	9
Figura 2.4 Algoritmo II de filtrado mediante PCNN [Ma, 2010]	10
Figura 2.5 Algoritmo III de filtrado mediante PCNN [Ma, 2010]	11
Figura 2.6 a) Imagen contaminada con ruido impulsivo de probabilidad 10% y ruido gaussiano de $\sigma^2=0.001$, resultado del filtro de b) mediana, c) reglas difusas, d) L&A-PCNN, e) método propuesto [Shao-Fa, 2010]	12
Figura 2.7 MAE y PSNR de diferentes tipos de filtrado [Jiang, 2010].....	13
Figura 3.1 Diagrama de bloques mostrando el proceso de Procesamiento Digital de Imágenes [Pajares, 2008]	17
Figura 3.2 Imágenes de prueba derivadas de la imagen Peppers_mon [Pratt, 2001]19	
Figura 3.3 Máscaras de 3 x 3 pixeles, comunes para filtrado pasa bajas [Pratt, 2001]	20
Figura 3.4 Filtrado de mediana de 5x5 y 7x7 para imágenes con ruido gaussiano [Pratt, 2001].....	20
Figura 3.5 Neurona tipo Eckhorn [Lindblad, 2005].....	24
Figura 3.6 Modelo de PCNN [Zhang, 2007]	25
Figura 3.7 Esquema de los algoritmos de filtrado de ruido mediante PCNN	27
Figura 4.1 Diagrama de casos de uso del sistema PCNN	32
Figura 4.2 Interfaz de la plataforma de experimentación	34
Figura 4.3 Sección ampliada de imágenes de Lena con ruido impulsivo de probabilidad de ocurrencia de 0 a 0.9	40
Figura 4.4 Imágenes miniatura de Lena con ruido impulsivo de probabilidad de ocurrencia de 0 a 0.9.....	41
Figura 4.5 Imágenes miniatura de Baboon con ruido impulsivo de probabilidad de ocurrencia de 0 a 0.9.....	41
Figura 4.6 Imágenes miniatura de Peppers con ruido impulsivo de probabilidad de ocurrencia de 0 a 0.9.....	41
Figura 4.7 Imágenes miniatura de Grass con ruido impulsivo de probabilidad de ocurrencia de 0 a 0.9.....	42

Figura 4.8 Imágenes miniatura de Cameraman con ruido impulsivo de probabilidad de ocurrencia de 0 a 0.9.....	42
Figura 4.9 Sección de imágenes con ruido gaussiano de $\mu = 0$ al incrementar la varianza.....	42
Figura 4.10 Imágenes miniatura de Lena con ruido gaussiano de $\mu = 0$ al incrementar la varianza.....	43
Figura 4.11 Imágenes miniatura de Baboon y Peppers con ruido gaussiano de $\mu = 0$ al incrementar la varianza.....	43
Figura 4.12 Sección ampliada de imágenes de Lena con ruido impulsivo procesadas con filtro de mediana de 3x3.....	44
Figura 4.13 Imágenes de Lena con ruido impulsivo de probabilidad p filtradas con el operador de mediana de 3x3.....	44
Figura 4.14 Imágenes de Baboon con ruido impulsivo de probabilidad p filtradas con el operador de mediana de 3x3.....	44
Figura 4.15 Imágenes de Peppers con ruido impulsivo de probabilidad p filtradas con el operador de mediana de 3x3.....	45
Figura 4.16 Imágenes de Cameraman con ruido impulsivo de probabilidad p filtradas con el operador de mediana de 3x3.....	45
Figura 4.17 Imágenes con ruido gaussiano de $\mu = 0$ filtradas con filtro de mediana.....	46
Figura 4.18 Imágenes miniatura de Lena con ruido gaussiano filtradas con filtro de mediana.....	46
Figura 4.19 Imágenes miniatura de Baboon y Peppers con ruido gaussiano filtradas con filtro de mediana.....	47
Figura 4.20 Sección ampliada de imágenes de Lena con ruido impulsivo filtradas con el filtro promedio de 3x3.....	47
Figura 4.21 Imágenes de Lena con ruido impulsivo de probabilidad p filtradas con el filtro promedio de 3x3.....	48
Figura 4.22 Imágenes de Baboon con ruido impulsivo de probabilidad p filtradas con el filtro promedio de 3x3.....	48
Figura 4.23 Imágenes de Peppers con ruido impulsivo de probabilidad p filtradas con el filtro promedio de 3x3.....	48
Figura 4.24 Imágenes de Cameraman con ruido impulsivo de probabilidad p filtradas con el filtro promedio de 3x3.....	49
Figura 4.25 Imágenes con ruido impulsivo de probabilidad p filtradas con el filtro morfológico.....	49
Figura 4.26 Imágenes miniatura con ruido gaussiano filtradas con filtro Wiener.....	50
Figura 4.27 Imágenes miniatura de Baboon y Peppers con ruido gaussiano filtradas con filtro Wiener.....	50

Figura 4.28 Imágenes ampliadas con ruido gaussiano de $\mu = 0$ filtradas con filtro Wiener	51
Figura 4.29 PSNR del filtro de mediana, promedio y morfológico para Lena con ruido impulsivo	51
Figura 4.30 PSNR del filtro de promedio y mediana para la imagen de Baboon con ruido impulsivo	52
Figura 4.31 PSNR del filtro de promedio y mediana para la imagen de Peppers con ruido impulsivo	52
Figura 4.32 PSNR del filtro de promedio y mediana para la imagen de Cameraman con ruido impulsivo.....	53
Figura 4.33 PSNR del filtro de promedio y mediana para la imagen de Grass con ruido impulsivo	53
Figura 4.34 PSNR para la imagen de Lena con ruido gaussiano con distintos filtros	54
Figura 4.35 MAE para imágenes con ruido impulsivo.....	54
Figura 4.36 MAE para imágenes con ruido gaussiano filtradas con Filtro de mediana y Wiener	55
Figura 4.37 MSE para diversas imágenes con ruido impulsivo	55
Figura 4.38 MSE para Lena con ruido gaussiano con el filtro Wiener y el filtro de mediana de 3x3.....	56
Figura 4.39 NMSE para imágenes con ruido impulsivo	56
Figura 4.40 NMSE para Lena con ruido gaussiano filtrada con filtro Wiener y filtro de mediana de 3x3.....	57
Figura 4.41 Tiempo de procesamiento del filtro de mediana, promedio y morfológico	57
Figura 4.42 Salidas binarias de la red PCNN para 10 iteraciones para una imagen sin ruido	58
Figura 4.43 Secciones ampliadas de Lena con ruido impulsivo procesada con la red PCNN	59
Figura 4.44 Lena con ruido gaussiano $\sigma^2 = 0.1$ procesada mediante la red PCNN ..	60
Figura 4.45 Lena con ruido gaussiano $\sigma^2 = 0.3$ procesada mediante la red PCNN ..	61
Figura 4.46 Imagen con 10% de ruido impulsivo procesada con la red ICM.....	62
Figura 4.47 Imágenes filtradas selectivamente por el operador de mediana utilizando la salida de la red ICM.....	63
Figura 4.48 Imágenes de 0 a 50% de ruido impulsivo filtradas selectivamente utilizando la salida de la red ICM	64
Figura 4.49 Imágenes de 60 a 90% de ruido impulsivo filtradas selectivamente utilizando la salida de la red ICM	65
Figura 4.50 Toma de muestra para experimentos con ICM.....	66
Figura 4.51 Valores de ejemplo de la ICM para la imagen de muestra	66

Figura 4.52 Valores de gris de la muestra examinada.....	66
Figura 4.53 Patrones de activación con ICM de la muestra examinada	67
Figura 4.54 Valores de gris de la muestra con ruido pimienta	67
Figura 4.55 Patrones de activación con ICM de la muestra con ruido pimienta	68
Figura 4.56 Valores de gris de la muestra con ruido sal	68
Figura 4.57 Patrones de activación con ICM de la muestra con ruido sal	69
Figura 4.58 Imagen con 10% de ruido impulsivo procesada con la red SCM.....	70
Figura 4.59 Imágenes filtradas selectivamente por el operador de mediana utilizando la salida de la red SCM	71
Figura 4.60 Imágenes con ruido impulsivo con probabilidad de ocurrencia de 0 a 0.4 filtradas selectivamente con el operador de mediana utilizando la salida de la red SCM	72
Figura 4.61 Imágenes con ruido impulsivo con probabilidad de ocurrencia de 0.5 a 0.9 filtradas selectivamente con el operador de mediana utilizando la salida de la red SCM	73
Figura 4.62 Relación entre iteraciones y recuperación de la imagen para ICM y SCM	74
Figura 4.63 Salidas binarias de la red PCNNNI para una imagen con ruido gaussiano	75
Figura 4.64 Imagen de a) Peppers sin ruido, b) ampliada, c) con ruido gaussiano de $\sigma^2 = 0.01$	75
Figura 4.65 Imágenes de salida de la red PCNNNI, iteraciones 1 a 10 para imagen sin ruido.....	76
Figura 4.66 Imágenes de salida de la red PCNNNI, iteraciones 1 a 10 para imagen con ruido.....	76
Figura 5.1 a) Imagen con ruido 10%, filtrada con b) filtro promedio, c) filtro de mediana, d) filtro morfológico,	79
Figura 5.2 Relación entre probabilidad de ruido y PSNR.....	80
Figura 5.3 Relación entre probabilidad de ruido y MAE.....	80
Figura 5.4 Relación entre probabilidad de ruido y NMSE	81
Figura 5.5 Relación entre probabilidad de ruido impulsivo y tiempo de procesamiento	81
Figura 5.6 Respuesta de los filtros al escalado de la imagen	82
Figura 5.7 Respuesta de los filtros a la rotación de la imagen.....	83
Figura 5.8 Ampliación de a) imagen ruido gaussiano de media= 0 y $\sigma^2 = 0.01$, b) filtrada con PCNNNI,	87
Figura 5.9 Relación entre la varianza y el PSNR para filtros de ruido gaussiano.....	88
Figura 5.10 Relación entre la varianza y el MAE	89
Figura 5.11 Relación entre la varianza y el NMSE.....	89

Figura 5.12 Relación entre la varianza del ruido gaussiano y el tiempo de procesamiento.....	90
Figura 5.13 Relación entre el coeficiente de atenuación de feeding y el PSNR.....	90
Figura 5.14 Relación entre el coeficiente de amplificación de feeding y el PSNR.....	91
Figura 5.15 Relación entre las constantes de umbral y el PSNR	91
Figura 5.16 Relación entre las constantes de umbral y el PSNR	92
Figura 5.17 a) Ampliación de imagen con 10% ruido impulsivo + ruido gaussiano de $\mu = 0$, $\sigma^2 = 0.01$, filtrada con b) mediana, c) promedio, d) filtro Wiener, e) SCM, f) ICM y g) PCNNNI.....	94
Figura 5.18 a) Ampliación de imagen con ruido gaussiano con $\mu = 0$, $\sigma^2 = 0.01$ + 10% ruido impulsivo, filtrada con b) mediana, c) promedio, d) filtro Wiener, e) SCM, f) ICM y g) PCNNNI.....	95
Figura 5.19 Relación PSNR con ruido mezclado cuando aumenta la varianza.....	95
Figura 5.20 Relación PSNR con ruido mezclado cuando aumenta la probabilidad...	96
Figura 6.1 Publicación realizada para ICMEAE 2015	100
Figura 6.2 Publicación realizada para MCPR2016	100
Figura 6.3 Publicación realizada para COMIA 2016	101
Figura 6.4 Publicación realizada para la revista CyS.....	101

Índice de tablas

Tabla 2.1 Síntesis de los trabajos relacionados.....	14
Tabla 4.1 Algoritmo de procesamiento de una imagen con la red PCNN.....	35
Tabla 4.2 Algoritmo de detección de ruido impulsivo mediante la red ICM	35
Tabla 4.3 Algoritmo de detección de ruido impulsivo mediante la red SCM.....	36
Tabla 4.4 Algoritmo de eliminación de ruido detectado mediante el operador de mediana.....	36
Tabla 4.5 Algoritmo de procesamiento de una imagen con la red PCNNNI	36
Tabla 4.6 Plan de pruebas del sistema prueba 4.5.1	37
Tabla 4.7 Plan de pruebas del sistema pruebas 4.5.2 a 4.5.6	38
Tabla 4.8 Plan de pruebas del sistema pruebas 4.5.7 a 4.5.12	39
Tabla 4.9 Plan de pruebas del sistema pruebas 4.5.13 a 4.5.15	40
Tabla 5.1 Capacidad de recuperación PSNR [dB] de los filtros de ruido impulsivo...	82
Tabla 5.2 Porcentaje de detección del ruido impulsivo de la red ICM	83
Tabla 5.3 Comparación en PSNR (dB) de filtrado entre detectores de impulsos	84
Tabla 5.4 Resumen de características de filtrado de ruido impulsivo con PCNN y filtros tradicionales.....	86
Tabla 5.5 Capacidad de recuperación en NMSE y PSNR de los filtros de ruido gaussiano	87
Tabla 5.6 Características de filtrado de ruido gaussiano con PCNN y filtros tradicionales	94
Tabla 6.1 Actividades realizadas para el logro de los objetivos.....	97
Tabla 6.2 Actividades realizadas para el logro de los alcances.....	98
Tabla 6.3 Descripción de la base de datos generada	99
Tabla 6.4 Ventajas y desventajas del filtrado de ruido impulsivo y gaussiano mediante filtros tradicionales.....	103
Tabla 6.5 Ventajas y desventajas del filtrado de ruido impulsivo y gaussiano mediante PCNN	104

Simbología

Símbolo	Nombre
$S(i, j), S_{ij}$	Imagen con ruido
E_0	límite inferior de cambio de rango de la imagen
L	Límite superior de cambio de rango de la imagen
$J(i, j)$	Imagen con nuevo rango de valores de grises
$T(i, j), T_{ij}[n]$	Matriz de tiempos
K	Ventana de 3x3 pixeles
$D(i, j)$	Imagen filtrada
$O(i, j)$	Representación de una imagen monocromática
z	Variable aleatoria (nivel de gris)
P_a	Probabilidad de que z tome el valor a
P_b	Probabilidad de que z tome el valor b
$p(z)$	Densidad de probabilidad del ruido
μ	Valor medio de z
σ	Desviación estándar de z
σ^2	Varianza de z
M	Número de filas de la imagen
N	Número de columnas de la imagen
x_{ij}	Pixel central de una vecindad de pixeles
X_{ij}	Conjunto de pixeles en la vecindad
r	Número de observaciones, pixeles en una ventana de filtrado
x_i	Conjunto de r observaciones
$med(X_{ij})$	Mediana de X_{ij}
$prom(X_{ij})$	Promedio de X_{ij}
$A \ominus B$	Erosión de la imagen A con la máscara B
$A \oplus B$	Dilatación de la imagen A con la máscara B
$A \circ B$	Apertura de la imagen A con la máscara B
$A * B$	Cerradura de la imagen A con la máscara B
$OC(A, B)$	Filtro morfológico <i>Open-Close</i>
$CO(A, B)$	Filtro morfológico <i>Close-Open</i>
$H(u, v)$	Función de degradación
$G(u, v)$	Transformada de Fourier de la imagen con ruido $g(x, y)$
$g(x, y)$	Imagen con ruido
K'	Constante de filtrado del filtro <i>Wiener</i>
$\hat{F}(u, v)$	Función del Mínimo Error Cuadrático Medio
$peakSignal$	Máximo nivel de gris de una imagen monocromática
I_{ij}	Imagen original sin ruido
n	Iteración de la Red Pulso-Acoplada
$U_{m,k}, U_{ij}[n]$	Voltaje de membrana de la Red Pulso-Acoplada
$F_k, F_{ij}[n]$	Señal de <i>feeding</i> de la Red Pulso-Acoplada
$L_k, L_{ij}[n]$	Señal de <i>linking</i> de la Red Pulso-Acoplada
$I(V, \tau, t)$	Estímulo externo de la Red Pulso-Acoplada
N'	Neuronas de la Red Pulso-Acoplada

Símbolo	Nombre
w, W_{ijkl}, M_{ijkl}	Matriz de pesos sinápticos
$Y, Y_{ij}[n]$	Salida binaria de la Red Pulso-Acoplada
$\Theta, E_{ij}[n]$	Umbral dinámico de la Red Pulso-Acoplada
V, f, g, h, γ	Constante de ajuste de la Red Pulso-Acoplada
$\tau, e^{-\alpha}$	Constante de tiempo de la Red Pulso-Acoplada
β	Coefficiente de interconexión de la Red Pulso-Acoplada
$rango(x_{ij})$	Posición del x_{ij} en el ordenamiento ascendente de los píxeles en X_{ij}
s	Umbral que define valores extremos
U_2	Umbral que define la distancia entre x_{ij} y $med(X_{ij})$

Capítulo 1 Introducción

En este primer capítulo se realiza la descripción de la motivación a la investigación, el problema de estudio que se resolvió, los objetivos general y particulares que se alcanzaron, los alcances y limitaciones que guiaron el desarrollo de la propuesta y finalmente la metodología empleada para completar la investigación.

1.1 Motivación

El ruido en las imágenes digitales es información no deseada que las distorsiona y causa problemas en tareas de más alto nivel.

Todas las imágenes digitales están sujetas a la presencia de ruido que impide su correcta interpretación en cualquier ámbito donde son requeridas, actualmente su uso está generalizado y ha llegado a ser imprescindible como es el caso de las imágenes médicas, satelitales, de video vigilancia, para robótica o para aplicaciones domésticas usuales como cámaras y *smartphones* (Figura 1.1).



Figura 1.1 Aplicación comercial de Visión Artificial [Sabia.tic.udc.es, 2016]

El filtrado, como etapa del procesamiento digital de imágenes, es capaz de disminuir el ruido. En este proceso cada pixel en la imagen de salida es calculado como función de algunos pixeles de la imagen de entrada, usualmente localizados cerca del pixel de salida.

Entre los métodos usados para el filtrado del ruido en imágenes digitales están los filtros diseñados en frecuencia, las Redes Neuronales Artificiales (RNA), las Wavelets, los Algoritmos Difusos, entre muchos otros [Bovik, 2009], sin embargo el filtrado de imágenes digitales es una tarea que aún se sigue explorando y perfeccionando.

El desarrollo de nuevas herramientas en el campo de la Visión Artificial se ha hecho posible mediante la emulación de la corteza visual de los mamíferos, de modo que al comprender cómo funciona ésta se elaboran modelos que hacen posible el filtrado del ruido en imágenes, la segmentación o el reconocimiento, entre otras tareas. Estos modelos matemáticos se conocen como Redes Neuronales Pulso-Acopladas o PCNN por sus siglas en inglés, y fueron desarrollados por Eckhorn en la década de los 90 y aplicados al procesamiento de imágenes por Kuntimad [Lindblad, 2005].

En el CENIDET se ha comenzado a experimentar con las Redes Neuronales Pulsantes para completar la extracción de características y en la segmentación de imágenes [Zárate, 2015] y [Cárdenas, 2015].

Mediante esta tesis se pretende implementar algoritmos de filtrado basados en PCNN, experimentando con dos tipos de ruido, utilizando imágenes digitales en escala de grises, y conformar un estudio comparativo de los parámetros de funcionamiento de este tipo de Redes Neuronales evaluando su desempeño en tiempo de procesamiento y calidad de recuperación con respecto a las técnicas tradicionales de filtrado.

1.2 Descripción del problema

Todas las imágenes digitales son susceptibles de ser afectadas por el ruido, desde su adquisición debido a los elementos sensores en cámaras, los dispositivos de captura y en el momento de su compresión y transmisión a causa de medios o canales defectuosos.

El ruido digital se caracteriza por su fuente y los efectos visuales que causa en las imágenes, por ejemplo, en una imagen degradada por ruido impulsivo los píxeles están afectados cuando sus valores de gris tienden a ser extremadamente altos o bajos teniendo poca correlación con los píxeles circundantes; por otro lado el ruido aditivo gaussiano afecta a todos los píxeles de la imagen, por lo que es difícil de remover.

Dependiendo del tipo de ruido que se desee tratar se han implementado distintos algoritmos de filtrado para recuperar la información de la imagen, no obstante el problema de filtrado aún no se ha podido resolver de modo definitivo, pues en algunos casos el ruido no es completamente eliminado o el proceso conlleva una indeseable pérdida de los detalles de la imagen [González, 2002].

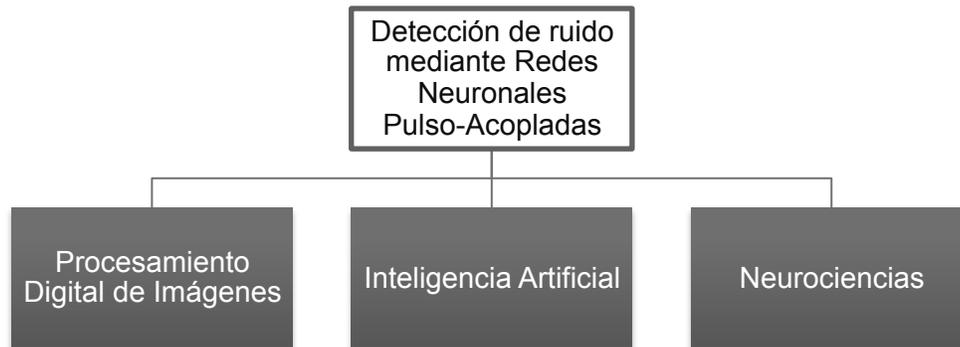


Figura 1.2 Ubicación de la solución del problema

Una forma propuesta recientemente para resolver este problema consiste en implementar técnicas de Inteligencia Artificial, como las Redes Neuronales, especialmente empleando las propiedades de las Redes Neuronales Pulso-Acopladas en la detección de ruido impulsivo y gaussiano (Figura 1.2). Las redes PCNN han mostrado ventaja experimentalmente con respecto a las RNA tradicionales [Ramírez, 2011], [Wang, 2010] y a las técnicas tradicionales de filtrado [Tang, 2007], [Ma, 2010], no obstante dichos trabajos no permiten destacar los mejores resultados ni llegar a afirmaciones especializadas acerca de su desempeño.

1.2.1 Delimitación del problema específico

El problema específico de la presente investigación se puede enunciar como: encontrar los parámetros de los algoritmos de detección de ruido impulsivo y gaussiano basados en Redes Neuronales Pulso-Acopladas que generen mejores resultados en remoción de ruido y conservación de detalles al probarse con imágenes digitales estándar monocromáticas.

1.2.2 Complejidad del problema

En el CENIDET se ha implementado la red PCNN para realizar tareas de segmentación y extracción de características, no obstante el filtrado de imágenes representa un problema distinto al corresponder a otra etapa del procesamiento de imágenes, dichas tareas coinciden sólo en la etapa inicial del proceso, en la que se centró la investigación y donde se propuso hallar la mejor práctica es en el filtrado de ruido.

Hasta el momento no existe un estudio previo que permita comparar, de manera ágil y confiable, a las técnicas de filtrado tradicionales y las técnicas basadas en PCNN para establecer cuáles son superiores en calidad de la imagen filtrada, conservación de detalles y tiempo de procesamiento.

No se cuenta con un estudio precedente que permita determinar cuáles son los parámetros de los algoritmos de filtrado basados en PCNN para obtener los mejores resultados en calidad del filtrado y tiempo de procesamiento puesto que el número de parámetros que se requieren ajustar varía entre tres y nueve, donde cada uno puede tomar valores reales, de modo que la combinatoria resultante es infinita.

Por medio de esta tesis se llevó a cabo el estudio, la implementación y la experimentación de algoritmos de detección de ruido de imágenes digitales basados en el modelo PCNN y comparar su desempeño contra las técnicas tradicionales, de modo que se puedan establecer claramente sus ventajas y desventajas para procesar imágenes en escala de grises con ruido impulsivo y ruido gaussiano.

1.3 Objetivo general

Implementar y evaluar la RNA tipo Pulso-Acoplada para problemas de detección de ruido en imágenes digitales.

1.4 Objetivos específicos

1. Analizar los tipos de ruido en imágenes digitales.
2. Implementar algoritmos para simular tipos de ruido específicos en imágenes digitales.
3. Revisar los problemas típicos de las técnicas de filtrado tradicionales.
4. Implementar algoritmos tradicionales de filtrado.
5. Revisar los problemas típicos de las RNAs en el dominio de Visión Artificial.
6. Implementar la RNA Pulso-Acoplada y variantes para problemas de filtrado de ruido en imágenes digitales.
7. Experimentar con bases de datos de imágenes digitales conocidas en CENIDET y de uso común en la comunidad de Visión Artificial.
8. Obtener los valores de las métricas de restauración de imágenes.
9. Analizar los resultados obtenidos.
10. Realizar un estudio comparativo.

1.5 Alcances y limitaciones del proyecto

1. Analizar por lo menos el ruido impulsivo bipolar y el ruido gaussiano en imágenes digitales.
2. Implementar algoritmos para la simulación de ruido impulsivo y gaussiano en imágenes digitales.
3. Analizar por lo menos tres técnicas tradicionales para el filtrado de imágenes.

4. Implementar por lo menos tres algoritmos tradicionales para el filtrado de imágenes.
5. Elegir e implementar por lo menos tres variantes de RNA Pulso-Acoplada para problemas de filtrado de ruido en imágenes digitales.
6. Experimentar con al menos cinco bases de datos de imágenes digitales.
7. Aplicar por lo menos dos métricas para evaluar la restauración de las imágenes.
8. Realizar el estudio comparativo de la calidad de la imagen y tiempo de procesamiento.
9. Las comparaciones se realizarán con respecto a lo reportado en la literatura y utilizando software desarrollado de uso libre.
10. No se busca mejorar resultados sino detectar claramente las ventajas y desventajas de la implementación de algoritmos de filtrado con PCNN.

1.6 Justificación y beneficios

Las Redes Neuronales Pulso-Acopladas son una importante herramienta para la Visión Artificial ya que pueden aplicarse en distintas etapas del procesamiento de imágenes y al ser una emulación de un sistema biológico representan un nuevo enfoque para resolver los problemas típicos del filtrado de imágenes como parte del área de la Inteligencia Artificial.

Las Redes Neuronales constituyen una técnica ampliamente explotada en los ámbitos de la clasificación, el seguimiento de señales, la aproximación de funciones, la predicción y el reconocimiento, no obstante las Redes Neuronales Pulso-Acopladas representan un reciente objeto de estudio en el área del procesamiento de imágenes y en el CENIDET, además de que sus parámetros de funcionamiento se encuentran en constante modificación y han atraído la atención de los investigadores alrededor del mundo.

Este trabajo aportará conocimiento y una implementación en cada una de sus etapas, elementos útiles para el CENIDET y para investigadores del ramo, primero con respecto a las técnicas de filtrado tradicionales, posteriormente con respecto a las técnicas de remoción de ruido basadas en las PCNN y finalmente en cuanto al pre-procesamiento de imágenes como parte de los sistemas de Visión Artificial.

1.7 Metodología de solución

El proceso completo de la investigación consta de cuatro etapas (Figura 1.3), en la primera etapa se llevó a cabo la investigación del principio y funcionamiento de las redes tipo PCNN.

En la segunda etapa se implementaron tres algoritmos tradicionales de filtrado y tres basados en las redes PCNN simulando los dos tipos de ruido y aplicando diversas métricas para comparar los resultados mediante una plataforma de experimentación creada específicamente.

En la tercera etapa se realizó la experimentación de la variación de los parámetros del algoritmo, modificando en primer lugar las imágenes de entrada, luego los parámetros del modelo y por último los algoritmos mismos; finalmente en la cuarta etapa se concretó un estudio comparativo entre técnicas de filtrado y se dieron a conocer los principales resultados obtenidos de la experimentación con los modelos de PCNN.

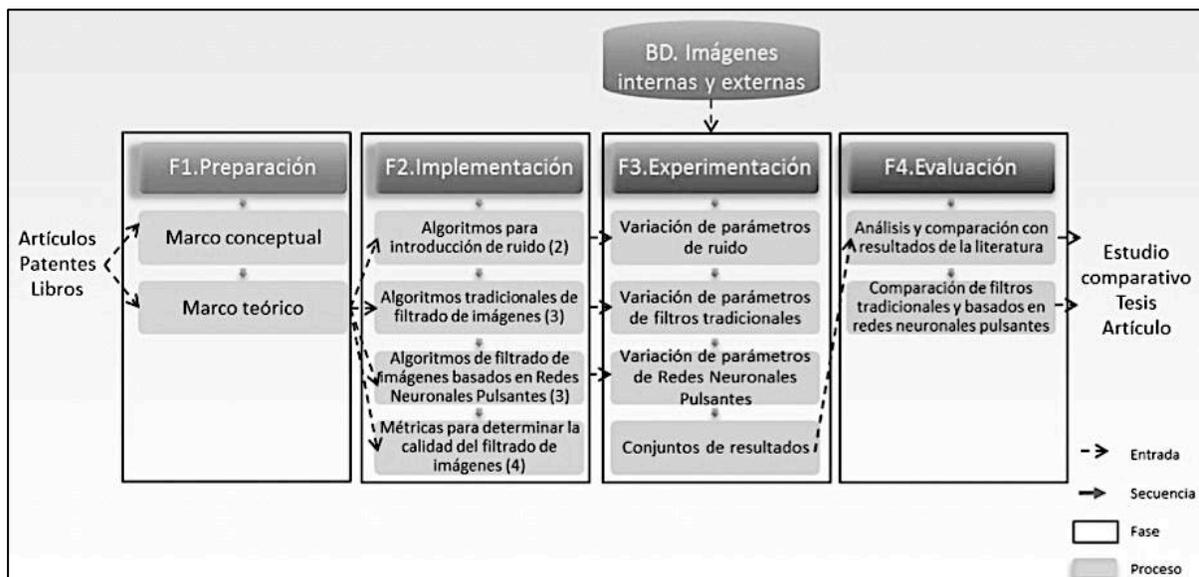


Figura 1.3 Metodología de solución

A partir de los algoritmos de filtrado de ruido tradicionales y basados en PCNN se seleccionaron los más completos para su implementación, una vez que los algoritmos fueron funcionales se aplicaron a bases de datos de imágenes afectadas con distintos tipos y niveles de ruido impulsivo y gaussiano, luego se determinaron las características de filtrado de los algoritmos midiendo la recuperación de la imagen, la conservación de los bordes y el tiempo de procesamiento, también se observó el comportamiento de los filtros basados en PCNN al variar sus parámetros. Con la información recabada se conformó el estudio comparativo que permite visualizar las ventajas y desventajas de las distintas técnicas de filtrado de imágenes digitales exploradas.

1.8 Organización de la tesis

Con el fin de orientar al lector acerca del contenido del presente trabajo se enumeran y describen brevemente sus partes constitutivas: el Capítulo 2 contiene el estudio del estado del arte, el Capítulo 3 contiene el marco teórico, el Capítulo 4 abarca los detalles de la implementación de los algoritmos de filtrado tradicionales y basados en PCNN, el Capítulo 5 contiene distintos experimentos con los algoritmos implementados y los resultados obtenidos.

Capítulo 2 Estado del Arte

En este capítulo se detallan los antecedentes más próximos e importantes, tanto internos como externos al CENIDET, sobre filtrado de ruido en imágenes y el uso de Redes Neuronales Pulso-Acopladas, del mismo modo se resume dicha información y se cierra el capítulo con una discusión sobre lo descrito en estos materiales.

2.1 Antecedentes institucionales

Se mencionan algunos proyectos desarrollados en el CENIDET que experimentaron con las redes tipo PCNN y están relacionados con el trabajo que se realizó.

Implementación y Evaluación de Redes Neuronales Artificiales tipo "Pulse-Coupled Neural Networks" (PCNN) Aplicadas a Visión Artificial [Cárdenas, 2015]. Consiste en un estudio comparativo sobre la segmentación de imágenes por medio de técnicas como PCNN y técnicas clásicas, utilizaron las metodologías de Entropía Cruzada y *Pixel Correspondance Metric* (PCM). Se aplicó un mecanismo para evaluar el desempeño de las PCNN en la segmentación de imágenes y se determinó la eficiencia de las variantes para la detección de bordes con apoyo de una plataforma de experimentación (Figura 2.1).

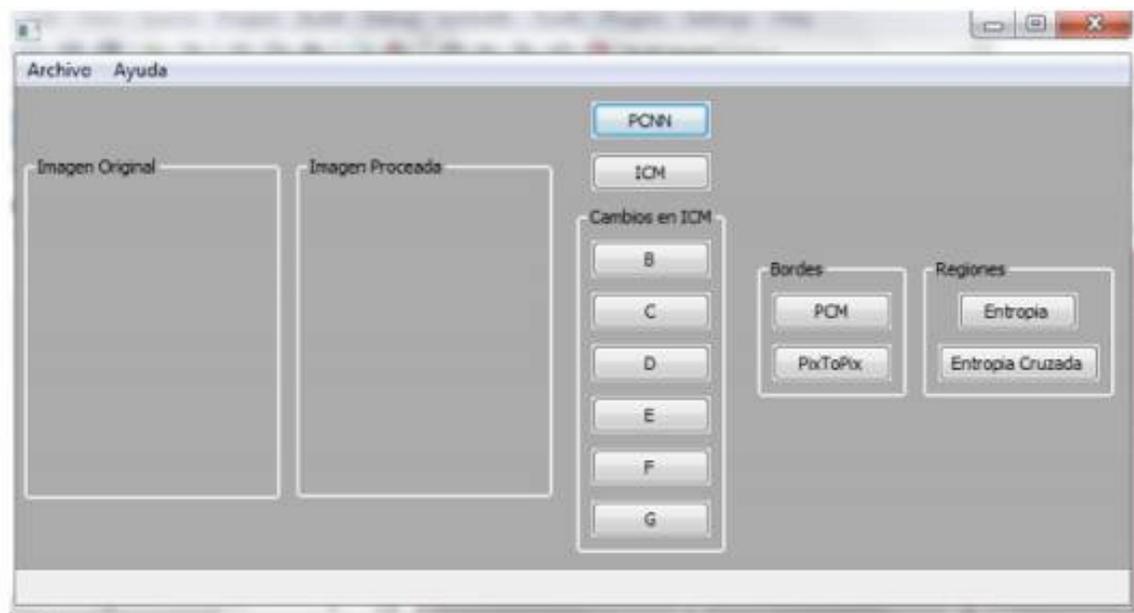


Figura 2.1 Plataforma de pruebas para segmentación con PCNN [Cárdenas, 2015]

Extracción de Características de Imágenes Digitales mediante una Red Neuronal Artificial Pulsante [Zárate, 2015]. Se trabajó con los modelos de PCNN, *Intersecting Cortical Model* (ICM) y *Spiking Cortical Model* (SCM) para la extracción de descriptores de las imágenes a partir de las firmas generadas por la red neuronal (Figura 2.2).

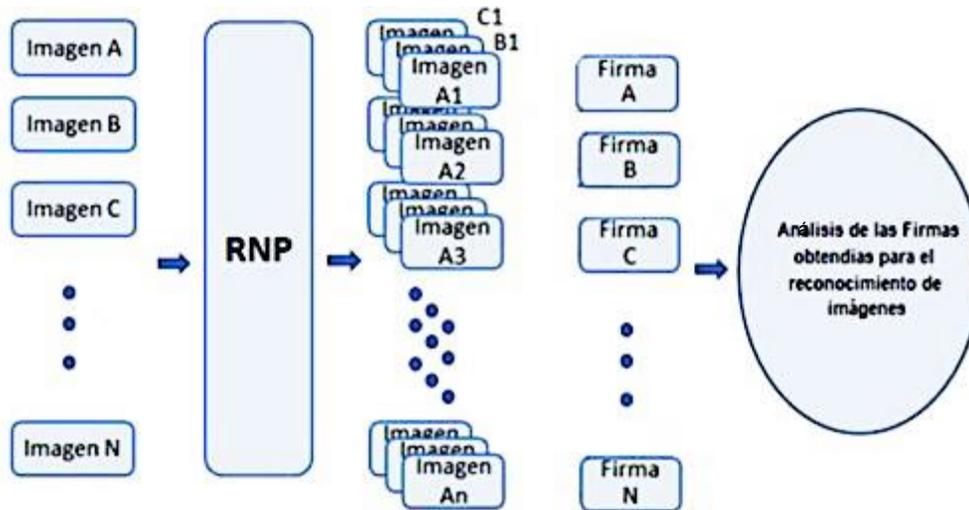


Figura 2.2 Metodología extracción de características con PCNN [Zárate, 2015]

2.2 Trabajos relacionados

Se mencionan algunos artículos recientes externos al CENIDET que fueron considerados inicialmente para llevar a cabo el estudio comparativo propuesto en esta tesis. En primer lugar se presentan tres artículos de filtros tradicionales y posteriormente cinco trabajos de filtros basados en las redes PCNN.

Removal of Salt and Pepper Noise for Various Images Using Median Filters: A Comparative Study [Sreenivasulu, 2014]. Se discutieron los resultados de un estudio comparativo sobre el filtrado de ruido de sal y pimienta, utilizaron varios filtros de mediana para distintas imágenes de prueba. Los algoritmos implementados fueron: *Standard Median Filter* (MF), *Adaptive Median Filter* (AMF), *Adaptive Weighted Algorithm* (AWA), *Switching Median Filter* (SMF), *Decision-Based Algorithm* (DBA) y *Modified Decision-Based Unsymmetric Trimmed Median Filter* (MDBUTMF), que fueron comparados utilizando el MSE (*Mean Square Error*) y el PSNR (*Peak Signal to Noise Ratio, dB*). Para analizar el desempeño de los filtros se emplearon imágenes con probabilidad de ocurrencia de ruido de 0.6, 0.7 y 0.8 (Figura 2.3), se consideraron exclusivamente algoritmos basados en el filtro de mediana y se trabajó con imágenes en escala de grises. No se comparó ni mencionó la complejidad de cada algoritmo al momento de programarlo.

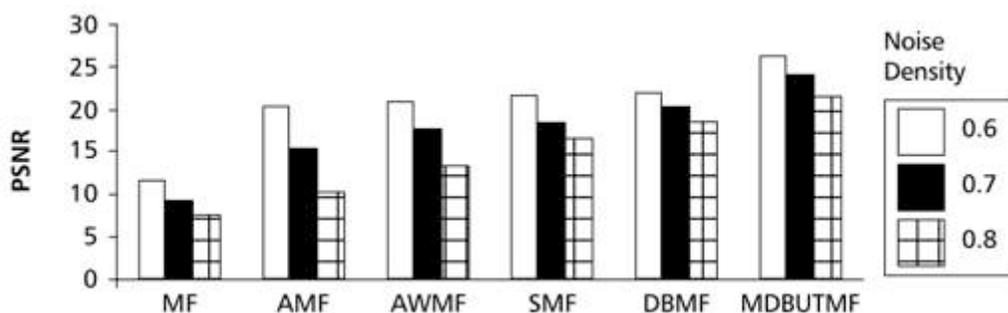


Figura 2.3 PSNR para filtros de mediana con Cameraman [Sreenivasulu, 2014]

Efficient Noise Reduction in Images Using Directional Modified Sigma Filter [Kang, 2013]. Se propuso la reducción de ruido utilizando un filtro sigma direccional modificado, para su aplicación la imagen se dividió en componentes con características en las direcciones horizontal, vertical y diagonal, después se aplicó un filtro pasa altas, un pasa bajas y el filtro sigma convencional en cada componente por separado y se reconstruyó la imagen de salida. Adicionalmente se calculó una aproximación a la desviación del ruido por medio del BNE (*Block Based Noise Estimation*) utilizando el AGF (*Adaptive Gaussian Filter*). El algoritmo se evaluó con imágenes de *Lena*, *Pepper* y *Airplane* de 512 X 512 de 8 bits, se aplicó ruido gaussiano de media 0 con varianzas de 0 a 25. Se compararon los resultados de PSNR contra el filtro Sigma, el filtro Sigma Modificado y el filtro Gaussiano. El método está actualmente implementado para aplicaciones de vigilancia.

Novel Linear Image Denoising Approach Based on a Modified Gaussian Filter Kernel [Barbu, 2012]. Se propuso una técnica para la reducción del ruido gaussiano en imágenes en escala de grises, utilizaron un núcleo de filtro gaussiano modificado basado en la ecuación hiperbólica de segundo orden. La técnica propuesta se fundamentó en modelos de ecuaciones diferenciales parciales formuladas mediante un algoritmo iterativo cuyos parámetros se obtuvieron experimentalmente. Los resultados se compararon con técnicas de filtrado lineales como el filtro Gaussiano bidimensional y el filtro promedio de 3x3 sobre cientos de imágenes estándar de 512x512 como *Lena* y *Peppers*, con ruido gaussiano de desviación de 0.5. El algoritmo hiperbólico de reducción de ruido gaussiano incrementó la calidad de la imagen, removió más ruido que la técnica tradicional, se ejecutó muy rápidamente y presentó baja complejidad computacional.

Adaptive Image de-Noiseing Algorithm in Intersecting Cortical Model [Ling, 2013]. Se propuso un marco y parámetros relacionados para la red ICM en un algoritmo de filtrado adaptativo que empleó la Matriz de Tiempos (TM) y se aplicó el filtro de mediana selectivamente. El algoritmo fue llamado IICM y disminuye el ruido impulsivo en imágenes en escala de grises, se mejoró el establecimiento de los parámetros del ICM, se introdujo la Matriz de Tiempos que determinó adaptativamente el número de iteraciones necesarias. Sólo se modificaron los píxeles ruidosos por lo que se preservaron detalles y bordes y se determinó adaptativamente el tamaño de la ventana de procesamiento de acuerdo al grado de contaminación.

La experimentación se llevó a cabo con Matlab 7.1, utilizando imágenes de prueba estándar de *Lena* y *Peppers* en escala de grises de 256 valores de tamaño 512x512 a las que se agregó ruido impulsivo de probabilidades de ocurrencia del ruido del 10% al 60%. Se comparó al algoritmo propuesto con el filtro de mediana, el PCNN, el método de Kenny KVT (Mediana Adaptativo Difuso), el propuesto por Srinivasan y el método de Lv Zongwei, sin embargo no se mostraron mediciones de MSE y PSNR.

Applications of Pulse-Coupled Neural Networks [Ma, 2010]. Se propusieron tres algoritmos basados en PCNN para reducir el ruido impulsivo y el ruido gaussiano en imágenes digitales. El Algoritmo I empleó la red SCM para remover el ruido impulsivo modificando progresivamente la intensidad de los píxeles ruidosos hasta que se encendieron al mismo tiempo que sus vecinos. El Algoritmo II (Figura 2.4) utilizó la combinación del modelo SCM con filtros promedio, de mediana y morfológicos. Se mostró que en imágenes de 5% y 10% de probabilidad de ruido impulsivo el método propuesto superó a los filtros tradicionales en PSNR (dB).

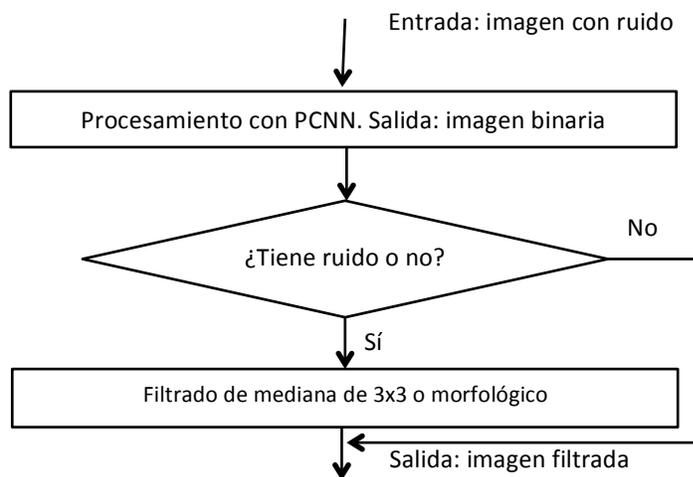


Figura 2.4 Algoritmo II de filtrado mediante PCNN [Ma, 2010]

Para reducir el ruido gaussiano emplearon el modelo PCNNNI que evita la interacción entre neuronas y propusieron un Algoritmo III (Figura 2.5), mediante el cual se hizo un cambio del rango dinámico de la imagen, se iteró la red PCNNNI, se obtuvo una Matriz de Tiempos donde se guardaron las distintas secuencias de activación y se seleccionó el método para restaurar la imagen. Se compararon los resultados de aplicar los Algoritmos II (SCM+morfológico y SCM+mediana) y el Algoritmo III contra el filtro *Wiener* en imágenes con ruido gaussiano de media 0 y varianza de 0.001 y 0.0005, los resultados mostraron que no se superó a los filtros tradicionales.

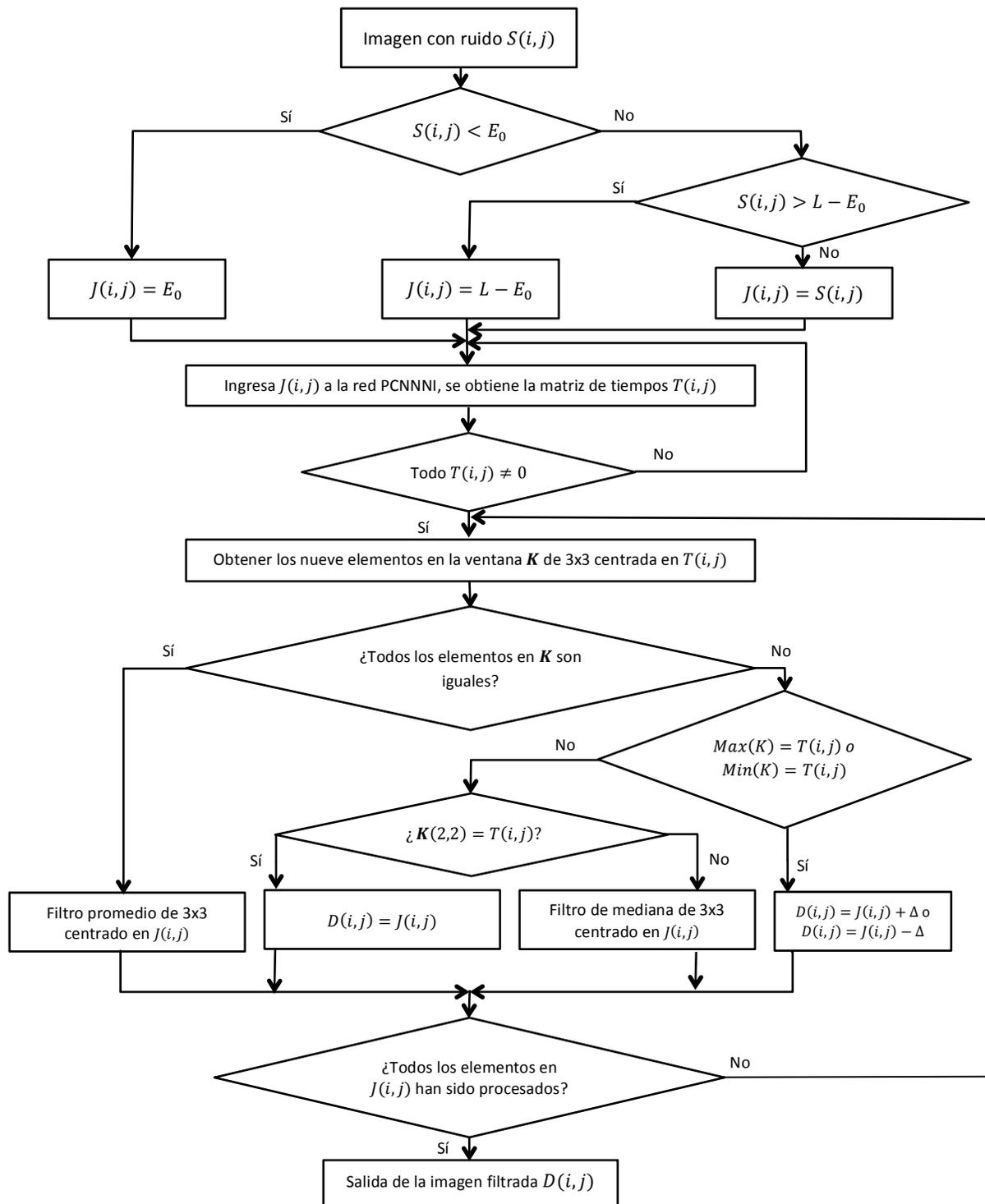


Figura 2.5 Algoritmo III de filtrado mediante PCNN [Ma, 2010]

Noise Reduction Using Genetic Algorithm Based PCNN Method [Zhang, 2010]. Se presentó un Algoritmo Genético Adaptativo basado en el modelo PCNN para disminuir el ruido blanco gaussiano (AWGN). Se utilizó un filtro de Difusión Anisotrópica (AD) que reemplazó al filtro de mediana y optimizó los parámetros de la red PCNN simplificada por medio de un Algoritmo Genético (GA). Se compararon los resultados obtenidos con el GA-PCNN contra el filtro de mediana, el filtro *Wiener*, el filtro de Difusión Anisotrópica y el filtro tradicional basado en PCNN. Los resultados mostraron que el filtro de mediana generó la peor reducción de ruido, el AD y el filtro *Wiener* fueron inferiores a la red PCNN-AD en reducción de ruido y preservación de los bordes. El GA-PCNN superó al método PCNN-AD y al filtro *Wiener*, además redujo a menos de la mitad el tiempo de ejecución de la técnica PCNN-AD.

Modified PCNN Model and Its Application to Mixed-noise Removal [Shao-Fa, 2010]. Se propuso un esquema de filtrado de ruido gaussiano y ruido impulsivo basado en un modelo de PCNN modificado llamado L&A-PCNN [Shao-Fa, 2008], que constó de un umbral de atenuación lineal y una salida de nivel medio de gris ponderado para los pixeles activados que reemplazaron a las salidas binarias del PCNN tradicional, el cual se aplicó con efectividad para el filtrado de ruido mezclado en imágenes monocromáticas [Shao-Fa, 2007].

En [Shao-Fa, 2010] se estableció un método automático para el cálculo del coeficiente de conexión a partir de la desviación cuadrática media del valor de gris cercano al pixel. El algoritmo fue comparado contra el filtro de mediana, el filtro de Reglas Difusas y el método L&A-PCNN propuesto en [Shao-Fa, 2007] como se muestra en la Figura 2.6. Se obtuvieron valores superiores a todos ellos en SNR (*Signal to Noise Ratio*), más el establecimiento automático de parámetros de la red.



Figura 2.6 a) Imagen contaminada con ruido impulsivo de probabilidad 10% y ruido gaussiano de $\sigma^2=0.001$, resultado del filtro de b) mediana, c) reglas difusas, d) L&A-PCNN, e) método propuesto [Shao-Fa, 2010]

An Effective Adaptive Median Filter Algorithm for Removing Salt & Pepper Noise in Images [Jiang, 2010]. Se propuso un algoritmo basado en el filtro de mediana adaptativo utilizado en conjunto con la red PCNN para remover ruido impulsivo. Se mostró que el modelo simplificado de PCNN falló al remover el ruido impulsivo y luego fue mejorado utilizando el método de divide y vence. El filtro de mediana fue utilizado para establecer el nivel de ruido presente, luego se determinó la localización específica del ruido y el tamaño del filtro de mediana adaptativo a partir de la PCNN modificada.

Se verificó la validez del método utilizando Matlab 7.0 y la imagen de Lena a la que se agregó ruido impulsivo de probabilidades desde 0.1 hasta 0.8. Se compararon los resultados contra el filtro de mediana, el filtro adaptativo de mediana y para cada caso se calculó el MAE (*Mean Absolute Error*) y el PSNR. Se logró la detección del nivel de ruido, su localización y la determinación del tamaño de la ventana de filtrado. El método propuesto tuvo un buen comportamiento, por ejemplo el MAE (Figura 2.7) mostró que el método propuesto generó un error menor a 5 para probabilidades superiores a 0.6, mientras que el PSNR mostró que para probabilidades de ruido mayores 0.5 se mantuvo entre 26dB y 30dB.

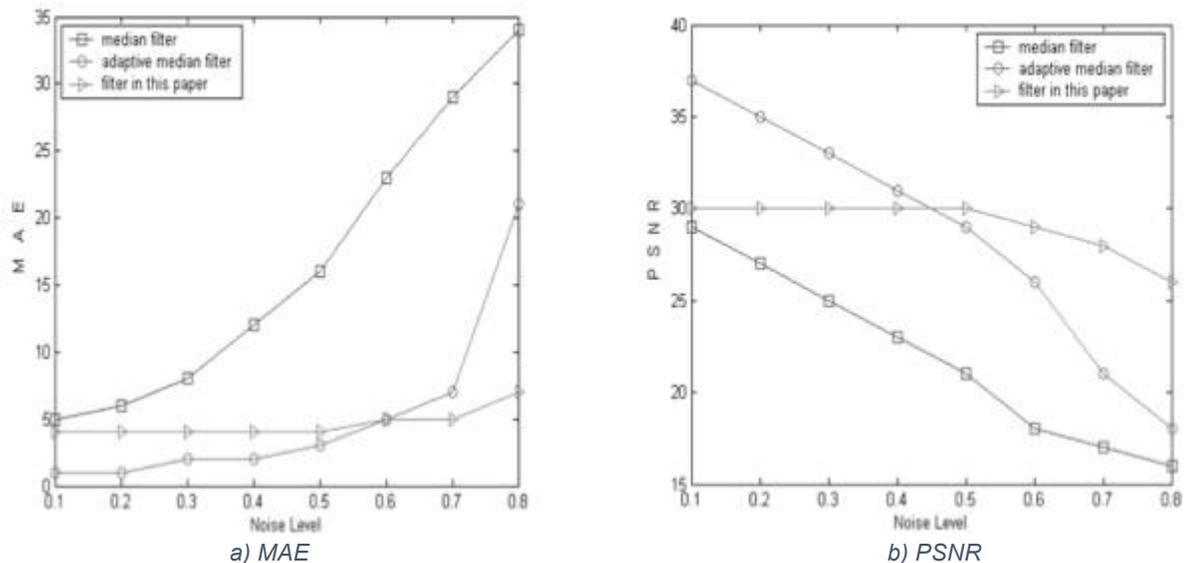


Figura 2.7 MAE y PSNR de diferentes tipos de filtrado [Jiang, 2010]

2.3 Discusión del estado del arte

El proyecto desarrollado en [Cárdenas, 2015] constituye el primer trabajo de implementación de PCNN en el CENIDET, realiza la comparación del desempeño de la segmentación utilizando PCNN y técnicas tradicionales, logra detectar algunas de las ventajas y desventajas de las PCNN en éste ámbito, no obstante la base de datos de las imágenes que utiliza es propia por lo que no se pueden comparar los resultados obtenidos con bases de datos estándar, así mismo no considera el tiempo de procesamiento como parte del estudio comparativo.

Otro trabajo [Zárate, 2015] constituye un segundo esfuerzo en CENIDET para implementar las redes PCNN y utiliza sus variantes para la extracción de características de imágenes digitales por lo que se pueden comparar algunos resultados de la implementación, fuera del tiempo de procesamiento, y utilizar la base de datos de imágenes disponible. En resumen, estos dos trabajos no abordan el filtrado de ruido con PCNN, por lo que la presente propuesta se orienta hacia la realización de esta tarea.

De las diversas técnicas revisadas, se obtuvo la información de la Tabla 2.1 que sintetiza las principales características de cada una. En dicha tabla se incluye el nombre de la técnica de filtrado, el tipo y rango de ruido tratado, si los parámetros del modelo se incluyen en el trabajo, la base de datos de imágenes utilizada, las métricas de evaluación del algoritmo, el tiempo de procesamiento, la aplicación actual de la propuesta y la plataforma de desarrollo empleada.

Tabla 2.1 Síntesis de los trabajos relacionados

	Técnica de filtrado	Tipo y rango de ruido que filtran	Parámetros del modelo	Base de datos de imágenes	Medida de evaluación	Tiempo de procesamiento, aplicación y plataforma de desarrollo
Filtros tradicionales	Mediana y sus variantes [Sreenivasulu, 2014]	Impulsivo ($p=[0.6-0.8]$)	No explícitos para todos los filtros.	Escala de grises. <i>Cameraman, coins, Color.</i>	PSNR MSE	No especificados.
	Sigma Direccional Modificado [Kang, 2013]	Gaussiano ($\mu=0, \sigma^2=0-0.013$)	Ventana de 5x5.	Escala de grises de 512x512 en 8 bits. <i>Lena, Peppers, airplane</i>	PSNR	Cámaras, TV y teléfonos digitales, actualmente en video vigilancia.
	Gaussiano Modificado [Barbu, 2012]	Gaussiano ($\mu=0, \sigma^2=0.5$)	Obtenidos experimentalmente	Escala de grises de 512x512, cientos de imágenes.	Cualitativo	Rápido, no cuantificado.
Filtros basados en PCNN	PCNN con Mediana Adaptativa [Ling, 2013]	Impulsivo ($p=[0.1-0.8]$)	Explícitos	Escala de grises. <i>Lena.</i>	PSNR, MAE	Rápido, no cuantificado. Matlab.
	SCM con Morfológico, Mediana y Promedio. PCNNNI [Ma, 2010]	Impulsivo (0.05 y 0.1) y gaussiano ($\mu=0, \sigma^2=[0.001, 0.0005]$)	Explícitos	Escala de grises. <i>Lena, Cameraman, characters.</i>	PSNR	No especificados.
	Algoritmo Genético Adaptativo+PCNN [Zhang, 2010]	Ruido blanco gaussiano ($\sigma^2=0.09$ y $[0.006-0.009]$)	Obtenidos experimentalmente	Escala de grises de 256x 256 del USC-SISP	PSNR MSE, Preservación de bordes	Especificado. DSP
	PCNN con salidas Ponderadas [Shao-Fa, 2010]	Impulsivo ($p=[0.1-0.3]$) y gaussiano ($\sigma^2=0.001, 0.005, 0.01$) mezclados	Obtenidos mediante heurística y método automático	Escala de grises e imágenes a color formato RGB <i>Lena, Peppers, rice, Cameraman, airplane</i>	PSNR, RMSE, error medio	No apto para tiempo real. Matlab.
	ICM Modificado [Jiang, 2010]	Impulsivo ($p=[0.1-0.6]$)	Explícitos	Escala de grises de 8 bites, 512x512. <i>Lena y Peppers</i>	PSNR MSE	Matlab.

Del análisis de los trabajos revisados se puede decir que las técnicas tradicionales de filtrado de ruido gaussiano han llegado a ser eficientes y complejas [Barbu, 2012], por lo que se han simplificado sus cálculos dividiendo el procesamiento en etapas [Kang, 2013]. El problema del ruido impulsivo ha sido resuelto por variedad de técnicas selectivas [Sreenivasulu, 2014] donde se muestra que aún en las técnicas robustas y aplicadas comercialmente se siguen generando mejoras, no obstante el tiempo de procesamiento y falta de efectividad con probabilidades de ruido altas aún representa una limitante en su predominancia con respecto a otros métodos alternativos de filtrado.

Las investigaciones sobre filtrado de imágenes basados en PCNN dejan ver que presentan niveles de efectividad superiores a las técnicas tradicionales de filtrado [Ma, 2010]. Su implementación es relativamente clara, cada vez se tiene una mayor exactitud en el establecimiento de sus parámetros [Zhang, 2010] y se dispone de mayor autonomía en su funcionamiento ante cualquier imagen con ruido que se les presente mediante la detección y medición del mismo [Jiang, 2010], tanto con imágenes en escala de grises o a color como con ruido mezclado [Shao-Fa, 2010].

Para hacer el estudio comparativo entre técnicas de filtrado tradicionales y las técnicas basadas en PCNN se eligió el conjunto de imágenes de prueba estándar, los tipos de ruido impulsivo y gaussiano, las métricas de comparación comúnmente usadas, y las técnicas tradicionales contra las cuales se comparan como los filtros básicos de mediana, promedio, morfológico y filtro *Wiener*.

Capítulo 3 Marco Teórico

En este capítulo se incluyen los conceptos colectados durante la investigación, los modelos matemáticos del ruido en imágenes digitales, las características de los filtros tradicionales, los principales modelos de Redes Neuronales Pulso-Acopladas y las métricas de calidad de la imagen restaurada empleadas para comparar la calidad de las imágenes filtradas.

3.1 Visión Artificial

La visión humana es un proceso biológico debido a la percepción de la luz por medio de los receptores en los ojos de donde viaja la información a la corteza visual en el cerebro, dicho proceso ha sido ampliamente estudiado desde la antigüedad y aún en la actualidad no ha sido descubierto por completo. Las imágenes digitales son obtenidas por un proceso similar donde los receptores son elementos sensibles a la radiación y la información es interpretada por medio de una computadora.

Un sistema de Visión Artificial actúa sobre una representación de la realidad que le proporciona información sobre brillo, colores, formas, etc. Estas representaciones suelen estar en forma de imágenes estáticas, escenas tridimensionales o imágenes en movimiento. Una imagen bidimensional es una función que a cada par de coordenadas (i, j) asocia un valor relativo a alguna propiedad del punto que representa.

Para la Visión Artificial una imagen puede ser acromática si carece de información del color y sólo depende de un valor referente al brillo, mientras que una imagen en el modelo de color RGB (*Red Green Blue*) se puede representar asociando a cada punto una terna de valores que indican la intensidad de tres componentes, una roja, otra verde y otra azul, pero también es posible representar una imagen de color de espectro completo asociando a cada punto un diagrama espectral de emisión de color.

3.2 Procesamiento Digital de Imágenes

3.2.1 Etapas del Procesamiento Digital de Imágenes

El Procesamiento Digital de Imágenes se encarga de los aspectos relacionados con la adquisición y posterior uso de las imágenes, algunas de sus principales etapas son el preprocesamiento de la imagen, que incluye el filtrado de ruido resultante de la adquisición, compresión y transmisión, el mejoramiento de la imagen y su restauración.

La segmentación es el siguiente paso, divide una imagen en sus partes constitutivas indicando límites entre regiones de píxeles para reconocer objetos, etc.

La representación consiste en la conversión de la información en una forma adecuada para su posterior procesamiento, dicha conversión se basa en la decisión de representar bordes y límites o regiones completas; por otra parte están la descripción y la selección de atributos, que da como resultado el conjunto de información de interés de una clase de objetos o permite distinguir una clase de otras.

El reconocimiento e interpretación es el proceso que asigna una etiqueta a un objeto basándose en sus descriptores. Dichas etapas interactúan con una base de conocimientos donde se encuentra codificado el conocimiento acerca del dominio del problema en forma de una base de datos que también controla la interacción entre los módulos [González, 2002].

El diagrama de la Figura 3.1 contiene las principales etapas del procesamiento de imágenes con especial énfasis en la etapa de pre procesamiento, en la cual se encuentra el trabajo que se propone realizar.

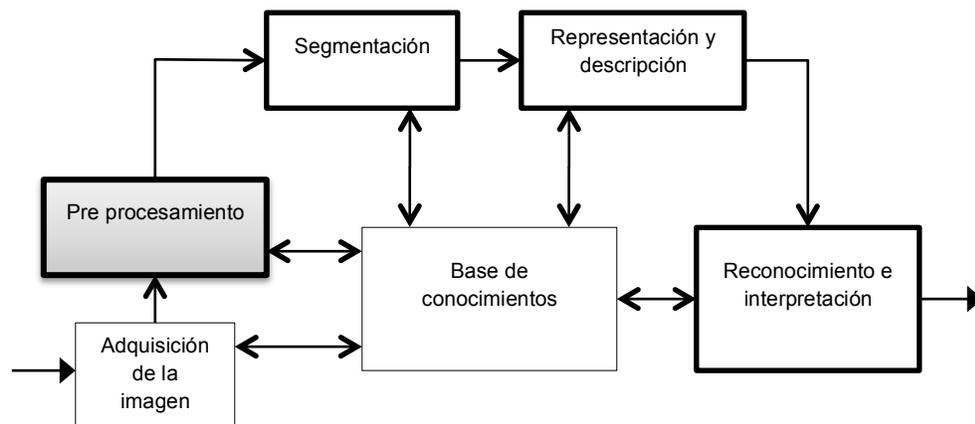


Figura 3.1 Diagrama de bloques mostrando el proceso de Procesamiento Digital de Imágenes [Pajares, 2008]

Las imágenes digitales son fotografías que han sido almacenadas en una computadora digital mediante una codificación, usualmente son obtenidas por medio de la conversión de una señal continua al formato digital por medio de algún dispositivo. Pueden ser visualizadas por distintos medios como las impresoras digitales, los monitores de las computadoras y los dispositivos de proyección digital.

La frecuencia con la cual la información es transmitida, almacenada, procesada y desplegada en un formato digital se incrementa rápidamente, y del mismo modo lo hacen el diseño de los métodos de ingeniería para hacer más eficiente su transmisión y mantenimiento. Por lo tanto, dotar de integridad visual a esta información es de un interés primordial.

3.2.2 Tipos de ruido en imágenes digitales

Una imagen digital puede estar sujeta a ruido e interferencia de algunas fuentes, incluyendo ruido eléctrico en el elemento sensor, ruido de grano fotográfico y errores en los canales de transmisión, el ruido distorsiona la imagen y dificulta su procesamiento posterior, por ejemplo el ruido impulsivo proveniente de un sensor o canal de transmisión defectuosos, usualmente aparece como variación en un pixel aislado que no está espacialmente correlacionado, el ruido puede ser descrito mediante variables estadísticas, en particular por una función de densidad de probabilidad.

Una imagen monocromática (en escala de grises) O es representada frecuentemente por un arreglo bidimensional donde una dirección (i, j) define una posición en O , llamada pixel. En una imagen en escala de grises el valor que posee un pixel es llamado intensidad y se guarda como un entero de 8 bits teniendo 256 posibles valores de gris que van del negro al blanco [Schulte, 2006], los cuales pueden ser representados como el intervalo $[0, 255]$ (Figura 3.2). Una imagen monocromática de dimensiones 512×512 con pixeles de resolución de 8 bits requiere un volumen de datos de 263 KB que la contenga [Bovik, 2009].

En una imagen en escala de gris, el ruido impulsivo bipolar tiene una función de densidad de probabilidad caracterizada como:

$$p(z) = \begin{cases} P_a & \text{para } z = a \\ P_b & \text{para } z = b \\ 0 & \text{en otro caso} \end{cases} \quad (\text{ec. 3.1})$$

Si $b > a$ el nivel de gris b puede aparecer como un punto de luz en la imagen, en cambio el nivel a aparece como un punto oscuro. Si P_a o P_b es cero, el ruido impulsivo es llamado unipolar. Si ninguna probabilidad es cero, y ambas son aproximadamente iguales, los valores de ruido impulsivo pueden parecer granos de sal y pimienta aleatoriamente distribuidos sobre la imagen. Por esta razón, el ruido impulsivo bipolar también es llamado ruido de sal y pimienta [González, 2002].

Una imagen con ruido gaussiano puede ser obtenida modificando los pixeles de una imagen original reemplazando aleatoriamente sus pixeles con ruido uniformemente distribuido, es más común y complejo de remover que el ruido impulsivo. El ruido aditivo gaussiano tiene una función de densidad de probabilidad dada por:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2} \quad (\text{ec. 3.2})$$

Donde z representa el nivel de gris, μ es el valor medio de z , σ es su desviación estándar y σ^2 es llamada varianza de z . Las imágenes constan de valores discretos, por lo cual en el texto se empleó el término función de distribución acumulada o probabilidad de ocurrencia en lugar de función de densidad de probabilidad.

Existen modelos para ruidos menos comunes y difíciles de eliminar como el ruido multiplicativo, el gamma, el uniforme, entre otros (Figura 3.2).



a) Original

b) Original con ruido uniforme

c) Original con ruido impulsivo

Figura 3.2 Imágenes de prueba derivadas de la imagen Peppers_mon [Pratt, 2001]

La tarea de eliminar o reducir el ruido impulsivo y gaussiano en imágenes digitales se simplifica al realizar la detección de los píxeles ruidosos y cambiar su valor por uno más adecuado, para lo cual se utilizan en primer lugar las técnicas de detección de impulsos y posteriormente un cambio de valor del píxel. Dentro de las técnicas para la detección de impulsos se encuentran los detectores estadísticos, los difusos y las Redes Neuronales Pulso-Acopladas. Por otro lado para la modificación del valor del píxel ruidoso pueden utilizarse algoritmos que actúen sobre toda la imagen de la misma manera o bien de manera selectiva como es el caso del operador de mediana, promedio, morfológico, entre otros.

3.2.3 Filtrado de ruido en imágenes digitales

Los efectos del ruido en las imágenes digitales, dependiendo del tipo que se trate, pueden ser reducidos mediante técnicas estadísticas clásicas de filtrado y la aplicación de diversos algoritmos de remoción del ruido. Por ejemplo los filtros lineales se utilizan para el procesamiento en el dominio espacial o para el procesamiento en el dominio de la frecuencia.

Una imagen filtrada espacialmente puede formarse por medio de la convolución de una imagen o arreglo de entrada con un arreglo de respuesta al impulso, donde los arreglos de entrada y salida están centrados uno con respecto al otro y el borde externo generalmente es puesto a cero. Para la remoción o filtrado del ruido, el arreglo de respuesta al impulso debe tener la forma de un filtro pasa bajas, con todos sus elementos positivos. Arreglos comunes de este tipo son de 3x3 píxeles, como los que se observan en la Figura 3.3, y son llamados máscaras de filtrado, están normalizados a pesos unitarios para no modificar la amplitud en la imagen procesada.

$$\mathbf{H} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{H} = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{H} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Figura 3.3 Máscaras de 3 x 3 pixeles, comunes para filtrado pasa bajas [Pratt, 2001]

Las técnicas no lineales tienden a producir un mejor resultado suavizando el ruido y retienen los detalles finos de la imagen. En esta categoría se encuentra el filtro de mediana, que suprime el ruido y su efecto depende del tamaño de la ventana en que es aplicado (Figura 3.4).



a) Filtro de mediana de 5x5

b) Filtro de mediana de 7x7

Figura 3.4 Filtrado de mediana de 5x5 y 7x7 para imágenes con ruido gaussiano [Pratt, 2001]

Si se considera una imagen de dimensiones MN y dentro de ella una vecindad de pixeles definida como $(2K + 1)(2K + 1)$, la cual está centrada en el pixel x_{ij} con coordenadas (i, j) en la imagen y donde X_{ij} denota a un conjunto de pixeles en el vecindario que se puede expresar como:

$$X_{ij} = \{x_{i-K, j-K}, \dots, x_{ij}, \dots, x_{i+K, j+K}\} \quad (\text{ec. 3.3})$$

El filtro de mediana tiene buenas características para eliminar ruido aditivo y ruido impulsivo, sin embargo destruye finos detalles y produce rayas y manchas. El filtro de mediana de r observaciones $x_i = 1, \dots, r$ se denota por $\text{med}(x)$, y está dado por la siguiente expresión [Varela, 2006]:

$$med(X_{ij}) = \begin{cases} x_{(\frac{r+1}{2})} & \text{para } r \text{ par} \\ \frac{1}{2} \left(x_{(\frac{r}{2})} + x_{(\frac{r+1}{2})} \right) & \text{para } r \text{ impar} \end{cases} \quad (ec. 3.4)$$

El filtro promedio trabaja como un filtro pasa bajas, para cada elemento de la imagen se calcula el promedio con los vecinos, para cada pixel el promedio se puede calcular como [Ma, 2010]:

$$prom(X_{ij}) = \frac{1}{(2K+1)^2} \sum_m^{2K+1} \sum_n^{2K+1} x_{mn} \quad (ec. 3.5)$$

El filtro morfológico también se emplea en la remoción de ruido y está basado en dos operaciones: la dilatación y la erosión.

La dilatación en escala de grises está definida como el máximo de la suma de una región local de una imagen A y una máscara B en escala de grises. Es utilizada para suavizar pequeñas regiones oscuras. La erosión en escala de grises está definida como el mínimo de la diferencia de una región local de la imagen A y una máscara en escala de grises B y se utiliza para suavizar pequeñas regiones claras.

La apertura morfológica en escala de grises está definida como la dilatación de la erosión de una imagen [Ma, 2010]:

$$Apertura(A, B) = A \circ B = (A \ominus B) \oplus B \quad (ec. 3.6)$$

La cerradura morfológica en escala de grises está definida como la erosión de la dilatación de una imagen [Ma, 2010]:

$$Cerradura(A, B) = A * B = (A \oplus B) \ominus B \quad (ec. 3.7)$$

Los filtros morfológicos en escala de grises son invariantes a la traslación, tienen propiedades de monotonía, dualidad e idempotencia y se forman a partir de la aplicación de las operaciones de apertura y cerradura en cascada, pudiendo surgir dos variantes, el filtro *Open-Close* (OC) y el *Close-Open* (CO) que están definidos como [Zhang, 2007]:

$$OC(A, B) = (A \circ B) * B \quad (ec. 3.8)$$

$$CO(A, B) = (A * B) \circ B \quad (ec. 3.9)$$

El filtro *Wiener* también llamado filtro del Promedio Mínimo Cuadrático incorpora la función de degradación y las características estadísticas del ruido, está fundado en un proceso aleatorio y persigue la estimación de la imagen original tal que el Error Cuadrático Medio sea minimizado. Su diseño deriva de la suposición de que el ruido es independiente de la imagen y que tiene media cero [Ma, 2010].

La función del Mínimo Error Cuadrático Medio en el dominio de la frecuencia puede ser expresada como [Ma, 2010]:

$$\hat{F}(u, v) = \left[\frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + K'} \right] G(u, v) \quad (ec. 3.10)$$

donde $H(u, v)$ es la función de degradación, $G(u, v)$ es la transformada de Fourier de la imagen con ruido $g(x, y)$ y K' es una constante específica determinada interactivamente para lograr el mejor comportamiento.

El procesamiento en el dominio de la frecuencia puede servir como filtro lineal del ruido, los efectos del ruido de alta frecuencia pueden ser reducidos mediante un filtrado de este tipo. El procesamiento en el dominio de la frecuencia provee la capacidad de entender de manera cuantitativa e intuitivamente la naturaleza del proceso del ruido, lo cual es útil en el diseño de filtros de ruido espaciales.

Por otro lado se encuentra el método de *Wavelets* para la remoción de ruido, este método se basa en la propiedad de que la mayor parte de la energía de una imagen transformada se concentra en los componentes de transformación más que en los de fluctuación. Los componentes de fluctuación pueden ser cuantificados sin degradar la imagen gravemente. Esta propiedad de compactación de la energía puede ser explotada para la remoción de ruido. Los componentes de la transformada *Wavelet* son umbralizados, de modo que los componentes de baja amplitud son puestos a cero.

El filtrado lineal de ruido tiene buen desempeño en imágenes con ruido continuo, tal como el ruido aditivo gaussiano uniformemente distribuido, pero tienen un efecto de suavizado grande.

3.2.4 Métricas de filtrado de imágenes digitales

El filtrado de imágenes en escala de grises puede calificarse mediante la medida de la calidad de la imagen restaurada [Zhang, 2010].

Entre las técnicas cuantitativas para la determinación de la calidad de una imagen restaurada por filtrado o compresión está el *Peak Signal-to-Noise Ratio* (PSNR), el cual se utiliza para medir la capacidad de suprimir el ruido, entre mayor sea su valor mejor es el efecto del filtrado obtenido:

$$PSNR = 10 \log_{10} \left(\frac{peakSignal^2}{MSE} \right) [dB] \quad (ec. 3.11)$$

donde *peakSignal* es el máximo nivel de gris de la imagen, generalmente tiene un valor de 255 para imágenes en gris de 8 bits.

El Error Cuadrático Medio (MSE) es la diferencia absoluta entre los pixeles de dos imágenes y se calcula por medio de la expresión:

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (D_{ij} - I_{ij})^2 \quad (ec. 3.12)$$

donde D_{ij} representa la imagen filtrada e I_{ij} a la imagen original sin ruido.

El Error Cuadrático Medio Normalizado (NMSE) es una normalización del MSE y que se expresa como:

$$NMSE = \frac{\sum_{i=1}^M \sum_{j=1}^N (O_{ij} - I_{ij})^2}{\sum_{i=1}^M \sum_{j=1}^N (O_{ij})^2} \quad (ec. 3.13)$$

El Error Absoluto Medio (MAE) que también debe ser minimizado se calcula mediante la expresión:

$$MAE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |O_{ij} - I_{ij}| \quad (ec. 3.14)$$

3.3 Redes Neuronales Pulsantes

El procesamiento de imágenes, en cualquiera de sus etapas, ha sido una ciencia por décadas. Emular una parte del cerebro, el procesador más eficiente del que se dispone, constituye un enfoque distinto a los algoritmos tradicionales y permite obtener múltiples ventajas en el campo de la Visión Artificial.

En el sistema visual de los mamíferos la imagen es captada a través de los ojos, los receptores en la retina al fondo del ojo no están uniformemente distribuidos y no todos tienen la misma función receptiva, siendo algunos receptores más sensibles al movimiento, el color o la intensidad. Después la imagen es recibida por la corteza visual, donde la información es analizada por el cerebro [Lindblad, 2005].

Existen diversos modelos neuronales pulsantes desarrollados para intentar emular y simplificar el comportamiento de la corteza visual, muchos desarrollos de fisiología aún continúan en progreso para comprender este proceso, sin embargo uno de los modelos más prometedores es el de Eckhorn, basado en la corteza visual del gato y cuyo modelo esquemático es el que se muestra en la Figura 3.5 como un conjunto de cuatro elementos que procesan las señales neuronales.

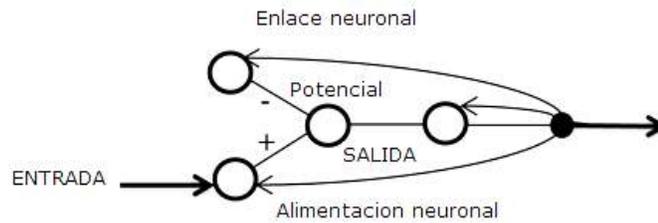


Figura 3.5 Neurona tipo Eckhorn [Lindblad, 2005]

La neurona contiene dos compartimientos de entrada: el *feeding* o alimentación y el *linking* o enlace. El *feeding* recibe un estímulo externo así como uno local, mientras que el *linking* recibe sólo un estímulo local. El *feeding* y el *linking* se combinan para crear un voltaje de membrana U_m que se compara con un umbral local dinámico Θ . Cada neurona de la red neuronal recibe retroalimentación de sus propios estímulos y también de sus vecinos, definiéndose así un radio de alimentación.

El modelo introducido se expresa por medio de las siguientes ecuaciones [Lindblad, 2005]:

$$U_{m,k}(t) = F_k(t)[1 + L_k(t)] \quad (\text{ec. 3.15})$$

$$F_k(t) = \sum_{i=1}^N [w_{ki}^f Y_i(t) + S_k(t) + N'_k(t)] \otimes I(V^a, \tau^a, t) \quad (\text{ec. 3.16})$$

$$L_k(t) = \sum_{i=1}^N [w_{ki}^l Y_i(t) + N'_k(t)] \otimes I(V^l, \tau^l, t) \quad (\text{ec. 3.17})$$

$$Y_k(t) = \begin{cases} 1 & \text{si } U_{m,k}(t) \geq \Theta_k(t) \\ 0 & \text{en otro caso} \end{cases} \quad (\text{ec. 3.18})$$

En el que $U_{m,k}$ representa el voltaje de membrana, F_k la señal de *feeding*, L_k la señal de *linking*, I es el estímulo externo, N es el número de neuronas, w son los pesos sinápticos, Y es la salida binaria, Θ es el umbral, V es una constante de ajuste y τ es una constante de tiempo. Los rangos típicos de los valores para segmentación fueron propuestos por [Lindblad, 2005] como: $\tau^a = [10,15]$, $\tau^l = [0.1, 1.0]$, $V^a = 0.5$, $V^l = [5,30]$, $V^s = [50,70]$, $\Theta_0 = [0.5,1.8]$.

3.3.1 Modelo computacional de Red Neuronal Pulso-Acoplada

El modelo original de Red Neuronal Pulsante o PCNN tiene algunas limitaciones para el procesamiento de imágenes en la práctica, por ejemplo el gran número de conexiones de retroalimentación existentes entre las neuronas y que los parámetros son difíciles de establecer, para reducir estos problemas Ranganath y Kuntimad en 1995 obtuvieron un modelo simplificado que puede aplicarse al procesamiento de imágenes.

El modelo de PCNN modificado para efectos de simulación se expresa como [Ma, 2010]:

$$U_{ij}[n] = F_{ij}[n](1 + \beta L_{ij}[n]) \quad (ec. 3.19)$$

$$F_{ij}[n] = e^{-\alpha_F} F_{ij}[n-1] + V_F \sum_{kl} M_{ijkl} Y_{kl}[n-1] + S_{ij} \quad (ec. 3.20)$$

$$L_{ij}(t) = e^{-\alpha_L} L_{ij}[n-1] + V_L \sum_{kl} W_{ijkl} Y_{kl}[n-1] \quad (ec. 3.21)$$

$$Y_{ij}[n] = \begin{cases} 1 & \text{si } U_{ij}[n] > E_{ij}[n] \\ 0 & \text{en otro caso} \end{cases} \quad (ec. 3.22)$$

$$E_{ij}[n] = E_{ij}[n-1]e^{-\alpha_E} + V_E Y_{ij}[n-1] \quad (ec. 3.23)$$

donde α_F , α_L , y α_E son constantes de tiempo; V_F , V_L , y V_E son constantes de ajuste y β , es la fuerza de relación de la PCNN.

Cada neurona es denotada con los índices (i, j) , y una de sus neuronas vecinas con (k, l) . *Feeding* y *linking* combinados originan la actividad interna U . La neurona recibe la señal de entrada S por el *feeding* y la sinapsis M , y en teoría, cada neurona está conectada con sus vecinas de modo tal que la señal de salida de una neurona modula la actividad de sus vecinas por medio del *linking* y la sinapsis W . El pulso de salida es capaz de retroalimentar y modular el umbral E por medio de un filtro integrador, incrementando el umbral por medio de la magnitud de V_E que decrece con la constante de tiempo α_E .

De este modo se pueden distinguir tres partes del modelo, la receptiva, la modulación y la generación del pulso como se observa en la Figura 3.6.

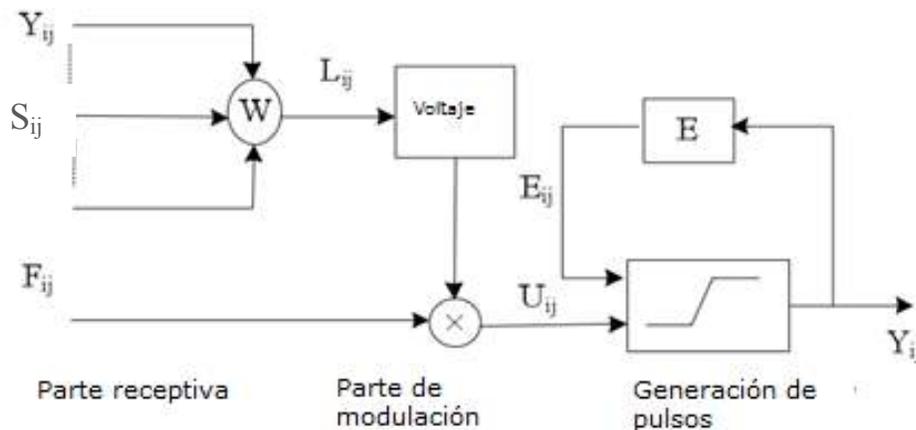


Figura 3.6 Modelo de PCNN [Zhang, 2007]

Cuando existe más de un estímulo por neurona surge la PCNN multi-canal, como en el caso de la detección de movimiento.

3.3.2 Modelo de Red Neuronal Pulso-Acoplada tipo ICM

Otros modelos modificados de PCNN que se pueden encontrar en la literatura son el *Intersection Cortical Model* (ICM), que es un caso especial donde no hay *linking* por lo que β es igual a cero y sólo existe el *feeding*.

El modelo simplificado *Intersection Cortical Model* está definido por el siguiente conjunto de ecuaciones [Ma, 2010]:

$$F_{ij}[n] = fF_{ij}[n-1] + \sum_{kj} W_{ijkl} Y_{kl}[n-1] + S_{ij} \quad (ec. 3.24)$$

$$E_{ij}[n] = gE_{ij}[n-1] + hY_{ij}[n-1] \quad (ec. 3.25)$$

$$Y_{ij}[n] = \begin{cases} 1 & \text{si } F_{ij}[n] > E_{ij}[n] \\ 0 & \text{en otro caso} \end{cases} \quad (ec. 3.26)$$

donde n es el número de iteración, f , g y h son constantes de ajuste.

3.3.3 Modelo de Red Neuronal Pulso-Acoplada tipo SCM

En el *Spiking Cortical Model* (SCM) las neuronas reciben la señal de entrada por medio de las sinapsis del *feeding*, y cada neurona está conectada con sus vecinas de modo tal que la señal de salida de cada neurona modula la actividad de sus vecinas por medio de las sinapsis del *linking*. El modelo está especificado por medio de las siguientes ecuaciones [Ma, 2010]:

$$U_{ij}[n] = fU_{ij}[n-1] + S_{ij} \sum_{kj} W_{ijkl} Y_{kl}[n-1] + S_{ij} \quad (ec. 3.27)$$

$$E_{ij}[n] = gE_{ij}[n-1] + hY_{ij}[n-1] \quad (ec. 3.28)$$

$$Y_{ij}[n] = \begin{cases} 1 & \text{si } 1/(1 + \exp(-\gamma(U_{ij}[n] - E_{ij}[n]))) > 0.5 \\ 0 & \text{en otro caso} \end{cases} \quad (ec. 3.29)$$

donde n es el número de iteración, f , g , h y γ son constantes de ajuste.

3.3.4 Modelo de Red Neuronal Pulso-Acoplada tipo PCNNNI

Otro modelo de PCNN modificado es el de Interconexión Nula o PCNNNI, el cual se apoya en la Matriz de Tiempos T_{ij} donde se guarda el número de iteración en el que se activa una neurona por primera vez, las expresiones para esta red son las que se colocan a continuación [Ma, 2010].

$$F_{ij}[n] = e^{-\alpha^F} F_{ij}[n-1] + V_F Y_{kl}[n-1] + S_{ij} \quad (\text{ec. 3.30})$$

$$U_{ij}[n] = F_{ij}[n] \quad (\text{ec. 3.31})$$

$$E_{ij}[n] = e^{-\alpha^E} E_{ij}[n-1] + V_E Y_{ij}[n-1] \quad (\text{ec. 3.32})$$

$$Y_{ij}[n] = \begin{cases} 1 & \text{si } U_{ij}[n] > E_{ij}[n] \\ 0 & \text{en otro caso} \end{cases} \quad (\text{ec. 3.33})$$

$$T_{ij}[n] = \begin{cases} n & \text{si } Y_{ij}[n] = 1, \\ T_{ij}[n-1] & \text{en otro caso} \end{cases} \quad (\text{ec. 3.34})$$

donde n es el número de iteración, α^F , V_F , α^E y V_E son constantes de ajuste.

3.3.5 Algoritmos de filtrado mediante Redes Neuronales Pulso-Acopladas

El filtrado de ruido en imagen digitales mediante las Redes Neuronales Pulso-Acopladas es posible debido a que al procesar una imagen contaminada con ruido impulsivo los pixeles ruidosos activarán la salida de la red antes o después que sus vecinos, lo cual permite localizarlos y filtrarlos posteriormente con alguna técnica elegida para este propósito. En el caso del ruido gaussiano se puede observar la secuencia de activación de los pixeles y elegir una forma de filtrado en base a esta información.

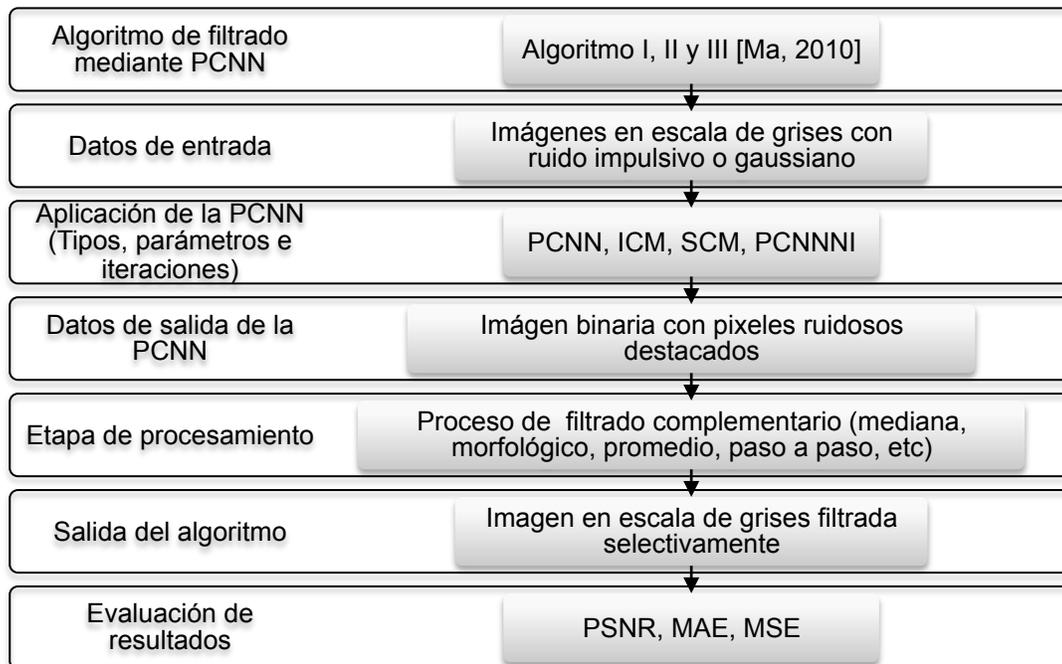


Figura 3.7 Esquema de los algoritmos de filtrado de ruido mediante PCNN

Se han propuesto diversas técnicas para llevar a cabo el filtrado de imágenes por medio de este tipo de redes, algunos de los trabajos reportados optan por implementar la PCNN o alguna de sus variantes como la SCM, la ICM o la PCNNNI en combinación con otras técnicas de filtrado y de Inteligencia Artificial, no obstante dichas técnicas comparten algoritmos similares como se observa en la Figura 3.7.

Se seleccionaron inicialmente algoritmos de filtrado de ruido impulsivo y gaussiano basados en [Ma, 2010], donde se describen y esquematizan dos técnicas de filtrado del ruido impulsivo, el Algoritmo I (Figura 2.4) consiste en aplicar la red tipo PCNN a la imagen ruidosa, y con base en los pixeles activados en la imagen de salida se aplica una modificación paso a paso del valor de los pixeles correspondientes, de modo que todos ellos puedan activar sus salidas al mismo tiempo. El Algoritmo II (Figura 2.5) consiste de modo similar en aplicar el filtro de mediana o el filtro morfológico sobre la localidad donde la matriz de salida de la PCNN se haya activado.

Las redes PCNN no requieren entrenamiento, su única función es clasificar los pixeles por niveles de intensidad, en estas redes cada pixel de la imagen corresponde a una neurona. Tienen una fuerte ventaja en el suavizado de imágenes y han sido utilizadas en combinación con el filtro de mediana, el morfológico y el filtro promedio con resultados superiores a diversas técnicas tradicionales de filtrado [Ma, 2010].

De esta manera el modelo de las redes PCNN es un sistema biónico que emula las neuronas de la corteza visual de los mamíferos y ha sido aplicado en variedad de dominios del procesamiento de imágenes tales como la remoción de ruido, la detección de objetos, la optimización, el adelgazamiento de imágenes, la segmentación y la remoción de sombras, por lo que ha atraído la atención de muchos investigadores del área de la Visión Artificial [Shao-Fa, 2010].

3.4 Detectores de impulsos

Con el fin de localizar los posibles pixeles con ruido en una imagen y evitar la pérdida de información como bordes, líneas y otros detalles se pueden utilizar diversas técnicas, una de ellas es el llamado *Enhanced Rank Impulse Detector* (ERID) [Aizenberg, 2003], el cual analiza el pixel de interés bajo el siguiente criterio: en primer lugar se determina si el pixel se encuentra en los extremos del ordenamiento ascendente de los pixeles en la ventana de filtrado, de ser así puede ser que tenga ruido impulsivo, en segundo lugar se determina qué tan alejado está el pixel de la mediana de la ventana de filtrado, para lo cual se define un umbral, si el pixel está alejado de la mediana superando del umbral definido podría tratarse de un pixel con ruido o bien un detalle o contorno. Si ambos criterios se cumplen se considera que el pixel está corrompido por ruido impulsivo.

El detector de impulsos está definido como [Varela, 2006]:

$$[(\text{rango}(x_{ij}) \leq s) \vee (\text{rango}(x_{ij}) \geq r - s)] \wedge [|x_{ij} - \text{med}(X_{ij})| \geq U_2] \quad (\text{ec. 3.35})$$

donde x_{ij} es el pixel central de la ventana de filtrado X_{ij} , $\text{rango}(x_{ij})$ es la posición del pixel central en el ordenamiento ascendente de los pixeles de la ventana de filtrado, s es un umbral que define los valores extremos, $\text{med}(X_{ij})$ es la mediana de la ventana de filtrado, U_2 es el umbral que define la distancia de x_{ij} con la mediana de la ventana de filtrado y r es el número de pixeles en la venta de filtrado.

También se han propuesto otras técnicas para localizar pixeles con ruido impulsivo como el Detector Difuso (FIDRM), el cual se basa en el establecimiento de gradientes y reglas difusos. Si la mayoría de los ocho gradientes en la ventana de filtrados son grandes entonces el pixel central es un pixel con ruido impulsivo. Con la información de los pixeles con ruido se construye un histograma del ruido, si el histograma contiene picos, se concluye que la imagen contiene pixeles con ruido impulsivo, en caso contrario se concluye que la imagen está libre de ruido [Schulte, 2006].

3.5 Discusión

Los métodos de filtrado más recientes usualmente incluyen dos pasos, la detección del ruido y la remoción del mismo. Algunas técnicas de filtrado pueden ser completadas utilizando las propiedades de las Redes Neuronales y las Redes Neuronales Pulsantes ya que tienen la capacidad de destacar los pixeles ruidosos por su nivel de gris que difiere del de sus vecinos, lo cual posibilita su empleo para la detección de pixeles correspondientes a ruido impulsivo y gaussiano en un número limitado de iteraciones de la red. Mediante la presente investigación se pretende visualizar el modo en que esto puede ser llevado a cabo de modo eficiente.

Capítulo 4 Implementación

En el presente capítulo se hacen los preparativos para llevar a cabo la experimentación, comenzando con la simulación del ruido impulsivo y gaussiano generando las imágenes de prueba. Se estudian los detalles de la implementación de los filtros tradicionales, de los algoritmos de filtrado basados en Redes Neuronales Pulso-Acopladas elegidos para su reimplementación y también de las métricas empleadas para cuantificar los resultados.

4.1 Requerimientos del sistema

Se detectaron las necesidades de experimentación con los diversos tipos de ruido, filtros, algoritmos con redes PCNN, métricas e imágenes, por lo que se propuso la creación de una interfaz gráfica que agilizó el procesamiento de imágenes y la generación de datos.

R1. Se elaboró una plataforma de experimentación con interfaz gráfica, de código abierto, en lenguaje C.

R2. Se lograron manipular imágenes de hasta 512x512 pixeles en formato bmp, a color o en escala de grises.

R3. Se implementó la simulación de ruido impulsivo bipolar de probabilidad de ocurrencia de entre 0 y 0.9 con incrementos de 0.1; se especificó la generación de hasta 10 imágenes distintas con ruido a partir de una imagen original.

R4. Se implementó la simulación de ruido gaussiano de $\mu = 0$, con $\sigma^2 = [0,0.3]$ con incrementos de 0.001; se especificó la generación de hasta 10 imágenes distintas con ruido a partir de una imagen original.

R5. Se implementó el filtro de mediana de 3x3, 5x5 y 7x7 pixeles.

R6. Se implementó el filtro promedio de 3x3.

R7. Se implementó el filtro morfológico con las operaciones de dilatación, erosión, apertura, cerradura y Open-Closed.

R8. Se implementó el procesamiento de imágenes mediante la red PCNN, con variación del número de iteraciones $[0 - 10]$, $M_{ijkl} = W_{ijkl} = [0.5 \ 1 \ 0.5; \ 1 \ 0 \ 1; \ 0.5 \ 1 \ 0.5]$, con parámetros variables de $\alpha_F = [0.1 - 1.0]$, $\alpha_L = [0.1 - 1.0]$, $\alpha_E = [0.1 - 1.0]$, $V_L = [0.1, 1.0]$, $V_E = [5.0 - 20.0]$, $V_F = [0.1 - 1.0]$, $\beta = [0.1 - 1.0]$.

R9. Se implementó el procesamiento de imágenes mediante la red ICM, con variación del número de iteraciones $[0 - 10]$, $W_{ijkl} = [0.5 \ 1 \ 0.5; \ 1 \ 0 \ 1; \ 0.5 \ 1 \ 0.5]$, con parámetros variables de $f = [0.1 - 1.0]$; $g = [0.1 - 1.0]$; $h = [1 - 20]$.

R10. Se implementó el procesamiento de imágenes mediante la red ICM, con variación del número de iteraciones $[0 - 10]$, $W_{ijkl} = [0.5 \ 1 \ 0.5; \ 1 \ 0 \ 1; \ 0.5 \ 1 \ 0.5]$ y parámetros variables $f = [0.1 - 1.0]$; $g = [0.1 - 1.0]$; $h = [1.0 - 20.0]$; $\gamma = [0.1 - 1.0]$.

R11. Se implementó el procesamiento de imágenes mediante la red PCNNNI con parámetros variables $\alpha_F = [0.1 - 1.0]$; $v_F = [0.1 - 1.0]$; $\alpha_E = [0.1, 1.0]$; $v_E = [0.1 - 5.0]$.

R12. Se implementó el algoritmo de filtrado mediante la red ICM (Figura 2.4) combinada con el filtro de mediana.

R13. Se implementó el algoritmo de filtrado mediante la red SCM (Figura 2.4) combinada con el filtro de mediana.

R14. Se implementó el algoritmo de filtrado mediante la red PCNNNI (Figura 2.5) combinada con un filtrado selectivo.

R15. Se implementó el cálculo de la calidad de las imágenes mediante el PSNR, MAE, NMSE, MSE y el tiempo de procesamiento.

R16. Se dispuso de un equipo de cómputo de escritorio con procesador Intel Pentium de 3GHz, 2GB de RAM, con sistema operativo Windows 7 Starter de 32 bits.

R17. Se realizó la codificación mediante Code Blocks 10.8, wxwidgets 2.8.1 y MINGW3.6.

4.2 Casos de uso del sistema

Se presenta el diagrama del caso de uso principal del sistema PCNN (Figura 4.1), en éste se observa que hay cinco acciones principales y tres complementarias que puede realizar el usuario, se describe brevemente cada una.

4.2.1 Seleccionar imagen original

Consiste en la selección de una imagen por el usuario; es necesario tener imágenes en el equipo o elemento de almacenamiento. La acción comienza con la selección de una imagen mediante una ventana de búsqueda, en caso de que no se seleccione alguna o que la imagen seleccionada no tenga formato bmp y RGB o de que su tamaño sea mayor a 512x 512 pixeles se genera un mensaje de error; en caso de que la imagen sea compatible se convierte a escala de grises, una copia en miniatura, su nombre y dimensiones se despliegan en la interfaz. Posteriormente se selecciona el tipo y rango de ruido deseado.

4.2.2 Simular ruido

Consiste en la introducción de ruido en una imagen por el usuario; es necesario haber seleccionado una imagen original. La acción comienza seleccionando el tipo de ruido, en caso de que sea impulsivo se elige luego la probabilidad; la acción se aplica al presionar el botón y concluye cuando aparece la imagen con ruido en la interfaz y una copia de la imagen se guarda en el equipo. Posteriormente la imagen se filtra o se procesa.

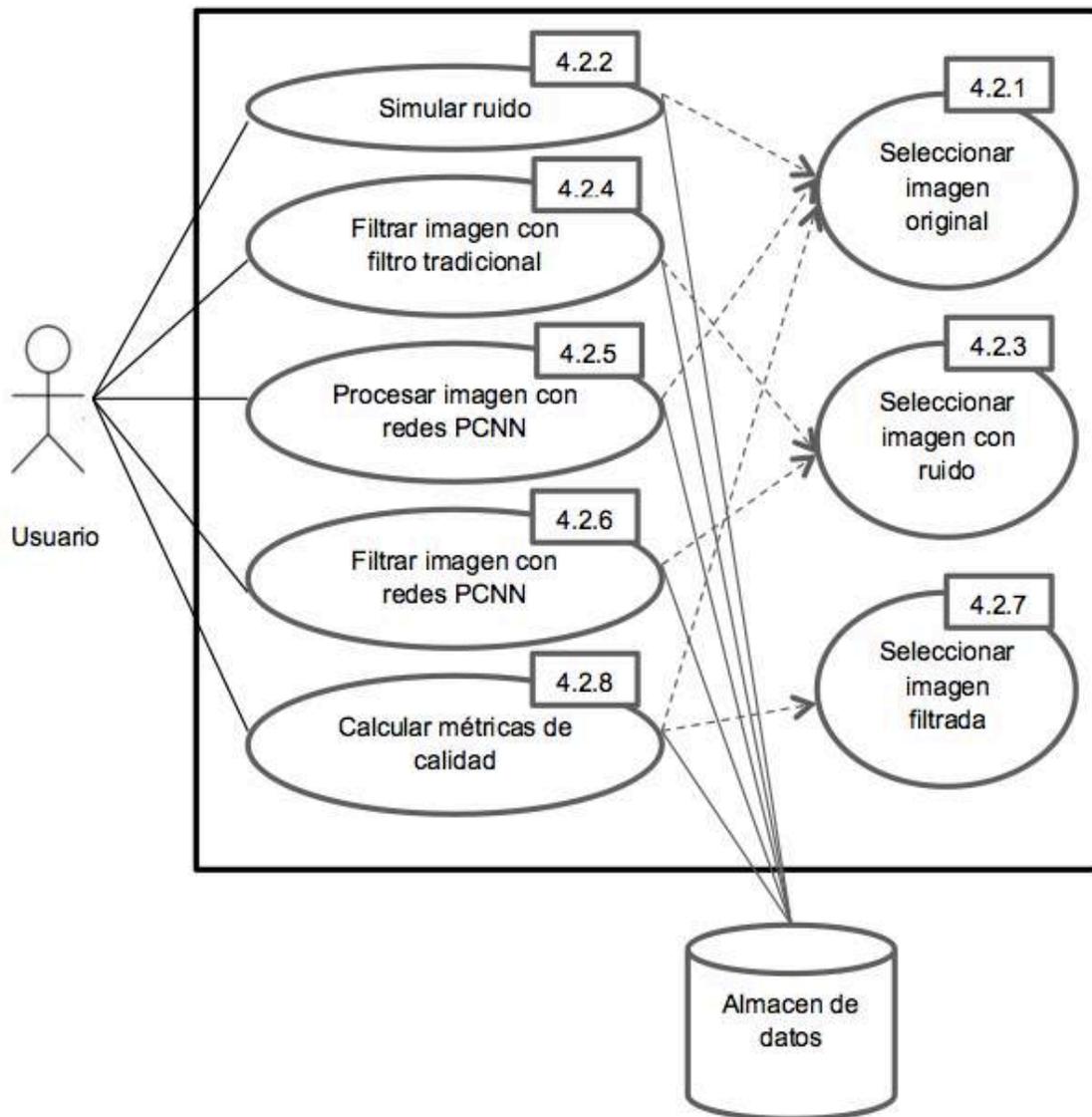


Figura 4.1 Diagrama de casos de uso del sistema PCNN

4.2.3 Seleccionar imagen con ruido

Consiste en la selección de una imagen con ruido por el usuario; es necesario tener imágenes con ruido en el equipo o elemento de almacenamiento. La acción comienza con la selección de una imagen mediante una ventana de búsqueda, en caso de que no se seleccione alguna o que la imagen seleccionada no tenga formato bmp y no esté en escala de grises o de que su tamaño sea mayor a 512 x 512 píxeles se genera un mensaje de error; en caso de que la imagen sea compatible una copia en miniatura se despliegan en la interfaz. Posteriormente se selecciona el filtro o el procesamiento con PCNN que se aplicará.

4.2.4 Filtrar imagen con filtro tradicional

Consiste en la aplicación del filtro de mediana, promedio o morfológico según seleccione el usuario; es necesario haber simulado ruido en una imagen o seleccionado una imagen con ruido. La acción comienza seleccionando el tipo de filtro que se va a aplicar; se puede utilizar el filtro de mediana, el filtro promedio o el filtro morfológico seleccionando una opción y luego el botón aplicar. La acción termina cuando se muestra la imagen filtrada en la interfaz, una copia se guarda en el equipo y aparece un mensaje de confirmación. Posteriormente se calculan las métricas de calidad.

4.2.5 Procesar la imagen con PCNN

Consiste en la generación de imágenes y datos de salida de la red PCNN seleccionada por el usuario; es necesario haber seleccionado una imagen con ruido o haber introducido ruido en una imagen. La acción comienza seleccionando el tipo de red que se desea utilizar, en cada caso se ajustan los parámetros y luego se presiona el botón aplicar. La acción termina cuando se muestra en pantalla una imagen binaria correspondiente a la última iteración de la red, se guarda una imagen por iteración y aparece un mensaje de confirmación. Posteriormente se selecciona otra imagen, aplicar un filtro o volver a procesar la imagen.

4.2.6 Filtrar imagen con red PCNN

Consiste en la aplicación de un algoritmo de filtrado con PCNN según seleccione el usuario; es necesario haber simulado ruido en una imagen o haber seleccionado una imagen con ruido. La acción comienza seleccionando el tipo de red que se va a emplear que puede ser la red ICM, SCM o PCNNNI ajustando parámetros y seleccionando el botón de aplicar. La acción termina cuando se muestra la imagen filtrada en la interfaz, una copia se guarda y aparece un mensaje de confirmación. Posteriormente se calcula la métrica de calidad.

4.2.7 Seleccionar imagen filtrada

Consiste en la selección de una imagen filtrada por parte del usuario; es necesario tener imágenes filtradas en el equipo o elemento de almacenamiento. La acción comienza con la selección de una imagen mediante una ventana de búsqueda, en caso de que no cumpla los criterios se genera un mensaje de error en caso contrario una copia en miniatura se despliegan en la interfaz. Posteriormente se selecciona la métrica a aplicar siempre y cuando se haya seleccionado una imagen original.

4.2.8 Calcular métricas de calidad

Consiste en determinar los valores de PSNR, MAE, NMSE, MSE y tiempo de procesamiento por el usuario; es necesario haber seleccionado una imagen original, haber agregado ruido a una imagen y haber filtrado una imagen o haber seleccionado alguna imagen filtrada.

La acción comienza al presionar el botón aplicar, en el caso de que se presione para la imagen con ruido aparecerán los datos en la parte inferior de la imagen procesada; para la imagen filtrada los datos aparecerán bajo la imagen correspondiente. La acción termina cuando los valores son desplegados en los campos de la interfaz y se genera un archivo xls con los datos formados.

4.3 Diseño de interfaz de la plataforma de experimentación

Se desarrolló una interfaz gráfica de usuario en lenguaje C (Figura 4.2) y un manual de instalación de los componentes necesarios para ejecutarla. Los comandos y opciones de la interfaz se colocaron en inglés con el fin de dar mayor portabilidad al sistema.



Figura 4.2 Interfaz de la plataforma de experimentación

La plataforma de experimentación desarrollada se compone de la siguiente manera:

- R18. Generación de imágenes con ruido y selección del nivel de ruido de una lista.
- R19. Filtrado tradicional con selección de ventanas de filtrado de una lista.
- R20. Filtros basados en PCNN para tres opciones con ajuste de parámetros de una lista.
- R21. Imagen original, se carga para aplicarle ruido o para adquirir métricas.

R22. Imagen ruidosa, se despliega el resultado del proceso de adición de ruido o se carga una ya contaminada, así mismo se calculan las métricas.

R23. Imagen filtrada, aquí se muestra la imagen procesada, se carga una o se calculan las métricas.

R24. Características, permite la visualización de los parámetros indicados.

4.4 Codificación

Se presenta el pseudocódigo asignado a cada opción de la interfaz relacionada con el filtrado de imágenes mediante los distintos tipos de red PCNN.

En la Tabla 4.1 se presenta el algoritmo de procesamiento de imágenes con la red PCNN original, donde se observa que son siete los parámetros a ajustar, se tiene la matriz de pesos sinápticos y las señales de alimentación y enlace combinadas para formar el potencial interno de la matriz, el principal producto es el conjunto de matrices binarias producidas en cada iteración, el número de iteraciones no está limitado por ningún parámetro.

Tabla 4.1 Algoritmo de procesamiento de una imagen con la red PCNN

Algoritmo de la red PCNN

Paso 1. Inicializar los valores de las constantes de tiempo α_F , α_L , y α_E , así como de las constantes de ajuste V_F , V_L , V_E y β .

Paso 2. Inicializar los valores de las matrices Y_{ij} , L_{ij} , E_{ij} , F_{ij} y U_{ij} del tamaño de la imagen con ruido (fil, col) con todos sus elementos en 0, y los de la matriz $M_{ijkl} = W_{ijkl}$.

Paso 3. Normalizar entre 0 y 1 los valores de gris de la imagen original con ruido O_{ij} y asignar los valores a S_{ij} .

Paso 4. Calcular el valor de las matrices F_{ij} , L_{ij} , U_{ij} , E_{ij} , Y_{ij} de acuerdo a las ecuaciones 3.20, 3.21, 3.19, 3.22 y 3.23 en ese orden, actualizando sus valores anteriores durante n iteraciones.

En la Tabla 4.2 se observa el pseudocódigo de procesamiento con la red PCNN más simple, la red ICM, que no requiere señal de enlace pues obtiene información de las neuronas vecinas por medio de la matriz de pesos sinápticos, sólo tiene tres parámetros para ajustar y tres matrices cuyo valor se actualiza con cada iteración, la salida es también una matriz binaria que es utilizada, en este caso, para detectar el ruido.

Tabla 4.2 Algoritmo de detección de ruido impulsivo mediante la red ICM

Algoritmo de la red ICM

Paso 1. Inicializar los valores de las constantes de ajuste f , g y h .

Paso 2. Inicializar los valores de las matrices Y_{ij} , E_{ij} y F_{ij} del tamaño de la imagen con ruido (fil, col) con todos sus elementos en 0, y los de la matriz W_{ijkl} .

Paso 3. Normalizar entre 0 y 1 los valores de gris de la imagen original con ruido O_{ij} y asignar los valores a S_{ij} .

Paso 4. Calcular el valor de las matrices F_{ij} , E_{ij} , Y_{ij} de acuerdo a las ecuaciones 3.24, 3.25 y 3.26 en ese orden, actualizando sus valores anteriores durante n iteraciones.

En la Tabla 4.3 se encuentra el pseudocódigo de la red SCM la cual combina las señales de alimentación y enlace en la señal de potencial interno de membrana, utiliza una expresión más compleja para el cálculo del umbral y tiene cuatro parámetros que ajustar.

Tabla 4.3 Algoritmo de detección de ruido impulsivo mediante la red SCM

Algoritmo de la red SCM

Paso 1. Inicializar los valores de las constantes de ajuste f , g , h y γ .

Paso 2. Inicializar los valores de las matrices Y_{ij} , E_{ij} y U_{ij} del tamaño de la imagen con ruido (fil, col) con todos sus elementos en 0, y los de la matriz W_{ijkl} .

Paso 3. Normalizar entre 0 y 1 los valores de gris de la imagen original con ruido O_{ij} y asignar los valores a S_{ij} .

Paso 4. Calcular el valor de las matrices U_{ij} , E_{ij} , Y_{ij} de acuerdo a las ecuaciones 3.27, 3.28 y 3.29 en ese orden, actualizando sus valores anteriores durante n iteraciones.

En la Tabla 4.4 se encuentra el pseudocódigo para la eliminación del ruido impulsivo que ha sido detectado mediante alguna red PCNN, puede ser reemplazado por alguna otra técnica de filtrado selectiva y no considera los bordes en el filtrado.

Tabla 4.4 Algoritmo de eliminación de ruido detectado mediante el operador de mediana

Algoritmo de eliminación de ruido

Paso 1. Recorrer la matriz $Y(i, j)$ desde $i = 2$ y hasta $i = fil - 1$, y de $j = 2$ y hasta $j = col - 1$. En caso de encontrar un elemento igual a 0 hacer $D(i, j) = O(i, j)$, si el elemento es 1 y por lo menos 5 de sus vecinos no están activos se procede a aplicar el operador de mediana definido por la ecuación 3.4 con $r = 9$ sobre la imagen con ruido $O(i, j)$ en la posición (i, j) correspondiente y asignar su valor a la imagen filtrada $D(i, j)$.

En la Tabla 4.5 se encuentra el pseudocódigo de la red PCNNNI que se utilizó para el filtrado de ruido gaussiano, inicia con cambiando el rango dinámico de la imagen para asegurar la convergencia del cálculo de la matriz de tiempos en menos de 10 iteraciones, luego se procesa la imagen mediante la red y finalmente se hace un filtrado selectivo.

Tabla 4.5 Algoritmo de procesamiento de una imagen con la red PCNNNI

Algoritmo de filtrado con la red PCNNNI

Cambio de rango dinámico de la imagen:

Paso 1. Definir los umbrales inferior E_0 y superior L entre 0 y 255.

Paso 2. Si el valor de un pixel (i, j) de la imagen con ruido $O(i, j)$ es menor que E_0 , se asigna el valor de E_0 al pixel correspondiente en la nueva imagen $J(i, j)$, en caso contrario y si el valor del pixel es mayor que $L - E_0$ se asigna el valor $L - E_0$ a la imagen $J(i, j)$ en el pixel correspondiente, en caso de que el valor del pixel no exceda la diferencia se hace $J(i, j) = O(i, j)$.

Procesamiento con la red PCNNNI:

Paso 3. Inicializar los valores de las constantes de ajuste α^F, V_F, α^E y V_E .

Paso 4. Inicializar los valores de las matrices $Y_{ij}, E_{ij}, F_{ij}, U_{ij}$ y T_{ij} del tamaño de la imagen con ruido modificada $J(i, j)$ con todos sus elementos en 0.

Paso 5. Normalizar entre 0 y 1 los valores de gris de la imagen con ruido $J(i, j)$ y asignar los valores a S_{ij} .

Paso 6. Calcular el valor de las matrices $F_{ij}, U_{ij}, E_{ij}, Y_{ij}$ y T_{ij} de acuerdo a las ecuaciones 3.30, 3.31, 3.32, 3.33 y 3.34 en ese orden actualizando sus valores.

Paso 7. Continuar iterando desde el Paso 6 hasta que todos los pixeles de T_{ij} sean distintos de 0.

Filtrado de la imagen:

Paso 8. Recorrer la matriz $T(i, j)$ desde $i = 2$ y hasta $i = fil - 1$, y de $j = 2$ y hasta $j = col - 1$ con una ventana K de 3x3 centrada en el pixel (i, j) , si todos los elementos en K son iguales se aplica el operador promedio dado en la ecuación 3.5 y se asigna el valor a la imagen filtrada $D(i, j)$. Si el valor máximo en K es igual al elemento $T(i, j)$ el valor $D(i, j)$ es igual que el valor $J(i, j)$ mas un incremento Δ , si $Min(K) = T(i, j)$ entonces $D(i, j) = J(i, j) - \Delta$. En otro caso y si $K(2,2) = T(i, j)$ se hace $D(i, j) = J(i, j)$, sino es así se aplica el operador de mediana dado en la ecuación 3.4 sobre esta posición (i, j) y se asigna a $D(i, j)$.

Paso 9. Repetir el paso 8 hasta que todos los elementos en $J(i, j)$ hayan sido procesados.

4.5 Plan de pruebas del sistema

Después de que se implementaron los algoritmos de procesamiento se estableció un conjunto de pruebas para validar la plataforma de experimentación y así continuar con la realización de experimentos con distintos niveles de ruido, variación de los parámetros del modelo y con imágenes de diversos tipos.

Para cada caso de prueba se tiene una descripción que lo identifica con los resultados obtenidos, se mencionan las características de la prueba, los criterios para darle comienzo y fin, así como comentarios de las dificultades encontradas, cómo se superaron y otros comentarios adicionales (Tabla 4.6-4.9).

Tabla 4.6 Plan de pruebas del sistema prueba 4.5.1

Descripción del objeto de prueba	Características	Criterios para comienzo	Criterios para fin de pruebas	Comentarios
4.5.1 Generación de imágenes de prueba con ruido impulsivo	A partir de 5 imágenes se generaron 50, donde de cada imagen se obtuvieron 10 con nivel de ruido impulsivo de probabilidad de ocurrencia de 0 a 0.9	5 imágenes de 512x512 pixeles de diversas bases de datos Casos de prueba del sistema 4.2.1 y 4.2.2	50 imágenes de 512x512 con ruido impulsivo guardadas en el equipo en formato bmp en escala de grises.	Las imágenes fueron comparadas visualmente con las obtenidas mediante Matlab.

Tabla 4.7 Plan de pruebas del sistema pruebas 4.5.2 a 4.5.6

Descripción del objeto de prueba	Características	Criterios para comienzo	Criterios para fin de pruebas	Comentarios
4.5.2 Generación de imágenes de prueba con ruido gaussiano	Se generaron 34 imágenes con ruido gaussiano que se obtuvieron mediante Matlab aumentando la varianza del ruido de 0 a 0.3 con incrementos de 0.01, utilizando media de 0.	5 imágenes de 512x512 pixeles de diversas bases de datos Casos de uso del sistema 4.2.1 y 4.2.2	34 imágenes de 512x512 con ruido gaussiano guardadas en el equipo en formato bmp en escala de grises.	No se logró igualar el efecto del ruido gaussiano con las varianzas especificadas, se utilizaron las imágenes generadas en Matlab.
4.5.3 Generación de imágenes filtradas con el filtro de mediana	Aplicar el filtro de mediana de 3x3, 5x5 y 7x7 a imágenes generadas en el sistema o imágenes precargadas.	50 Imágenes con ruido impulsivo generadas en el sistema y 34 imágenes con ruido gaussiano generadas mediante Matlab. Casos de uso del sistema 4.2.3 y 4.2.4	50 imágenes con ruido impulsivo filtradas con mediana. 34 imágenes con ruido gaussiano filtradas con mediana.	Las imágenes con ruido gaussiano filtradas mediante mediana sólo fueron las de <i>Lena</i> , <i>Peppers</i> y <i>Baboon</i> .
4.5.4 Generación de imágenes filtradas con el filtro promedio	Aplicar el filtro promedio de 3x3 a imágenes generadas en el sistema o imágenes precargadas.	50 imágenes con ruido impulsivo de probabilidad de ocurrencia de entre 0 y 0.9 generadas en el sistema de experimentación. Casos de uso del sistema 4.2.3 y 4.2.4	50 imágenes con ruido impulsivo filtradas con filtro promedio	Este filtro sólo se aplicó para imágenes con ruido impulsivo.
4.5.5 Generación de imágenes filtradas con el filtro morfológico	Aplicar el filtro morfológico a imágenes generadas en el sistema o imágenes precargadas.	50 imágenes con ruido impulsivo generadas en el sistema a partir de 5 imágenes base. Casos de uso del sistema 4.2.3 y 4.2.4	10 imágenes filtradas con el filtro morfológico Open-Closed 10 imágenes con operaciones de dilatación, erosión, apertura y cerradura.	Se descartó este método de filtrado por su baja efectividad.
4.5.6 Generación de imágenes filtradas con el filtro Wiener	Aplicar el filtro Wiener a imágenes generadas en el sistema o imágenes precargadas.	34 imágenes con ruido gaussiano de <i>Lena</i> , <i>Peppers</i> y <i>Baboon</i> generadas mediante Matlab.	34 imágenes con ruido gaussiano filtradas con el filtro Wiener mediante Matlab.	No se logró igualar el filtro Wiener mediante el sistema por lo que se empleó Matlab.

Tabla 4.8 Plan de pruebas del sistema pruebas 4.5.7 a 4.5.12

Descripción del objeto de prueba	Características	Criterios para comienzo	Criterios para fin de pruebas	Comentarios
4.5.7 Métrica PSNR implementada	Mediciones de PSNR para las imágenes con ruido y filtradas	Imágenes con y sin ruido precargadas en el equipo. Caso de uso del sistema 4.2.1, 4.2.3, 4.2.7, 4.2.8	6 gráficas de medición de PSNR para imágenes con ruido impulsivo, con ruido gaussiano y filtradas	Los resultados se compararon con las funciones de Matlab obteniéndose resultados similares.
4.5.8 Métrica MAE implementada	Mediciones de MAE para las imágenes con y sin ruido	Imágenes con ruido y sin ruido precargadas en el equipo. Caso de uso del sistema 4.2.1, 4.2.3, 4.2.7, 4.2.8	2 gráficas de medición de MAE para imágenes con ruido impulsivo y con ruido gaussiano filtradas	Los resultados se compararon con las funciones de Matlab obteniéndose resultados similares.
4.5.9 Métrica MSE implementada	Mediciones de MSE para las imágenes con ruido y filtradas	Imágenes con ruido y sin ruido precargadas en el equipo. Caso de uso del sistema 4.2.1, 4.2.3, 4.2.7, 4.2.8	2 gráficas de medición de MSE para imágenes con ruido impulsivo, con ruido gaussiano filtradas	Esta métrica fue descartada por redundancia con el NMSE.
4.5.10 Métrica NMSE implementada	Mediciones de NMSE para las imágenes con ruido y filtradas	Imágenes con ruido y sin ruido precargadas en el equipo. Caso de uso del sistema 4.2.1, 4.2.3, 4.2.7, 4.2.8	2 gráficas de medición de MSE para imágenes sin ruido, con ruido impulsivo, con ruido gaussiano y filtradas	Los resultados se compararon con los reportados en la literatura obteniéndose resultados similares.
4.5.11 Medición del tiempo de procesamiento	Medición del tiempo de procesamiento de los filtros y el procesamiento con PCNN	Imágenes de prueba. Casos de uso del sistema 4.2.4, 4.2.5, 4.2.6, 4.2.8 Algoritmo de la Tabla 4.1	3 gráficas de medición del tiempo de procesamiento	Se tomaron promedios del tiempo para cada técnica.
4.5.12 Implementación de la red PCNN	Obtención de las matrices de salida en forma de imágenes binarias	2 imágenes de prueba con ruido impulsivo, 2 imagen con ruido gaussiano. Casos de uso del sistema 4.2.5, 4.2.6 Algoritmo de la Tabla 4.1	40 imágenes de salida generadas a partir de las imágenes procesadas por la red PCNN	La red fue descartada por el alto número de parámetros de ajuste y bajo nivel de detección de píxeles con ruido.

Tabla 4.9 Plan de pruebas del sistema pruebas 4.5.13 a 4.5.15

Descripción del objeto de prueba	Características	Criterios para comienzo	Criterios para fin de pruebas	Comentarios
4.5.13 Implementación de la red ICM	Obtención de las matrices de salida de la red en forma de imágenes binarias	10 imágenes de prueba con ruido impulsivo Casos de uso del sistema 4.2.5,4.2.6 Algoritmo de la Tabla 4.2	10 imágenes de salida generadas por la red ICM. 90 imágenes filtradas	Se analizó la función de la red ICM como detector de impulsos.
4.5.14 Implementación de la red SCM	Obtención de las matrices de salida de la red en forma de imágenes binarias	Imágenes de prueba. Casos de uso del sistema 4.2.5,4.2.6 Algoritmo de la Tabla 4.3	10 Imágenes de salida generadas por la red SCM. 40 imágenes filtradas.	Se determinó el número de iteraciones necesarias para el filtrado.
4.5.15 Implementación de la red PCNNNI	Obtención de las matrices de salida de la red en forma de imágenes binarias	Imágenes de prueba. Casos de uso del sistema 4.2.5,4.2.6 Algoritmo de la Tabla 4.5	30 Imágenes de salida generadas por la red PCNNNI 10 imágenes correspondientes a la Matriz de Tiempos.	Se obtuvieron imágenes de salida a partir de la Matriz de Tiempos.

4.5.1 Generación de imágenes de prueba con ruido impulsivo

Se consideraron imágenes en escala de grises correspondientes a rostros (Figura 4.1), animales (Figura 4.2), texturas (Figura 4.3) y (Figura 4.4), y a paisajes (Figura 4.5) a las cuales se agregó ruido impulsivo de probabilidad de ocurrencia de 0.0 a 0.9.

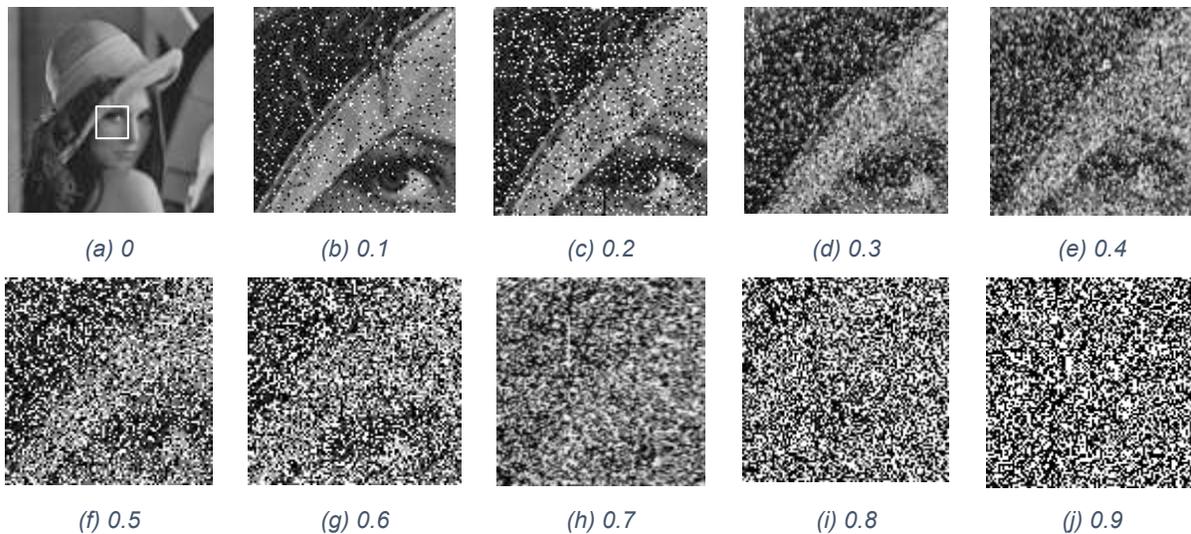


Figura 4.3 Sección ampliada de imágenes de Lena con ruido impulsivo de probabilidad de ocurrencia de 0 a 0.9

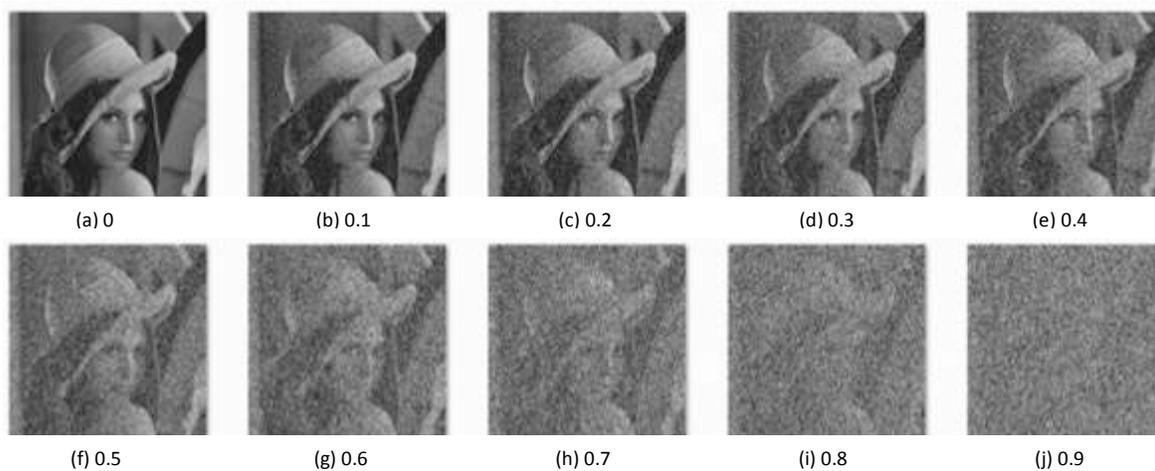


Figura 4.4 Imágenes miniatura de Lena con ruido impulsivo de probabilidad de ocurrencia de 0 a 0.9

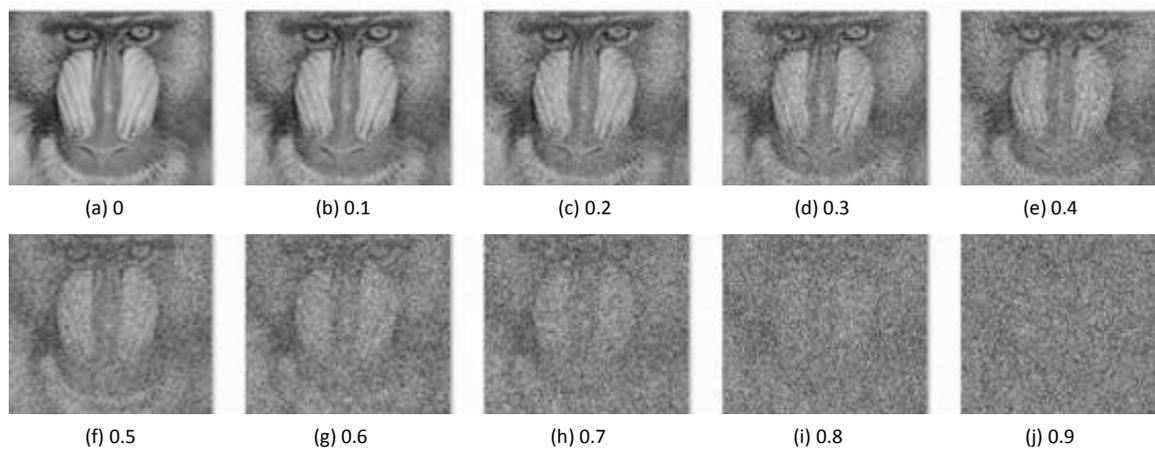


Figura 4.5 Imágenes miniatura de Baboon con ruido impulsivo de probabilidad de ocurrencia de 0 a 0.9

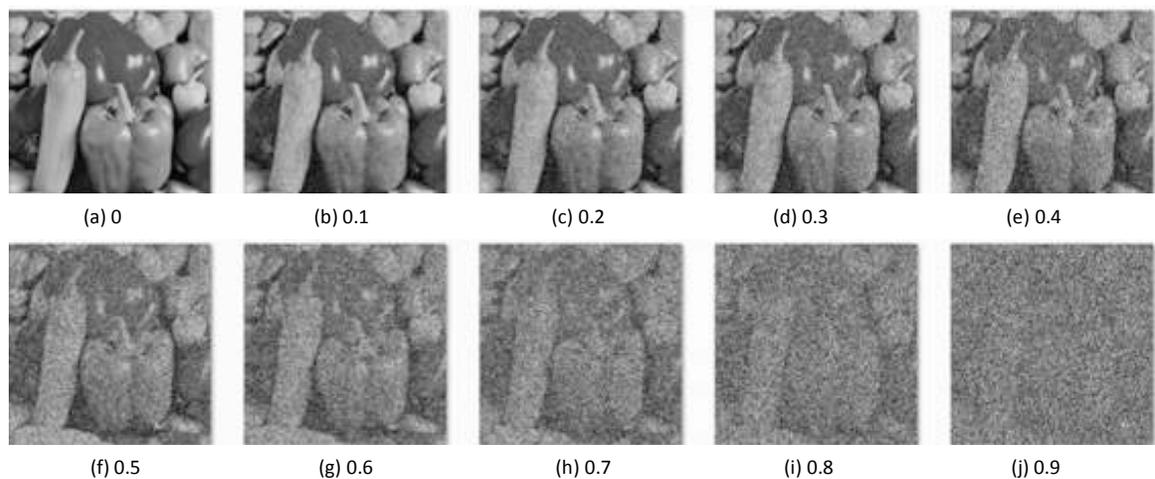


Figura 4.6 Imágenes miniatura de Peppers con ruido impulsivo de probabilidad de ocurrencia de 0 a 0.9

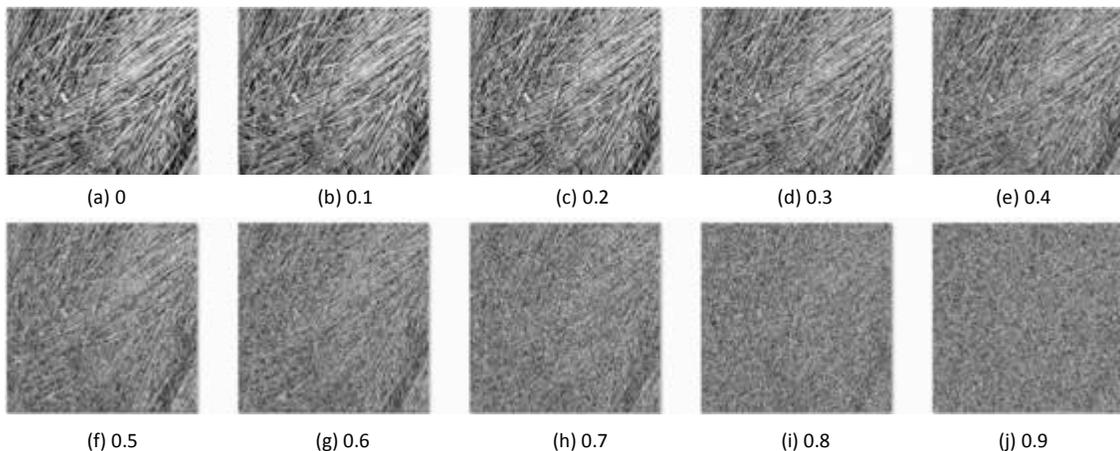


Figura 4.7 Imágenes miniatura de Grass con ruido impulsivo de probabilidad de ocurrencia de 0 a 0.9

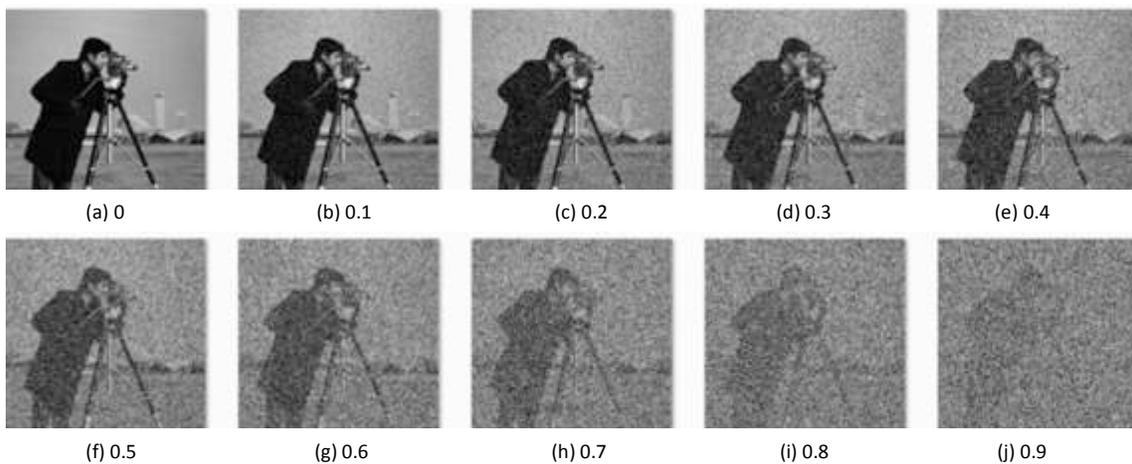


Figura 4.8 Imágenes miniatura de Cameraman con ruido impulsivo de probabilidad de ocurrencia de 0 a 0.9

4.5.2 Generación de imágenes de prueba con ruido gaussiano

Se llevó a cabo la simulación de ruido aditivo gaussiano con $\mu = 0$ y varianzas entre 0 y 0.3. En las secciones ampliadas se observa que el ruido afecta a todos los píxeles de la imagen en cierto grado y se acentúa conforme aumenta la varianza (Figura 4.9). Se generaron 34 imágenes en escala de grises de 512x512 píxeles utilizando *Lena* (Figura 4.10), *Peppers* y *Baboon* (Figura 4.11) afectadas por ruido gaussiano.

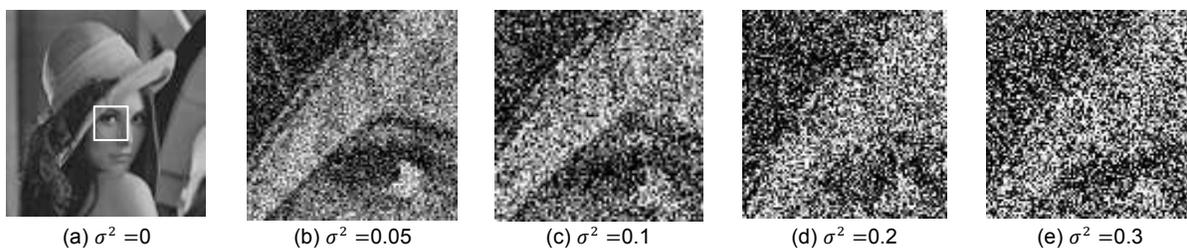


Figura 4.9 Sección de imágenes con ruido gaussiano de $\mu = 0$ al incrementar la varianza



Figura 4.10 Imágenes miniatura de Lena con ruido gaussiano de $\mu = 0$ al incrementar la varianza

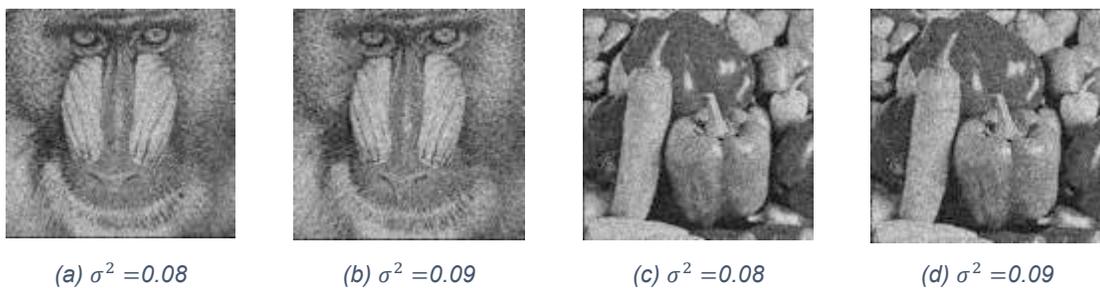


Figura 4.11 Imágenes miniatura de Baboon y Peppers con ruido gaussiano de $\mu = 0$ al incrementar la varianza

4.5.3 Generación de imágenes filtradas con el filtro de mediana

A través de la plataforma se generaron 40 imágenes con ruido impulsivo filtradas mediante operador de mediana de 3×3 , se muestra una sección ampliada de *Lena* (Figura 4.12), y las imágenes completas en miniatura de *Lena*, *Baboon*, *Peppers* y *Camerman* (Figura 4.13-4.16).

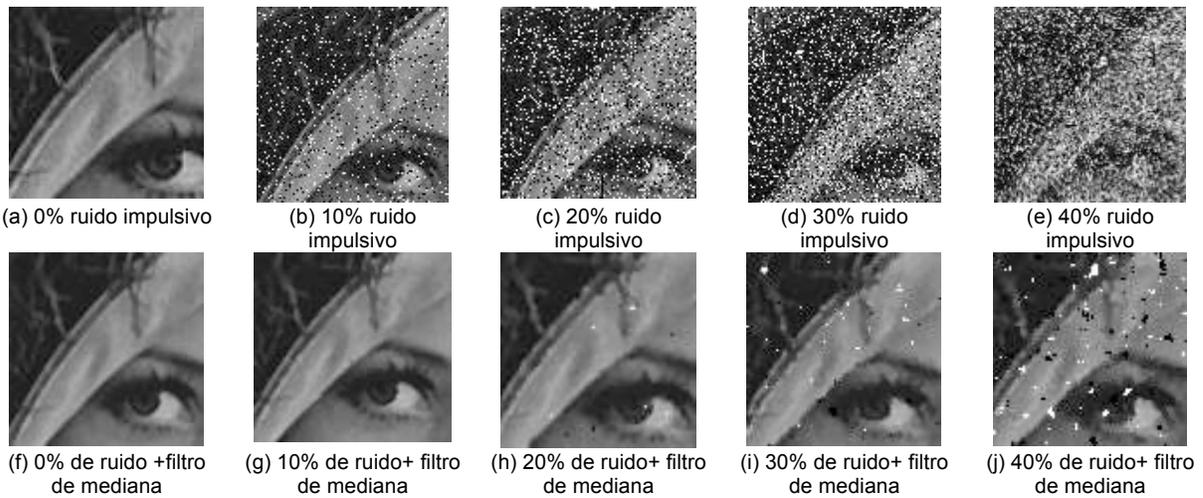


Figura 4.12 Sección ampliada de imágenes de Lena con ruido impulsivo procesadas con filtro de mediana de 3x3

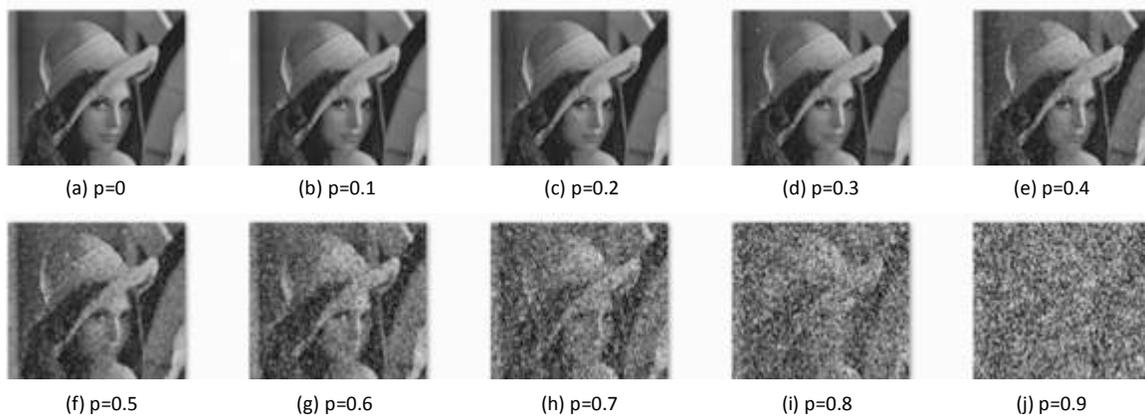


Figura 4.13 Imágenes de Lena con ruido impulsivo de probabilidad p filtradas con el operador de mediana de 3x3

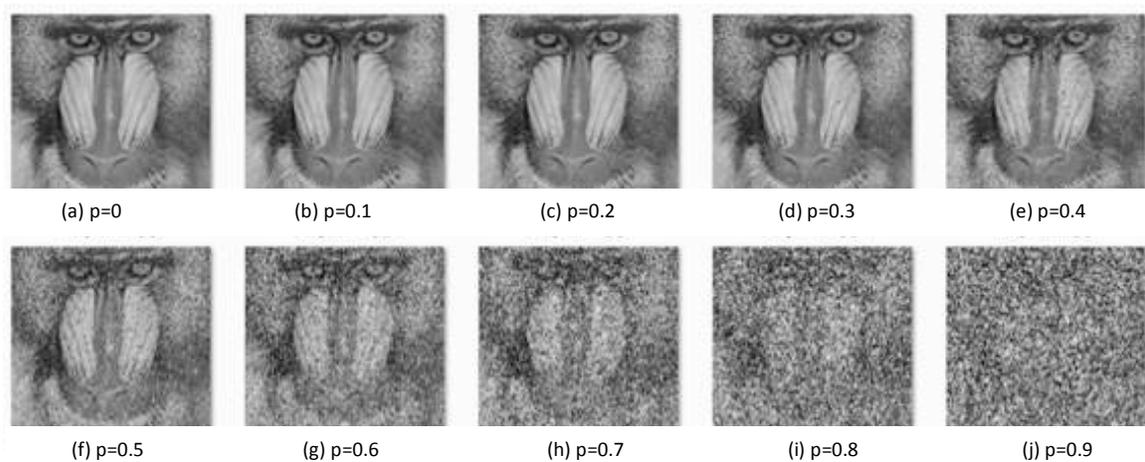


Figura 4.14 Imágenes de Baboon con ruido impulsivo de probabilidad p filtradas con el operador de mediana de 3x3

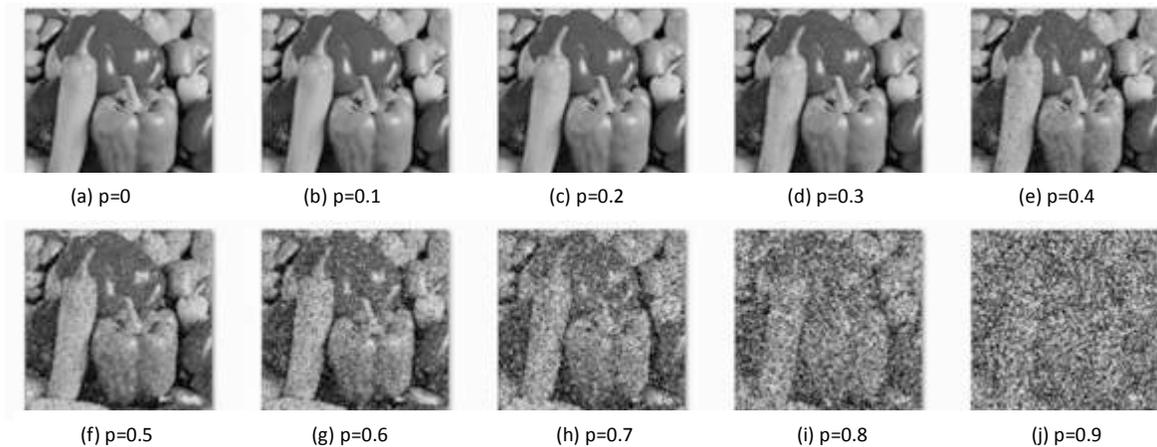


Figura 4.15 Imágenes de Peppers con ruido impulsivo de probabilidad p filtradas con el operador de mediana de 3×3

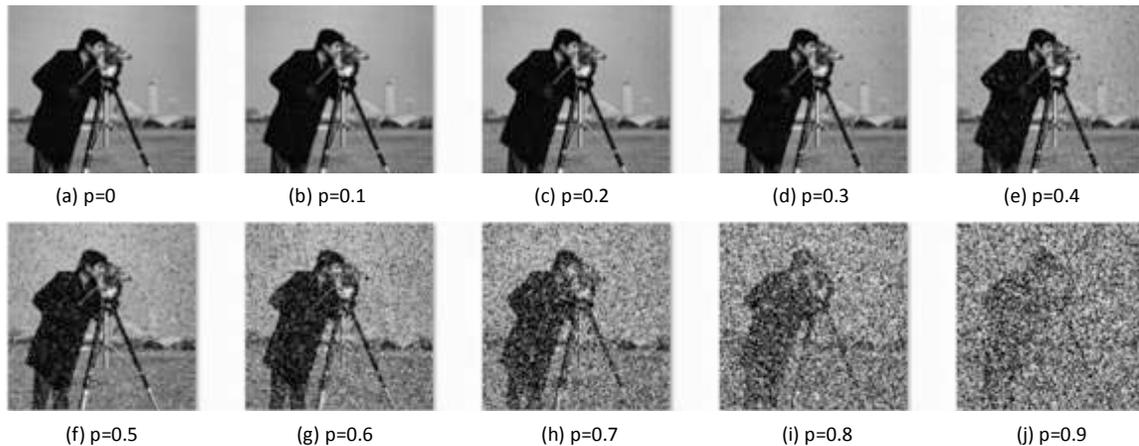


Figura 4.16 Imágenes de Cameraman con ruido impulsivo de probabilidad p filtradas con el operador de mediana de 3×3

Se observa que cuando la probabilidad de ocurrencia del ruido impulsivo es mayor a 0.7 la recuperación del filtro de mediana ya es pobre.

Por otro lado se generaron 36 imágenes con ruido gaussiano filtradas mediante el operador de mediana de 3×3 implementado en el sistema, se observa que la recuperación visual es buena para varianzas menores de 0.1 (Figura 4.17). Se muestran resultados del filtrado de mediana de 3×3 para *Lena*, *Baboon* y *Peppers* (Figura 4.18, 4.19).

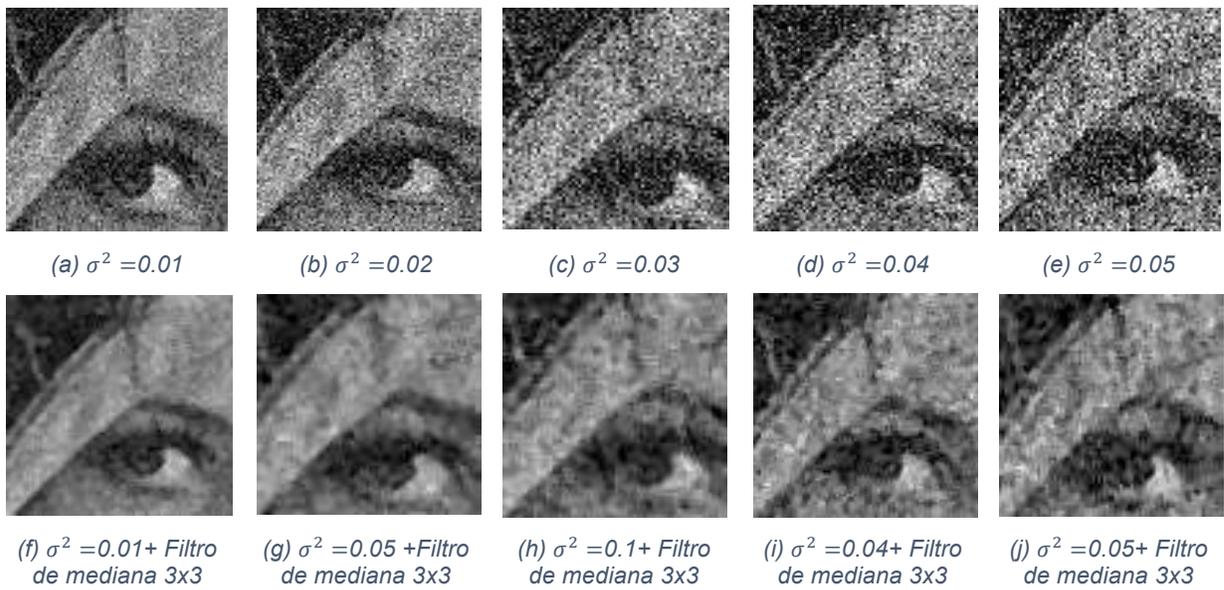


Figura 4.17 Imágenes con ruido gaussiano de $\mu = 0$ filtradas con filtro de mediana



Figura 4.18 Imágenes miniatura de Lena con ruido gaussiano filtradas con filtro de mediana

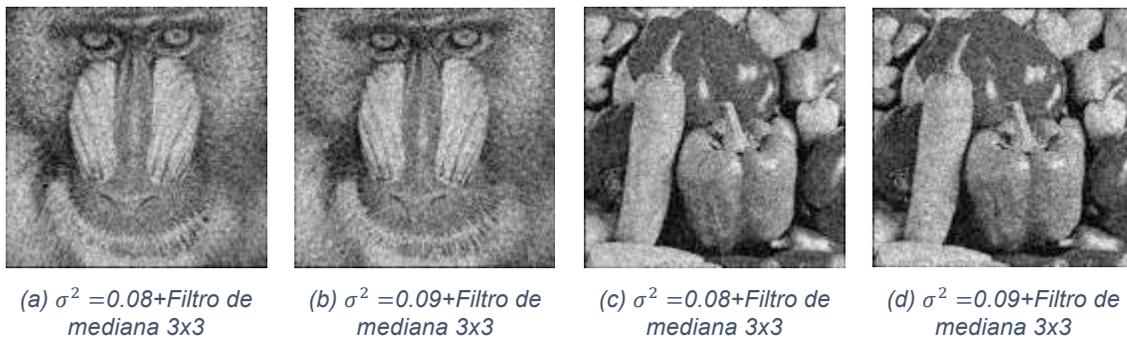


Figura 4.19 Imágenes miniatura de Baboon y Peppers con ruido gaussiano filtradas con filtro de mediana

4.5.4 Generación de imágenes filtradas con el filtro promedio

En las imágenes procesadas con el filtro promedio se observa la poca efectividad para eliminar el ruido impulsivo, mientras que una imagen sin ruido se ve afectada en sus detalles, además se observa que la recuperación visual para imágenes con probabilidad de ruido impulsivo superior a 0.4 es muy pobre (Figura 4.20). Se generaron 40 imágenes con ruido impulsivo filtradas mediante operador promedio de 3x3 (Figura 4.21-4.24).

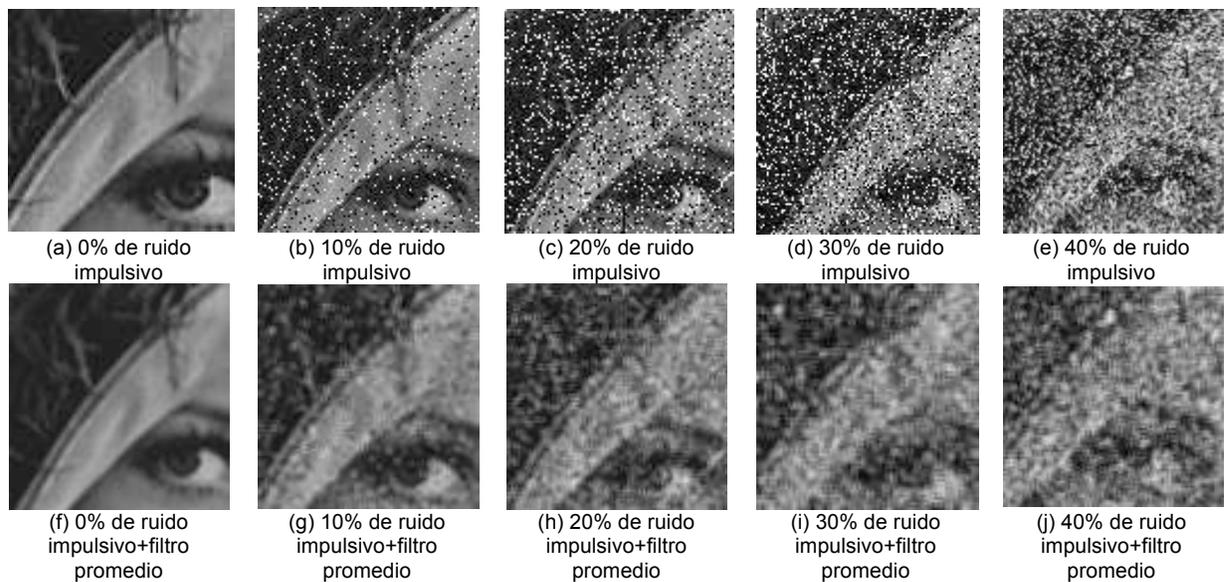


Figura 4.20 Sección ampliada de imágenes de Lena con ruido impulsivo filtradas con el filtro promedio de 3x3

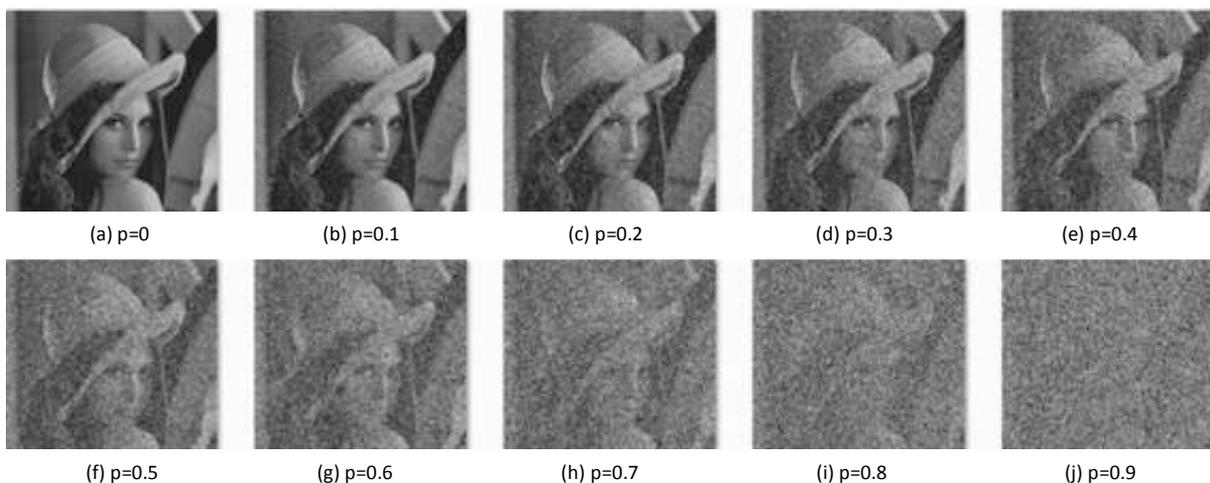


Figura 4.21 Imágenes de Lena con ruido impulsivo de probabilidad p filtradas con el filtro promedio de 3×3

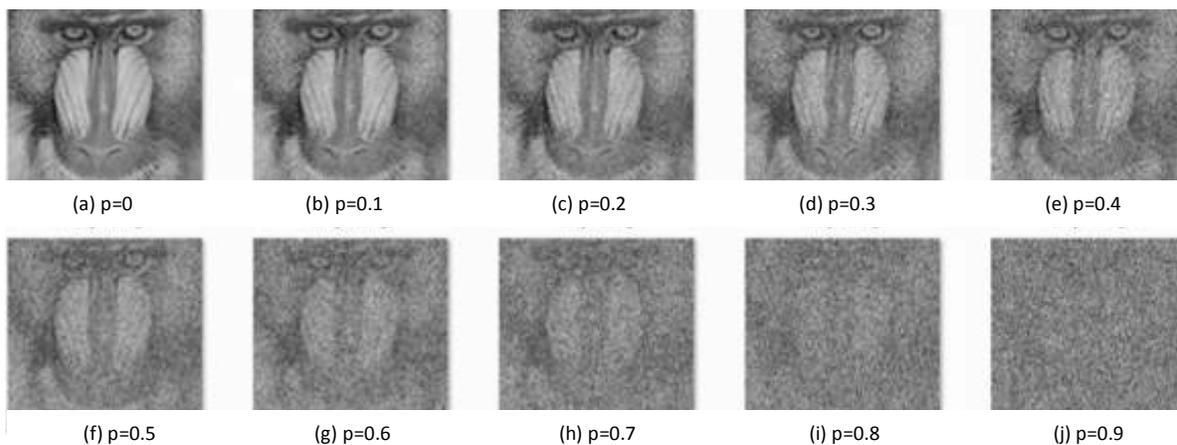


Figura 4.22 Imágenes de Baboon con ruido impulsivo de probabilidad p filtradas con el filtro promedio de 3×3

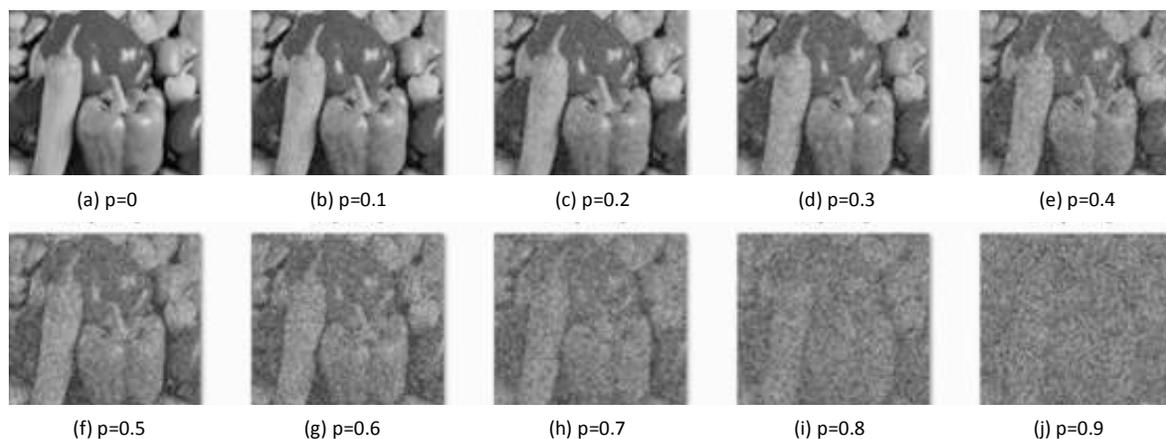


Figura 4.23 Imágenes de Peppers con ruido impulsivo de probabilidad p filtradas con el filtro promedio de 3×3

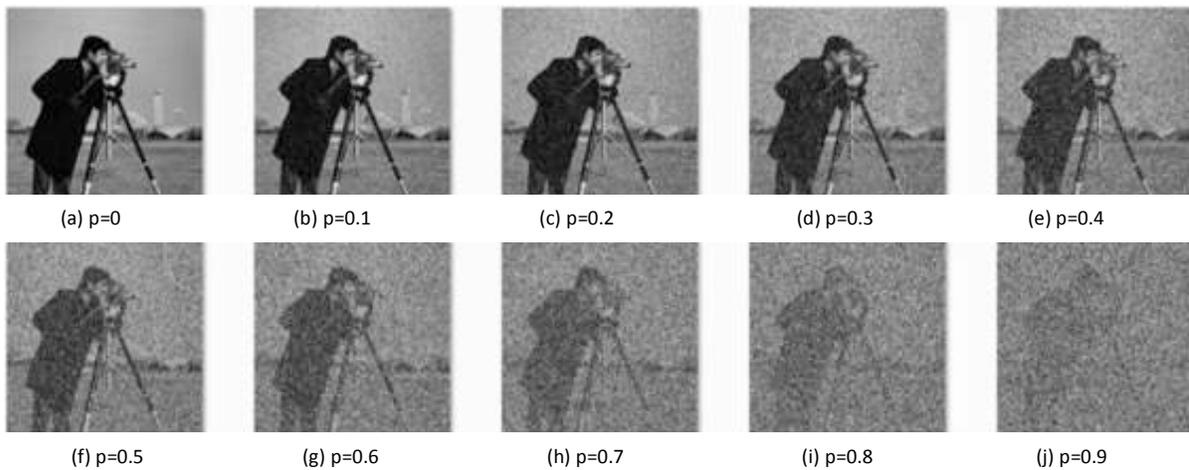


Figura 4.24 Imágenes de Cameraman con ruido impulsivo de probabilidad p filtradas con el filtro promedio de 3×3

4.5.5 Generación de imágenes filtradas con el filtro morfológico

Se generaron 10 imágenes con ruido impulsivo de probabilidad de ocurrencia de 0 a 0.9 filtradas mediante el filtro morfológico tipo apertura-cerradura, en ellas se observa que la recuperación se lleva a cabo por regiones que se vuelven nítidas, no obstante la recuperación visual es pobre cuando la probabilidad del ruido supera el 0.5 (Figura 4.25).

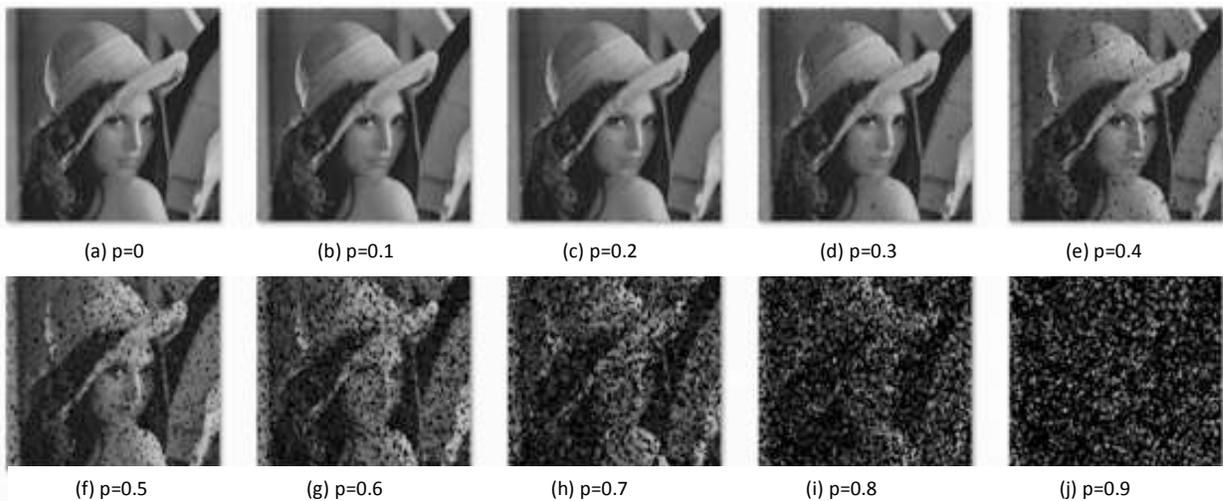


Figura 4.25 Imágenes con ruido impulsivo de probabilidad p filtradas con el filtro morfológico

Sólo se generaron imágenes filtradas con el filtro morfológico a partir de la imagen de *Lena*, no obstante la plataforma permite manipular cualquier imagen en formato bmp de 512×512 píxeles.

4.5.6 Generación de imágenes filtradas con el filtro *Wiener*

Se generaron 34 imágenes con ruido gaussiano filtradas mediante el filtro *Wiener* por medio de Matlab. Se filtraron imágenes con ruido gaussiano de varianzas de 0.01 a 0.3 de *Lena* (Figura 4.26), *Baboons* y *Peppers* (Figura 4.27). Como se observa, la recuperación del filtro es buena para ruido gaussiano de varianzas inferiores a 0.05 y en general la preservación de detalles es buena (Figura 4.28).

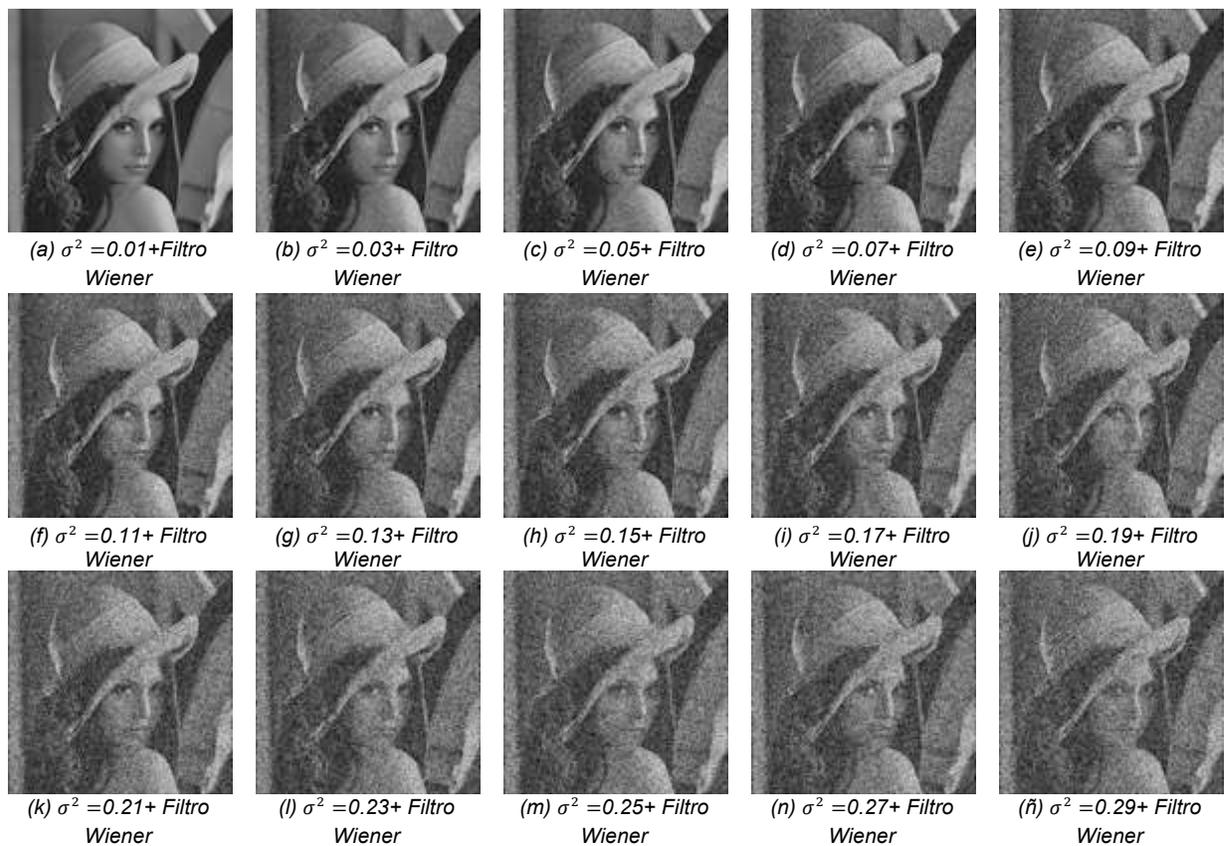


Figura 4.26 Imágenes miniatura con ruido gaussiano filtradas con filtro *Wiener*

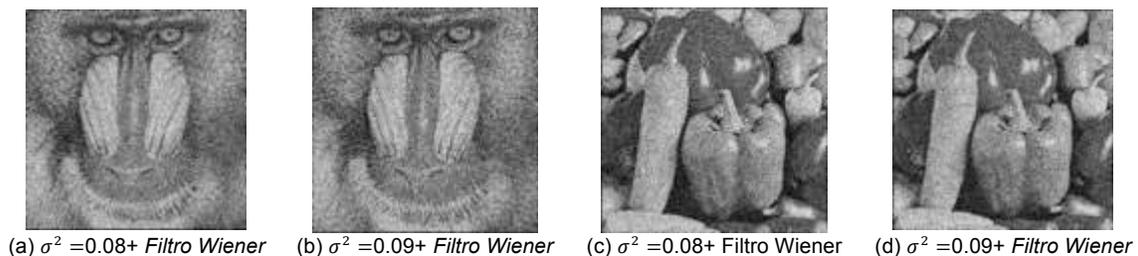


Figura 4.27 Imágenes miniatura de *Baboon* y *Peppers* con ruido gaussiano filtradas con filtro *Wiener*

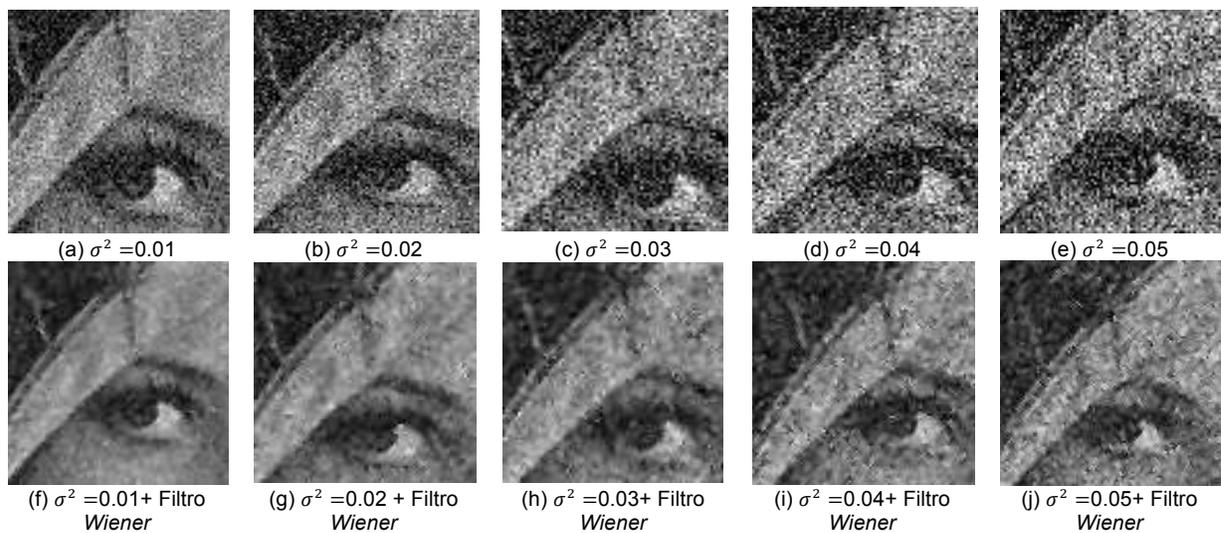


Figura 4.28 Imágenes ampliadas con ruido gaussiano de $\mu = 0$ filtradas con filtro Wiener

4.5.7 Métrica PSNR implementada

Se realizó la medición del PSNR de imágenes de *Lena* con ruido impulsivo filtradas mediante el filtro de mediana, el filtro promedio y el filtro morfológico, se observa que el filtro de mediana tiene mejor comportamiento para probabilidades menores a 0.3 y el filtro morfológico para probabilidades de entre 0.3 y 0.5 (Figura 4.29). Se repitió la medición con las imágenes de *Baboon* (Figura 4.30), *Peppers* (Figura 4.31), *Cameraman* (Figura 4.32) y *Grass* (Figura 4.33) en donde se muestra el nivel de PSNR de la imagen con ruido es decir sin filtrar y la recuperación con respecto a esta línea de los filtros promedio y de mediana.

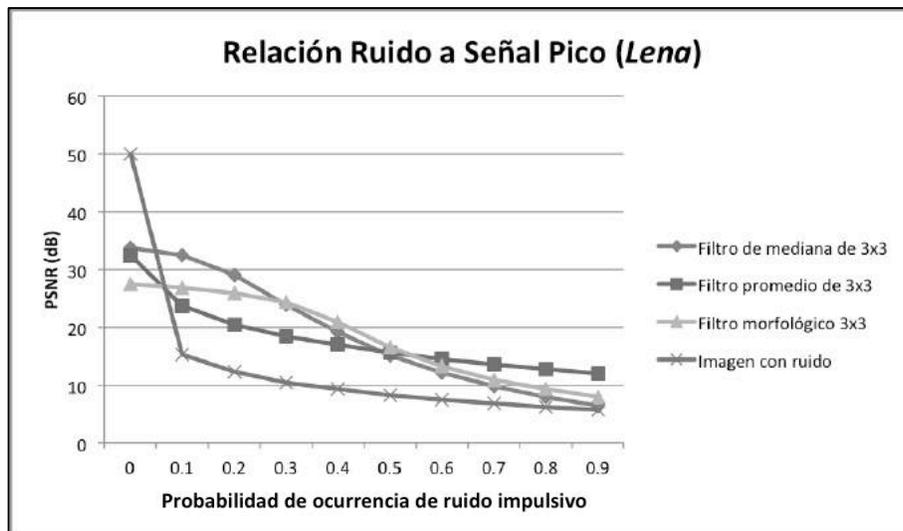


Figura 4.29 PSNR del filtro de mediana, promedio y morfológico para *Lena* con ruido impulsivo

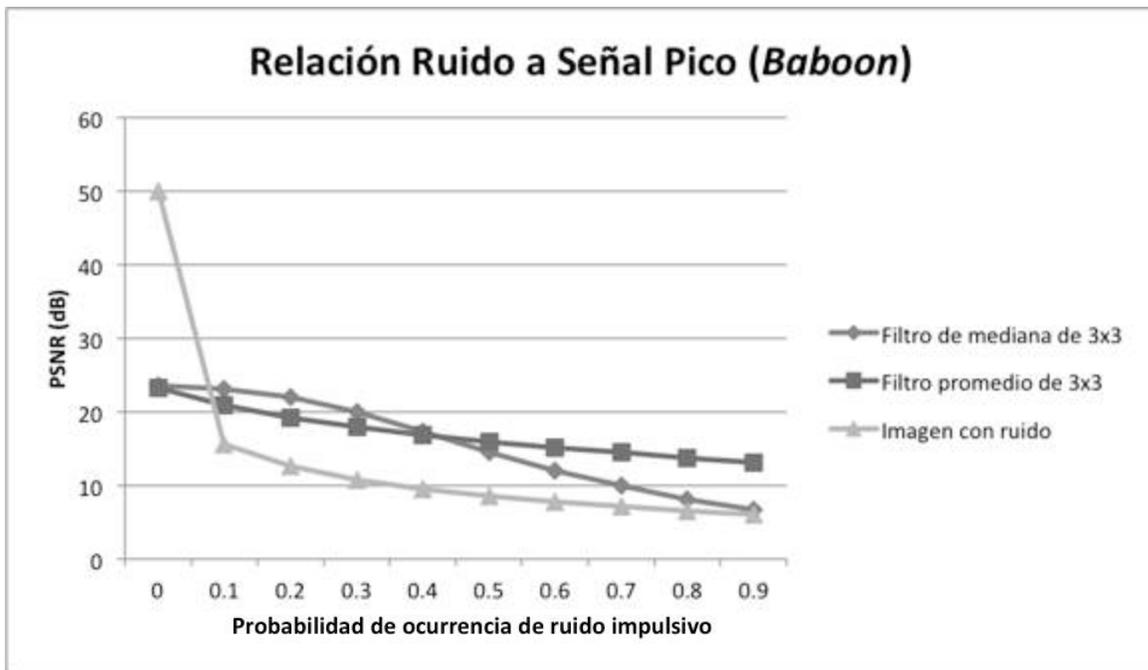


Figura 4.30 PSNR del filtro de promedio y mediana para la imagen de Baboon con ruido impulsivo

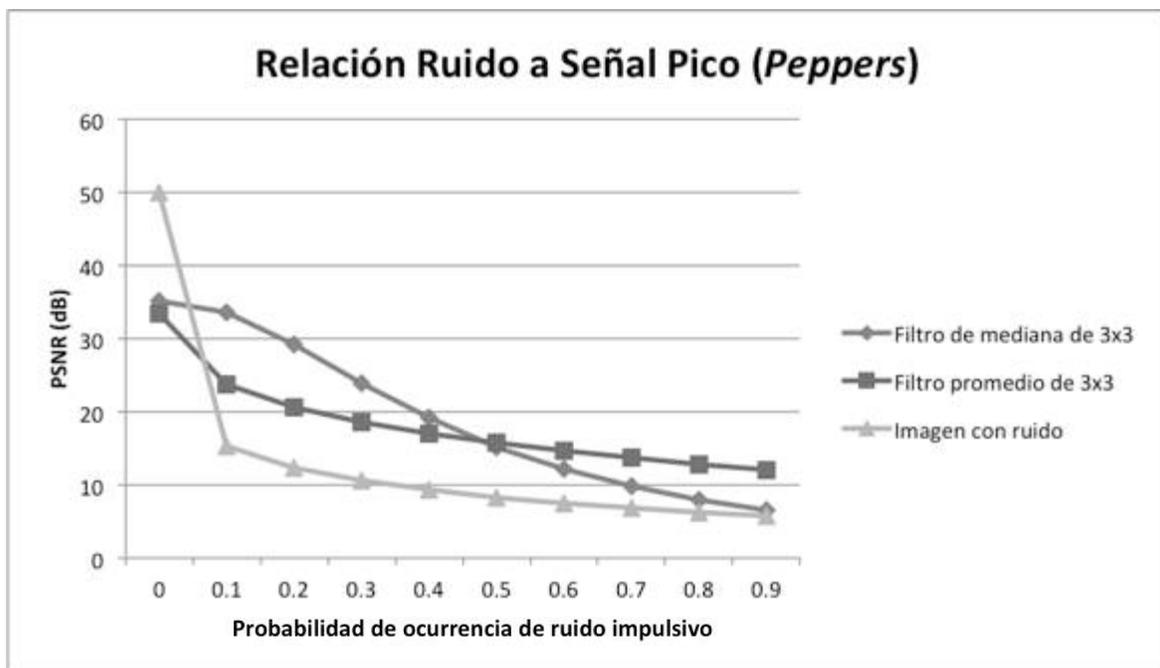


Figura 4.31 PSNR del filtro de promedio y mediana para la imagen de Peppers con ruido impulsivo

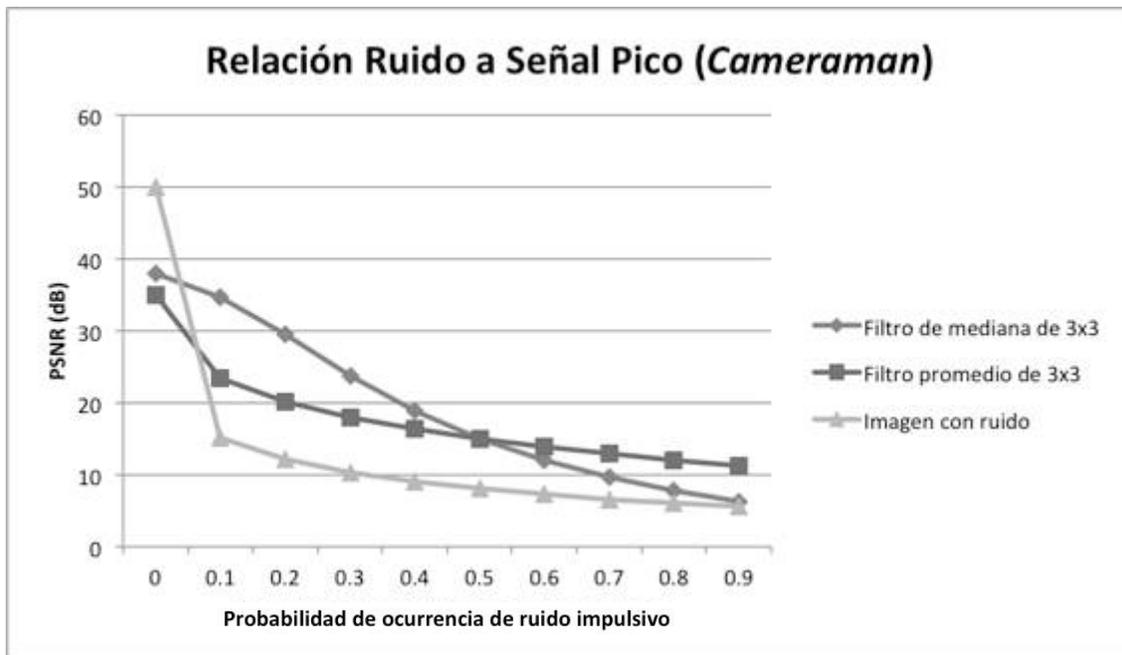


Figura 4.32 PSNR del filtro de promedio y mediana para la imagen de *Cameraman* con ruido impulsivo

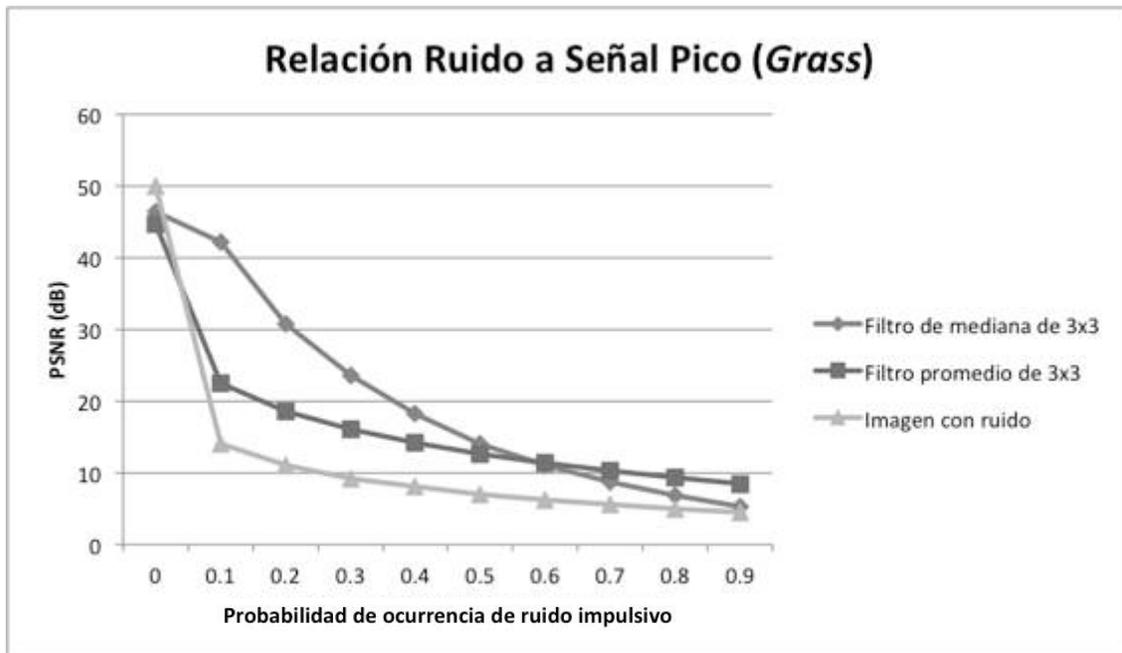


Figura 4.33 PSNR del filtro de promedio y mediana para la imagen de *Grass* con ruido impulsivo

La medición de la calidad de la imagen para las imágenes con ruido gaussiano de distintas media y varianza muestra las diferencias en PSNR, el filtro *Wiener* tiene un desempeño superior al filtro de mediana por al menos 1dB para varianzas inferiores a 0.15 y máximo 3dB para varianzas superiores a 0.15 y menores a 0.3 (Figura 4.34).

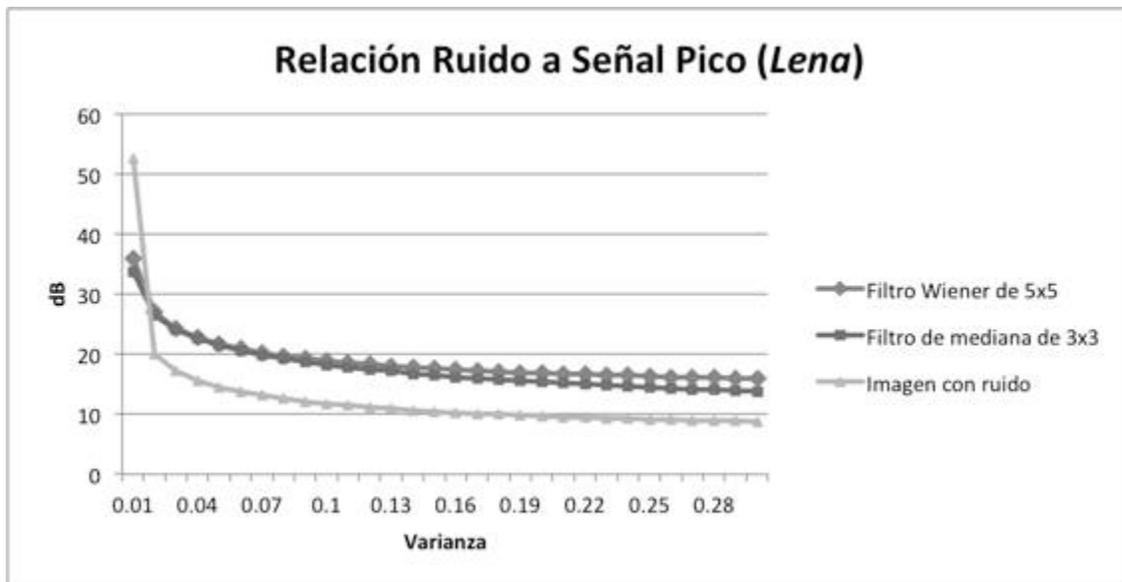


Figura 4.34 PSNR para la imagen de Lena con ruido gaussiano con distintos filtros

4.5.8 Métrica MAE implementada

Por otro lado el MAE indica que para imágenes con ruido impulsivo de probabilidad de ocurrencia entre 0.1 y 0.9 el error aumenta de 10 a casi 120 se observa que esta métrica muestra coherencia del efecto del ruido sobre los distintos tipos de imágenes con ruido impulsivo ya que para todas ellas presenta un comportamiento similar (Figura 4.35).

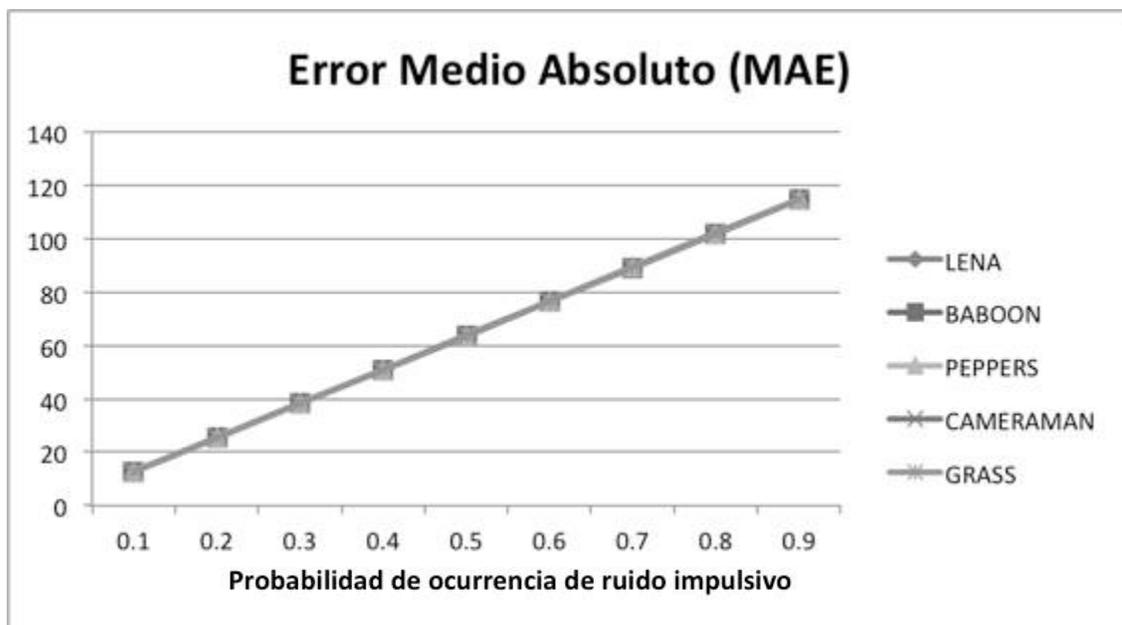


Figura 4.35 MAE para imágenes con ruido impulsivo

Se calculó el Error Medio Absoluto (MAE) para imágenes de *Lena* con ruido gaussiano filtradas con el Filtro *Wiener* y el filtro de mediana de 3x3 se compararon las líneas con la de la imagen con ruido, es decir sin filtrar (Figura 4.36).

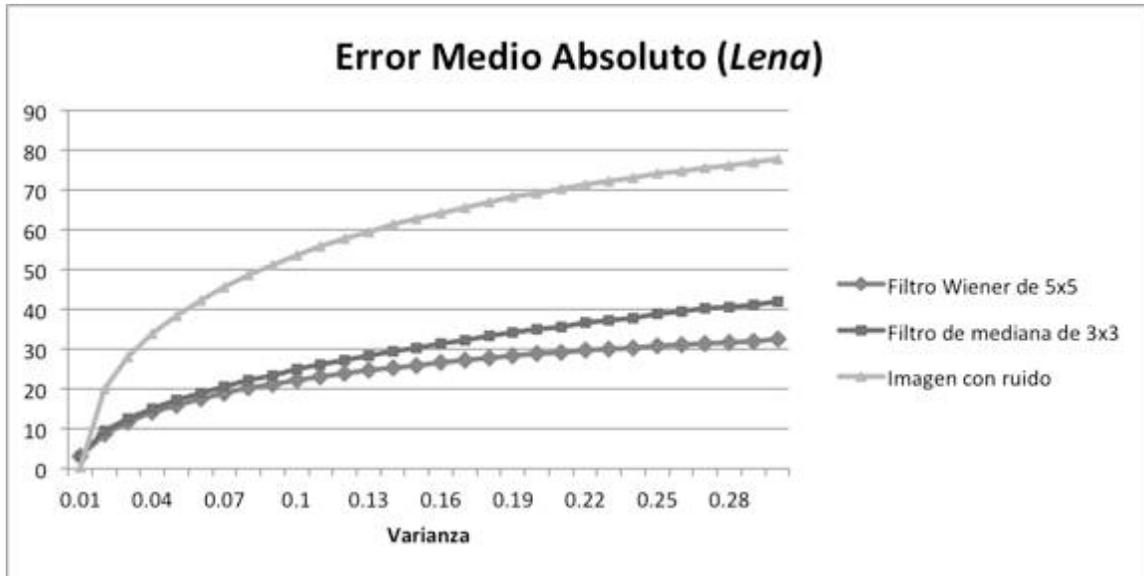


Figura 4.36 MAE para imágenes con ruido gaussiano filtradas con Filtro de mediana y Wiener

4.5.9 Métrica MSE implementada

Se determinó la calidad de la imagen para las imágenes de *Lena*, *Baboon*, *Peppers*, *Camerman* y *Grass* con ruido impulsivo de probabilidad entre 0.1 y 0.9 mediante el Error Cuadrático Medio (Figura 4.37).

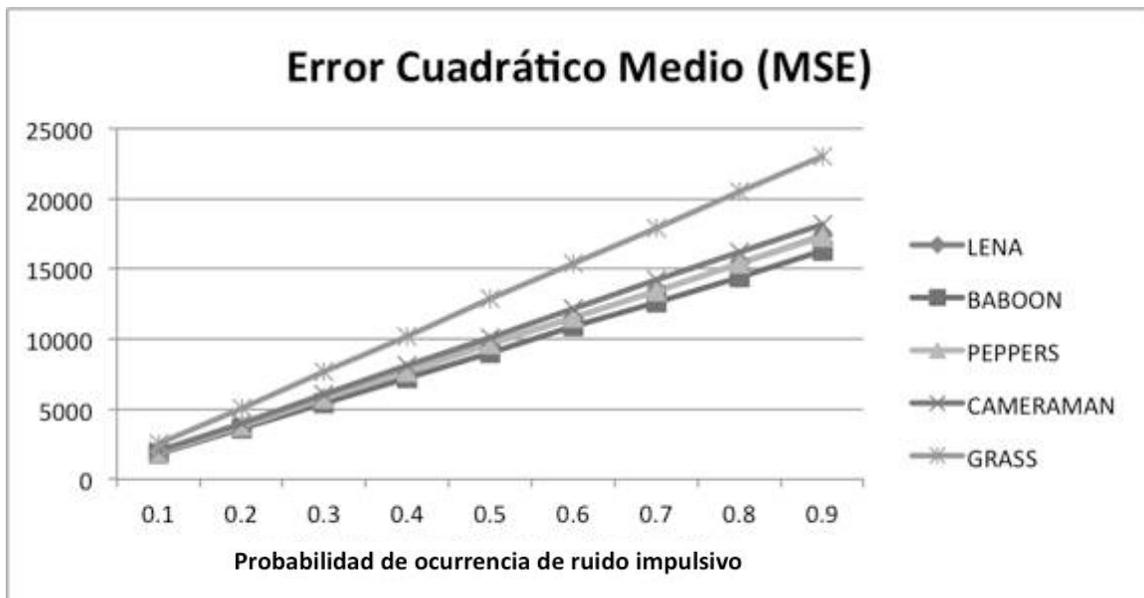


Figura 4.37 MSE para diversas imágenes con ruido impulsivo

Se calculó el Error Cuadrático Medio (MAE) para imágenes de *Lena* con ruido gaussiano de diversas varianzas sin filtrar y se comparó con el de imágenes filtradas por medio del filtro de mediana de 3x3 y el filtro *Wiener*, se observó la reducción substancial del ruido utilizando el filtro de mediana y una mayor reducción de error utilizando el filtro *Wiener* (Figura 4.38).

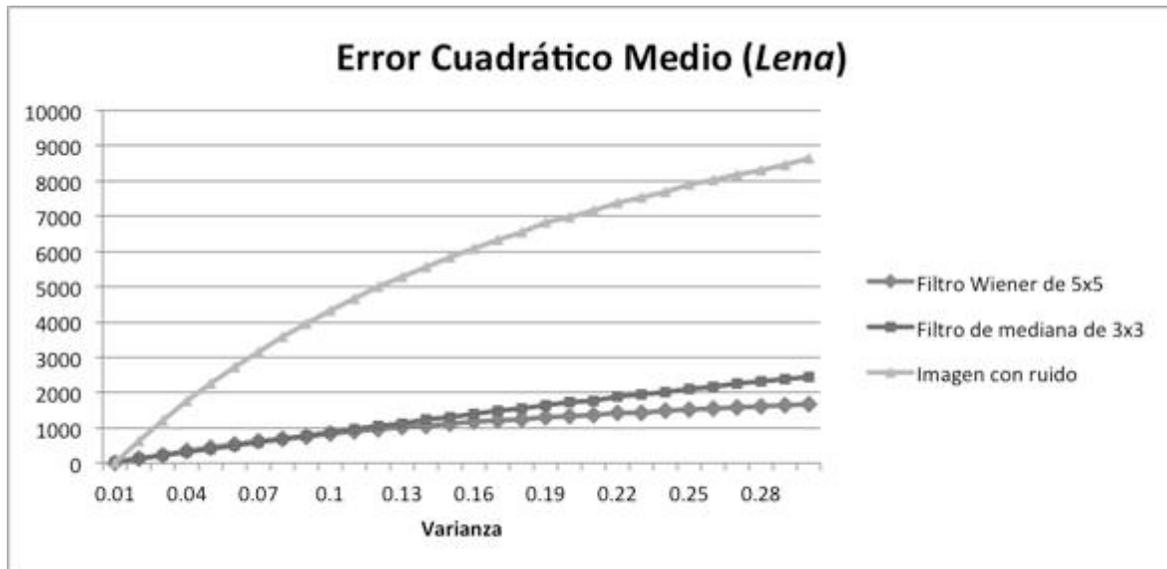


Figura 4.38 MSE para *Lena* con ruido gaussiano con el filtro *Wiener* y el filtro de mediana de 3x3

4.5.10 Métrica NMSE implementada

Se realizó la comparación en NMSE para las distintas imágenes de prueba con ruido impulsivo de probabilidad de ocurrencia de 0.1 a 0.9 sin filtrar donde se observa un comportamiento acorde al PSNR pero acotado entre 0 y 1 (Figura 4.39).

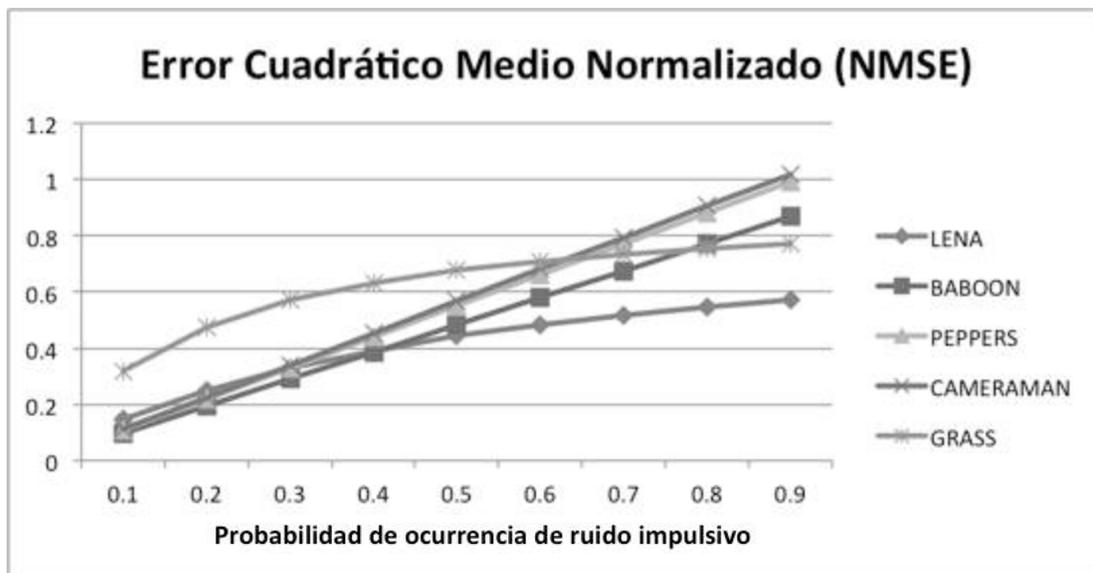


Figura 4.39 NMSE para imágenes con ruido impulsivo

Se realizó el cálculo del NMSE (Figura 4.40) de las imágenes de *Lena* con ruido gaussiano filtradas mediante el filtro *Wiener* y el filtro de mediana de 3x3 comparándolo con el NMSE de la imagen con ruido sin filtrar para observar la recuperación que logran ambos filtros.

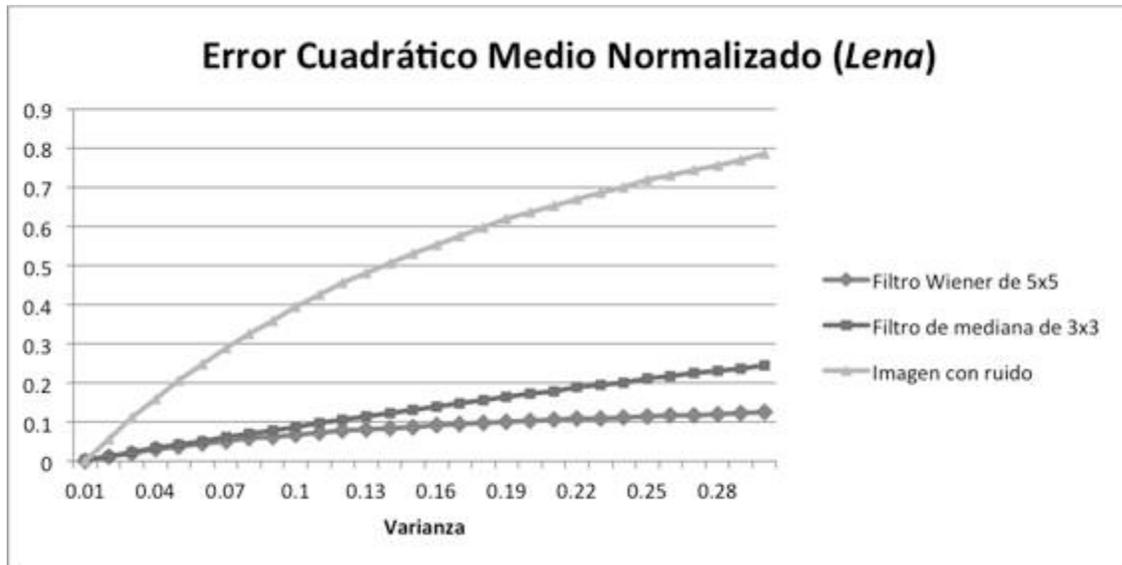


Figura 4.40 NMSE para *Lena* con ruido gaussiano filtrada con filtro Wiener y filtro de mediana de 3x3

4.5.11 Medición del tiempo de procesamiento

En cuanto al tiempo que tarda cada método en procesar la imagen de *Lena* de 512x512 píxeles con ruido impulsivo, el filtro más rápido fue el de promedio con menos de 100ms, seguido por el de mediana con 250 ms y por último el morfológico (Figura 4.41).

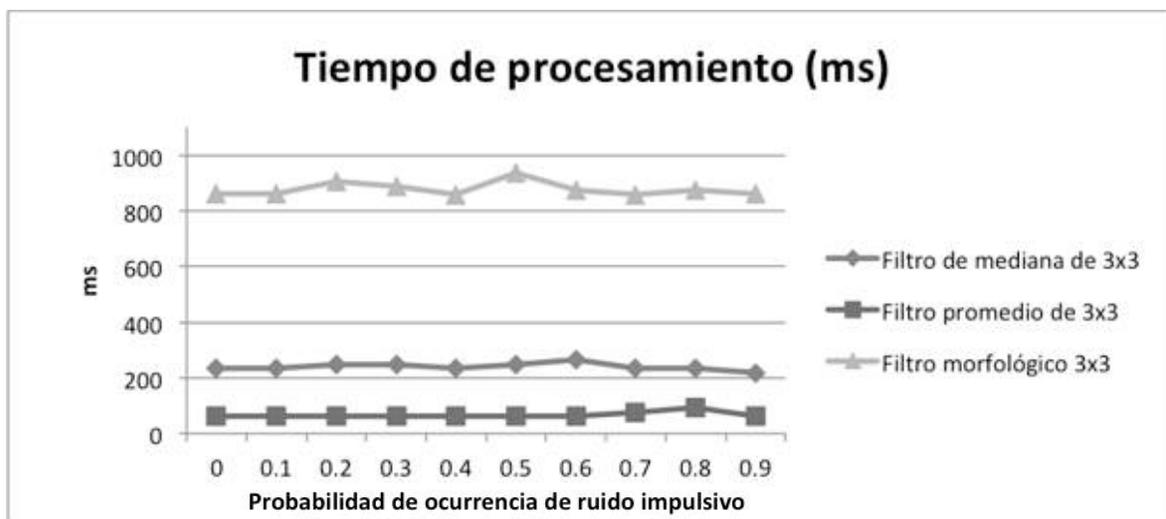


Figura 4.41 Tiempo de procesamiento del filtro de mediana, promedio y morfológico

4.5.12 Implementación de la red PCNN

Se implementó la red PCNN en la plataforma de experimentación de acuerdo al algoritmo de la Tabla 4.1 con valores de $\alpha_F = 0.1$; $\alpha_L = 1.0$; $\alpha_E = 1.0$; $V_L = 0.2$; $V_E = 20.0$; $V_F = 0.5$; $\beta = 0.1$ dados en [Ma, 2010]. Se procesó una imagen de *Lena* en escala grises de 512x512 pixeles sin ruido, se realizaron 10 iteraciones de la red PCNN en las que se observaron las imágenes binarias de salida para estudiar los cambios producidos en cada iteración (Figura 4.42).

En la primera iteración todas las salidas de neuronas se activaron por lo que todos los pixeles tienen valor de 1, en la segunda y tercera iteraciones ninguna neurona se activó por lo que todos los pixeles tienen valor de 0. A partir de la cuarta iteración se observa la activación de todas las neuronas excepto de unas pocas aisladas; en la quinta iteración se observa el patrón de activación contrario al anterior; en iteraciones posteriores se observa que los patrones de activación corresponden a los bordes de la imagen procesada.

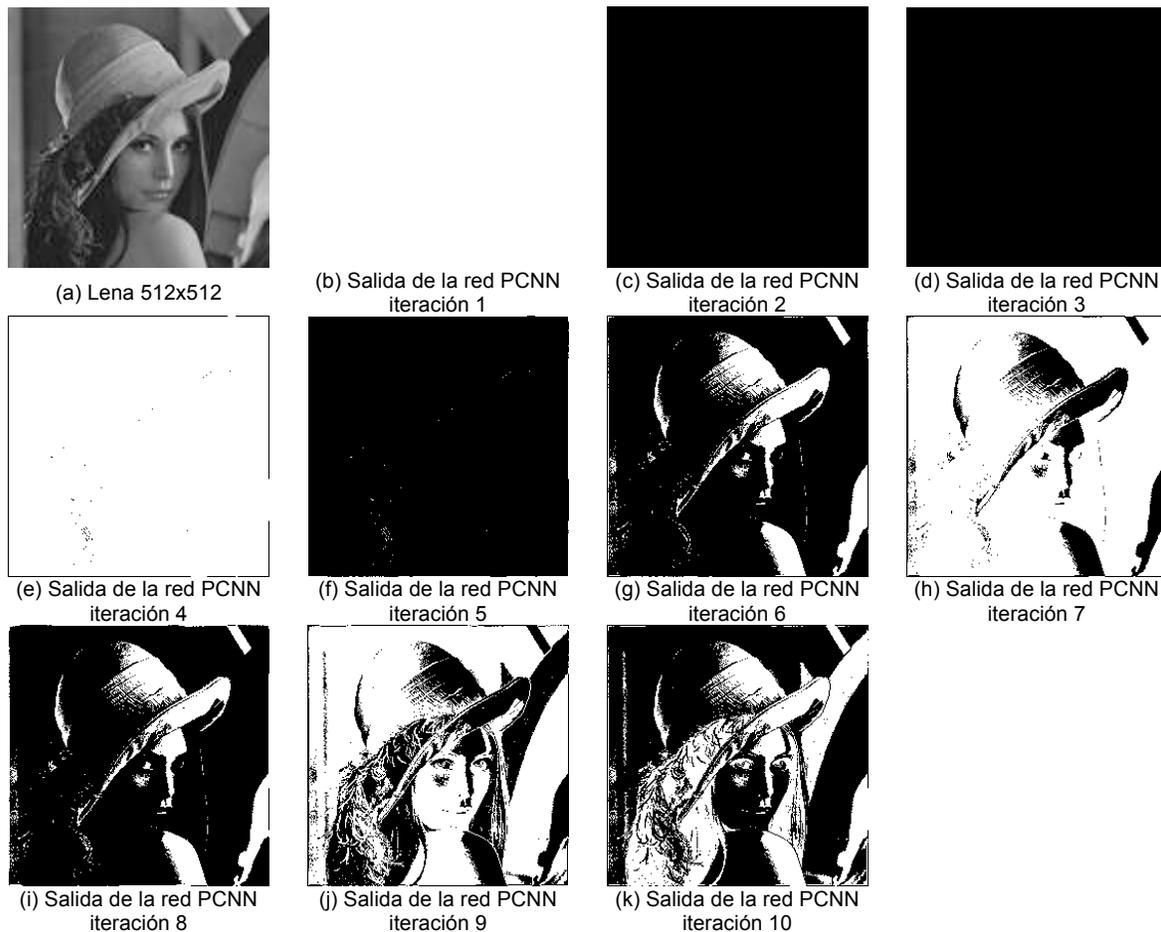


Figura 4.42 Salidas binarias de la red PCNN para 10 iteraciones para una imagen sin ruido

Se repitió la prueba de procesamiento mediante la red PCNN utilizando imágenes con ruido, para ello se empleó la imagen de *Lena* con 10% de ruido impulsivo. Se analizó una sección ampliada de la imagen con ruido y las correspondientes salidas binarias de la red, se observa que en la primera iteración se activaron todas las neuronas excepto aquellas correspondientes a píxeles afectados con ruido tipo pimienta, en la segunda iteración el patrón de salida mostró que los píxeles activos corresponden a los píxeles con ruido pimienta (Figura 4.43).

En la tercera iteración se observó que no hubo salidas activas, en la cuarta y quinta iteraciones se encontró que se repitieron los patrones de activación de la primera y segunda iteración respectivamente, es decir, volvieron a destacar los píxeles con ruido pimienta. En la sexta iteración destacan los píxeles contaminados con ruido pimienta y algunos píxeles sin ruido, en iteraciones posteriores destacan píxeles con y sin ruido indistintamente.

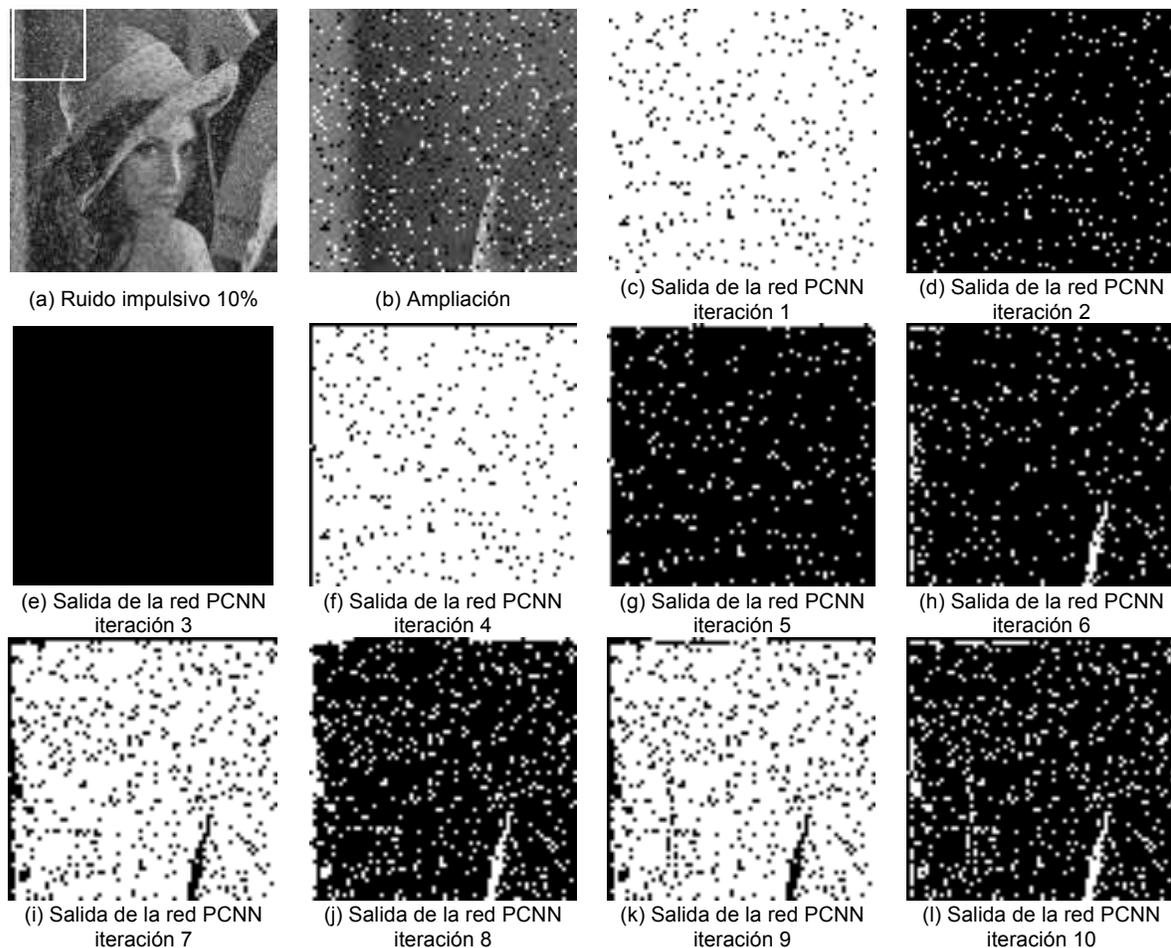


Figura 4.43 Secciones ampliadas de *Lena* con ruido impulsivo procesada con la red PCNN

Por otro lado se llevó a cabo el procesamiento de imágenes con ruido gaussiano mediante la red PCNN, se utilizaron imágenes de *Lena* con ruido gaussiano de varianza 0.1 (Figura 4.44). Se observó que el comportamiento de la red PCNN es similar para el ruido impulsivo y para el ruido gaussiano, pues en cada iteración destacaron los pixeles con nivel de gris igual a 255 que se activan y desactivan en la primera, segunda, cuarta y quinta iteración respectivamente.

Las iteraciones posteriores mostraron la activación de neuronas correspondientes a los pixeles con los niveles de gris más altos y más bajos de la imagen, en este caso no es posible identificar fácilmente a los pixeles ruidosos y patrones de activación que permitan su filtrado selectivo.

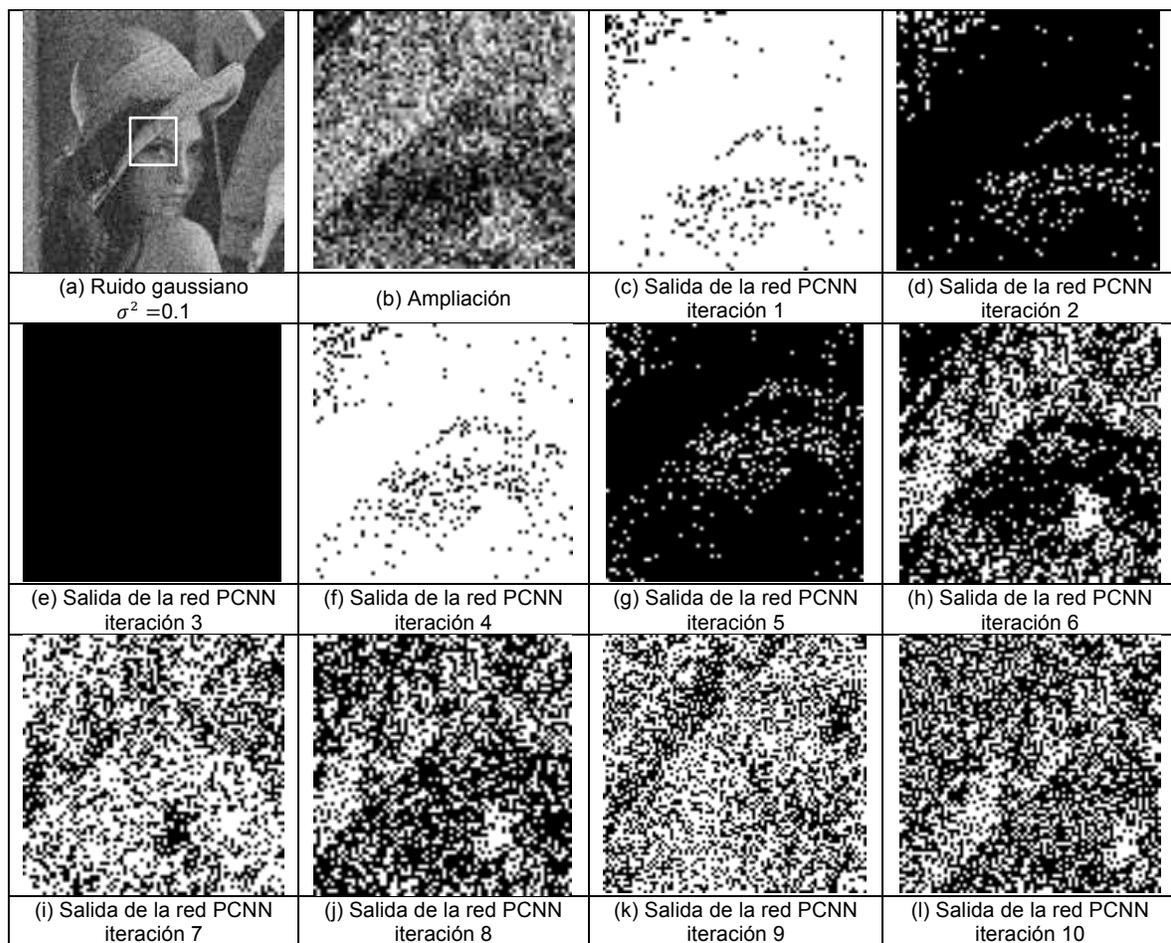


Figura 4.44 *Lena* con ruido gaussiano $\sigma^2 = 0.1$ procesada mediante la red PCNN

Se repitió la prueba con una imagen de *Lena* con ruido gaussiano de varianza 0.3 (Figura 4.45) y se observó que las salidas de las iteraciones 1 y 4 son muy similares entre sí, así como las salidas de las iteraciones 2 y 5, mientras que la iteración 3 no tiene salidas activas.

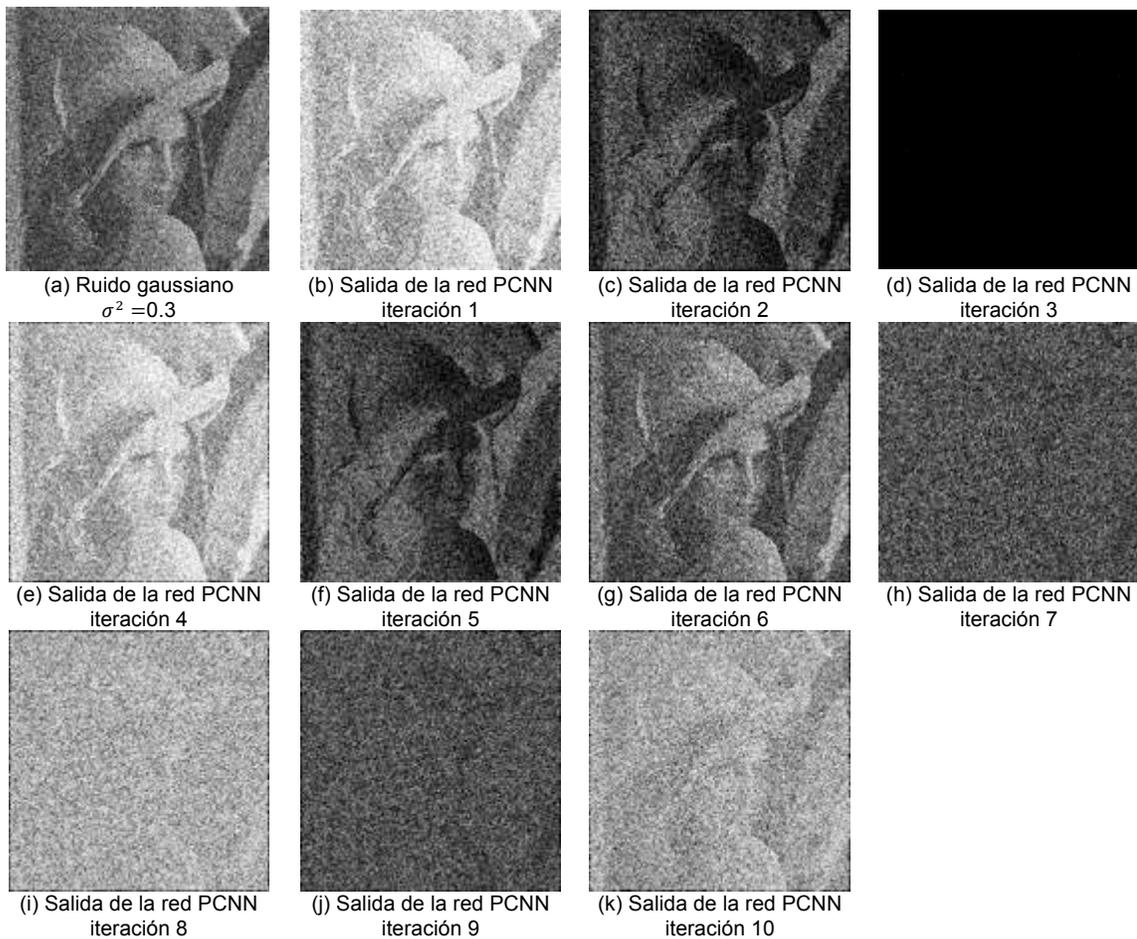


Figura 4.45 Lena con ruido gaussiano $\sigma^2 = 0.3$ procesada mediante la red PCNN

4.5.13 Implementación de la red ICM

Se implementó la red simplificada ICM en la plataforma de experimentación de acuerdo al algoritmo de la Tabla 4.2 con valores dados en [Ma, 2010] para las constantes de ajuste $f = 0.9$; $g = 0.8$; $h = 20$; $W_{ijkl} = [0.5 \ 1 \ 0.5; \ 1 \ 0 \ 1; \ 0.5 \ 1 \ 0.5]$. Se procesó una imagen de Lena en escala grises de 512x512 píxeles con 10% de ruido impulsivo, se realizaron 10 iteraciones de la red ICM en las que se observaron las imágenes binarias de salida para estudiar los cambios producidos en cada iteración.

En la sección ampliada de la imagen procesada (Figura 4.46) se observó que en la primera iteración, de modo similar a las salidas de la red PCNN, se activaron todas las neuronas excepto las correspondientes a píxeles contaminados con ruido pimienta. En la segunda iteración se activaron todos los píxeles, excepto los de los bordes que no fueron procesados. En la tercera iteración el patrón de activación fue contrario al de la primera iteración, destacando así los píxeles con ruido pimienta por la activación de sus neuronas respectivas.

De la cuarta a la novena iteración las salidas no mostraron pixeles activos, en la novena iteración se observó la activación de las neuronas correspondientes a los pixeles con ruido sal, y finalmente en la décima iteración se obtuvieron patrones de activación que no corresponden a los pixeles con ruido impulsivo.

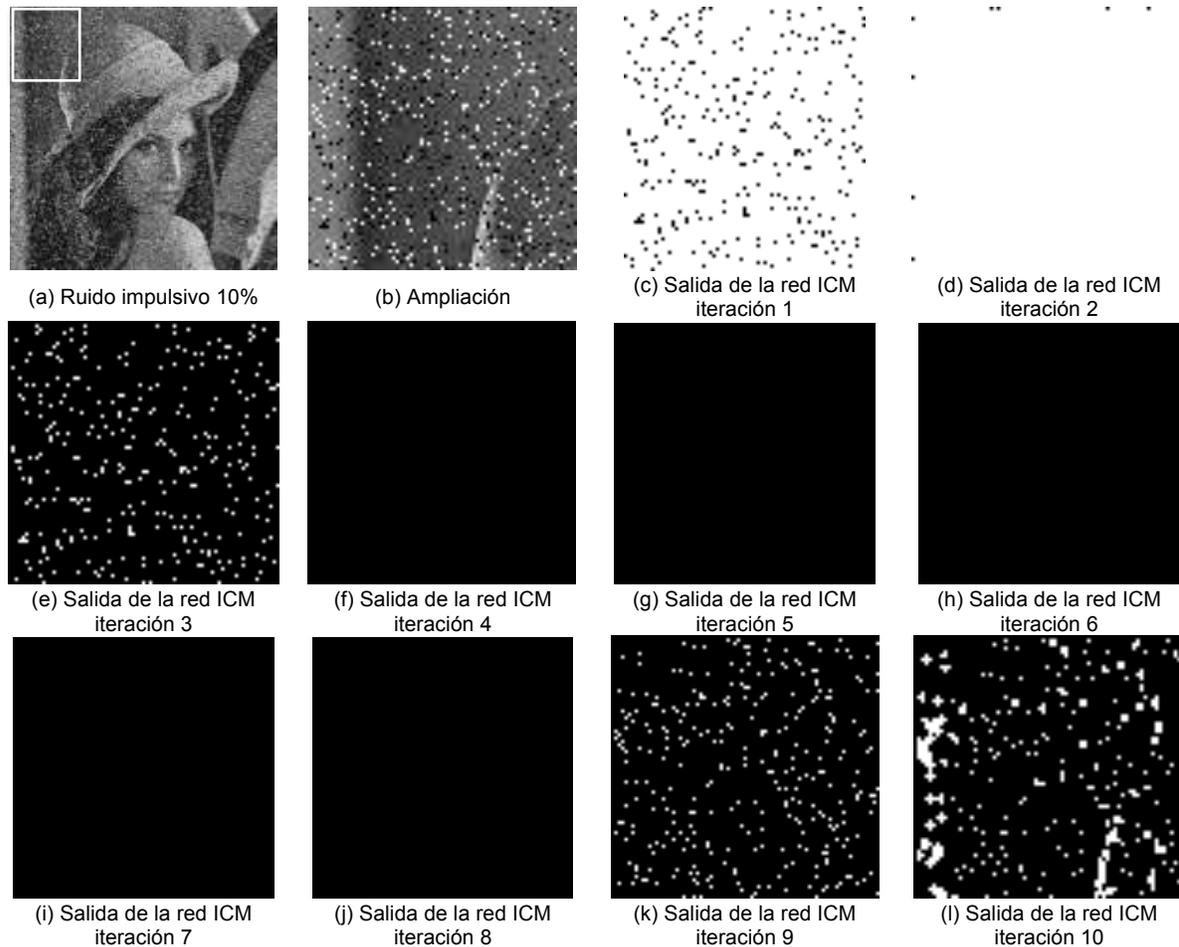


Figura 4.46 Imagen con 10% de ruido impulsivo procesada con la red ICM

Dado el comportamiento de la red ICM y su habilidad para destacar los pixeles ruidosos en distintas iteraciones, se aplicó el algoritmo de filtrado selectivo de la Tabla 4.2, que consistió en utilizar el operador de mediana de 3x3 sobre los pixeles que se activaron en cada iteración, de modo que se obtuvieron imágenes filtradas de manera subsecuente (Figura 4.47).

Se observó que en la primera iteración se activaron todos los pixeles de la salida, excepto los que contienen ruido pimienta, dado el criterio de filtrado ningún pixel fué modificado; en la segunda iteración todos los pixeles de la salida se activaron y ningún pixel de la imagen fue modificado.

En la tercera iteración se activaron los pixeles correspondientes al ruido pimienta, por lo que al aplicar el operador mediana en las posiciones respectivas de la imagen con ruido se obtiene una imagen sin ruido pimienta.

No se observaron cambios en la imagen filtrada de la cuarta a la octava iteración, no obstante en la novena iteración se activaron los pixeles correspondientes al ruido sal y se observa que en la imagen filtrada ya no contiene pixeles con ruido, si se continúa filtrando la imagen en las posiciones activas se ven modificados de manera negativa los pixeles correspondientes a líneas y bordes.



Figura 4.47 Imágenes filtradas selectivamente por el operador de mediana utilizando la salida de la red ICM

Se repitió la prueba incrementando la probabilidad de ocurrencia del ruido impulsivo de 0.1 a 0.9, se muestran los resultados del filtrado en las iteraciones con activaciones relevantes, la primera, tercera y novena iteraciones (Figura 4.48 y Figura 4.49). Se observó que para las imagen con más de 20% de ruido impulsivo ya no se logran filtrar todos los pixeles con ruido. Se obtuvo una imagen filtrada eficazmente hasta una probabilidad de ruido de 50%, para ruido mayor la recuperación es pobre.

La ventaja de filtrar selectivamente sin afectar los bordes y líneas se mantiene para las imágenes con los distintos niveles de ruido.



Figura 4.48 Imágenes de 0 a 50% de ruido impulsivo filtradas selectivamente utilizando la salida de la red ICM

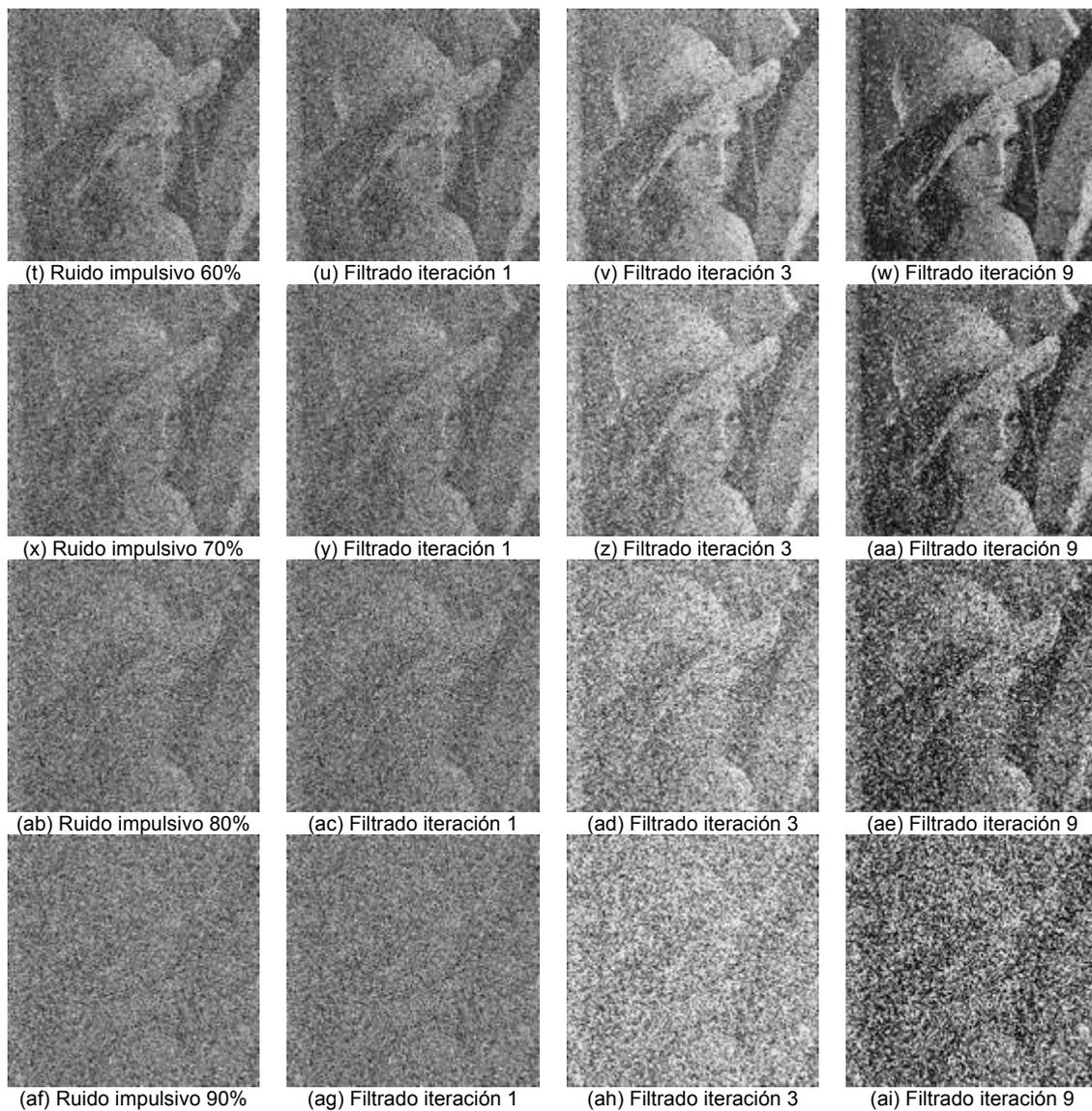


Figura 4.49 Imágenes de 60 a 90% de ruido impulsivo filtradas selectivamente utilizando la salida de la red ICM

Para determinar el mecanismo por el cual se activan las neuronas correspondientes al ruido impulsivo se generó una serie de experimentos en Matlab, se realizó el muestreo en ventanas de 3x3 de la imagen original de *Lena* de 512x512 píxeles con 10% de ruido (Figura 4.50) que se procesó con la red ICM durante 15 iteraciones para observar los valores de las matrices de la red.

Las ventanas muestreadas corresponden al píxel (7,15) que no está afectado por ruido, el píxel (7,10) que tiene ruido pimienta y el píxel (9,6) que tiene ruido sal, se analizó la vecindad de 8 píxeles de cada uno.

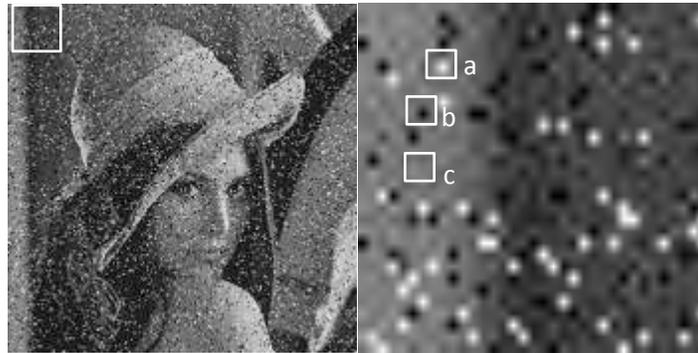


Figura 4.50 Toma de muestra para experimentos con ICM

Para cada vecindad de 9 píxeles se obtuvieron y graficaron los valores correspondientes a la imagen de entrada normalizada S_{ij} , como son: el potencial de su neurona correspondiente F_{ij} , el umbral dinámico E_{ij} y la salida de la red Y_{ij} tomados de las ventanas muestreadas cuyos valores en la primera iteración pueden ser como los que se muestran en la Figura 4.51.

S_{ij}			F_{ij}			E_{ij}			Y_{ij}		
0.18	0.15	0.14	0.18	0.15	0.14	0	0	0	1	1	1
0.15	0.15	0.13	0.15	0.15	0.13	0	0	0	1	1	1
0.12	0.24	0.15	0.12	0.24	0.15	0	0	0	1	1	1

Figura 4.51 Valores de ejemplo de la ICM para la imagen de muestra

Se muestra en primer lugar la vecindad ampliada del píxel sin ruido (7,15) y sus respectivos valores de gris (Figura 4.52).

182	159	145
154	151	132
129	146	156

Figura 4.52 Valores de gris de la muestra examinada

Se obtuvieron los patrones de activación para 15 iteraciones de la red ICM (eje horizontal), los valores para cada píxel de la vecindad se encuentran en la posición respectiva de la figura. Para cada gráfico se observan tres señales, donde la salida Y_{ij} se denota por el cuadro relleno, el umbral E_{ij} por el cuadro vacío en negro y el potencial interno F_{ij} por el cuadro vacío en rojo (Figura 4.53).

De este experimento se observó que la salida Y_{ij} de todas las neuronas se activa en la primera iteración, si $S_{ij} \neq 0$ y $F_{ij} > E_{ij}$, esto provoca que E_{ij} crezca rápidamente y a partir de ahí comienza a decaer lentamente. Luego F_{ij} va creciendo nuevamente hasta la sexta y séptima iteración donde $F_{ij} > E_{ij}$ entonces $Y_{ij} = 1$ para un segundo ciclo de activaciones.

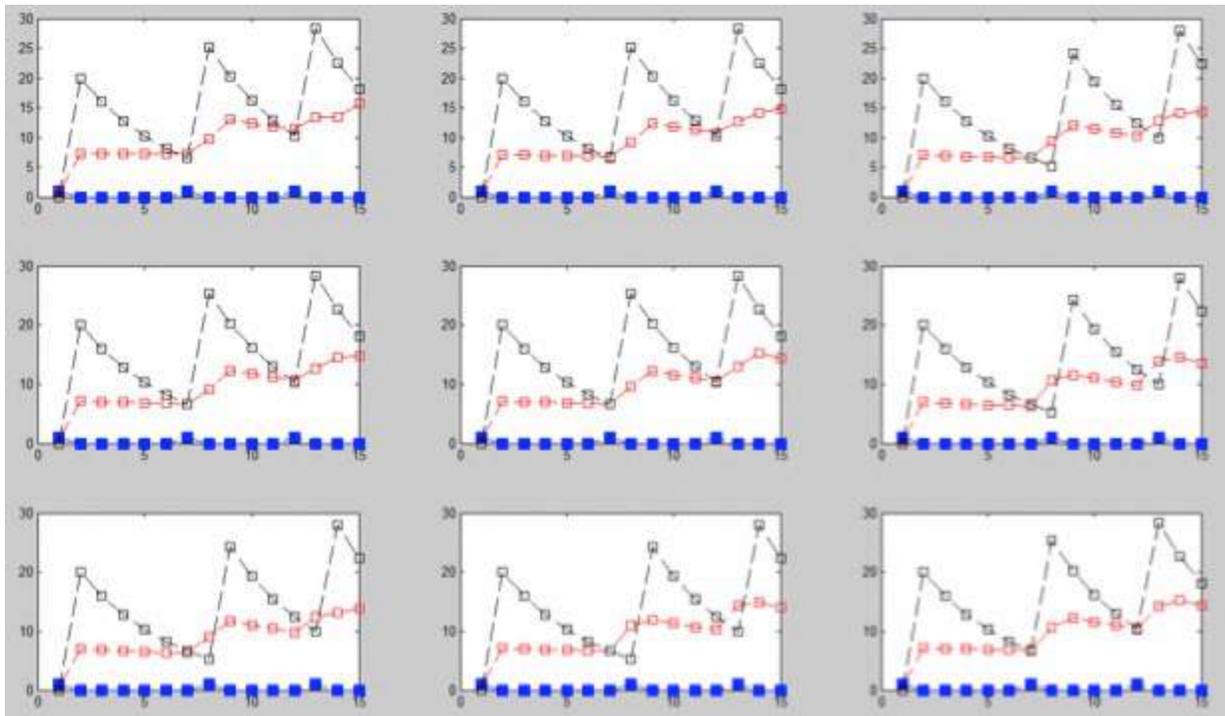


Figura 4.53 Patrones de activación con ICM de la muestra examinada

Se muestra en segundo lugar la vecindad ampliada del pixel con ruido pimienta (7,10) y sus respectivos valores de gris (Figura 4.54).

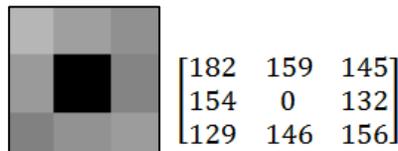


Figura 4.54 Valores de gris de la muestra con ruido pimienta

Se obtuvieron también los patrones de activación para 15 iteraciones de la red ICM (eje horizontal), donde la salida Y_{ij} se denota por el cuadro relleno, el umbral E_{ij} por el cuadro vacío en negro y el potencial interno F_{ij} por el cuadro vacío en rojo (Figura 4.55).

Se observó que la salida de la neurona que corresponde al ruido pimienta no se activó en la primera iteración donde $S_{ij} = 0$, mientras que la salida de sus vecinas por el contrario se activa, la salida de dicha neurona se activa en la segunda iteración cuando el valor del potencial ha superado al umbral, cuando ello ocurre el valor del umbral toma su máximo. La segunda serie de activaciones ocurre en distinto tiempo para los pixeles con ruido y sin ruido en la ventana considerada.

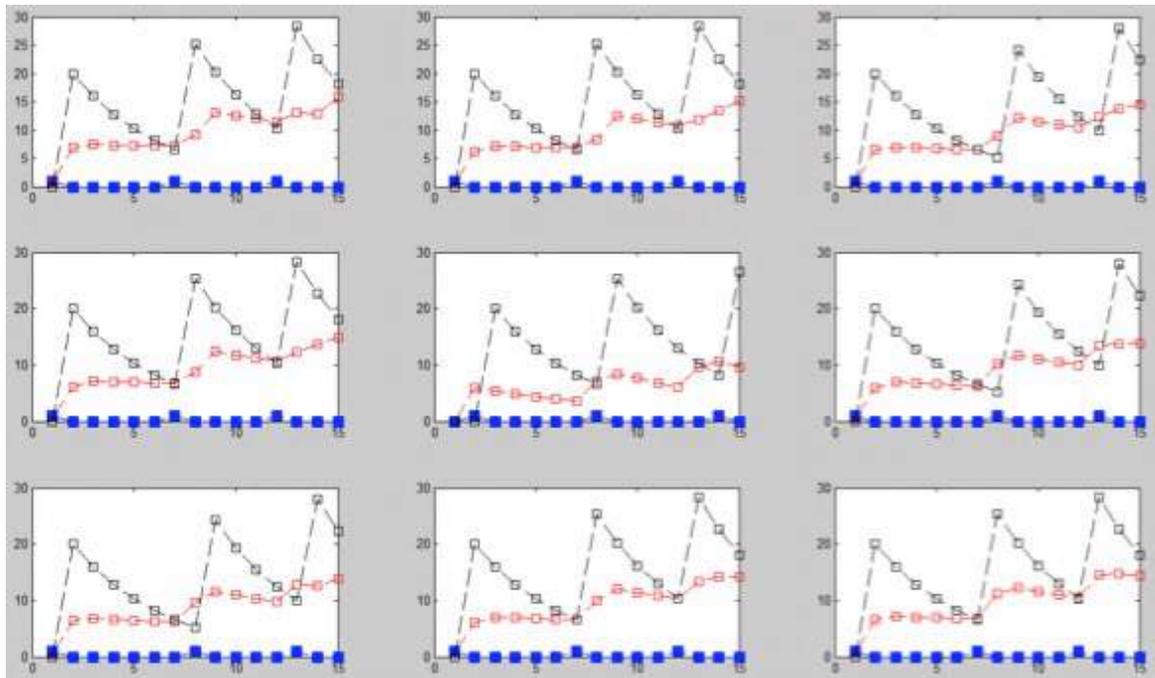


Figura 4.55 Patrones de activación con ICM de la muestra con ruido pimienta

Se muestra en tercer lugar la vecindad ampliada del pixel con ruido sal (9,6) y sus respectivos valores de gris (Figura 4.56).

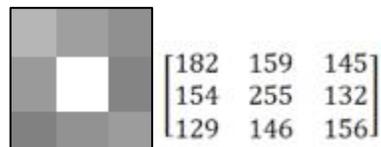


Figura 4.56 Valores de gris de la muestra con ruido sal

Se obtuvieron también los patrones de activación para 15 iteraciones de la red ICM (eje horizontal), donde la salida Y_{ij} se denota por el cuadro relleno, el umbral E_{ij} por el cuadro vacío en negro y el potencial interno F_{ij} por el cuadro vacío en rojo (Figura 4.57). Se observó que la neurona se activa en la primera iteración al igual que sus vecinas, sin embargo en una segunda serie de activaciones el pixel con ruido sal es el que activa su salida antes que sus vecinos lo que permite identificarlo.

Se encontraron diferencias en la activación de las neuronas correspondientes a pixeles con ruido pimienta entre la plataforma de experimentación, donde los pixeles se destacan en la tercera iteración, y en Matlab donde se activan en la segunda. Para el caso del ruido sal, las activaciones se generaron en la novena iteración mientras que en Matlab se dieron en la sexta, ello debido al modo de inicialización de las matrices y el uso de funciones propias de Matlab para algunos cálculos.

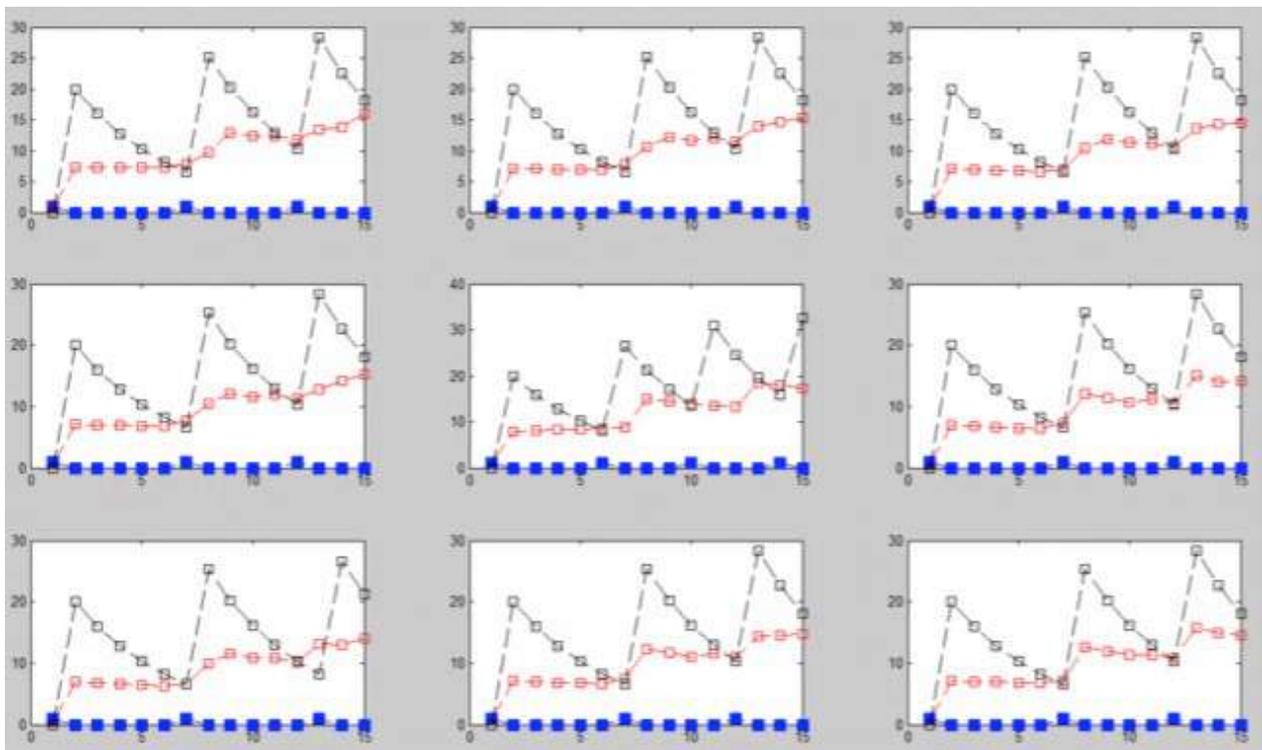


Figura 4.57 Patrones de activación con ICM de la muestra con ruido sal

4.5.14 Implementación de la red SCM

Se implementó la red SCM en la plataforma de experimentación de acuerdo al algoritmo de la Tabla 4.4 con valores dados en [Ma, 2010] para las constantes de ajuste $f = 0.9$; $g = 0.8$; $h = 20$; $\gamma = 1$; $W_{ijkl} = [0.5 \ 1 \ 0.5; 1 \ 0 \ 1; 0.5 \ 1 \ 0.5]$. Se procesó una imagen de *Lena* en escala grises de 512x512 píxeles con 10% de ruido impulsivo, se realizaron 10 iteraciones de la red SCM en las que se observaron las imágenes binarias de salida para estudiar los cambios producidos en cada iteración (Figura 4.58).

Se observó que en la primera iteración destacan los píxeles con ruido pimienta por no tener salidas activas, mientras que en la segunda iteración ninguna neurona tiene salidas activas. En la tercera iteración se observó que se activaron algunos píxeles correspondientes al ruido impulsivo de sal y en la cuarta iteración se activaron otros píxeles con ruido sal, algunos de manera redundantes con los de la tercera iteración.

Las iteraciones a partir de la quinta muestran patrones de activación correspondientes a los bordes y regiones de la imagen, es decir que no corresponden a los píxeles con ruido impulsivo.

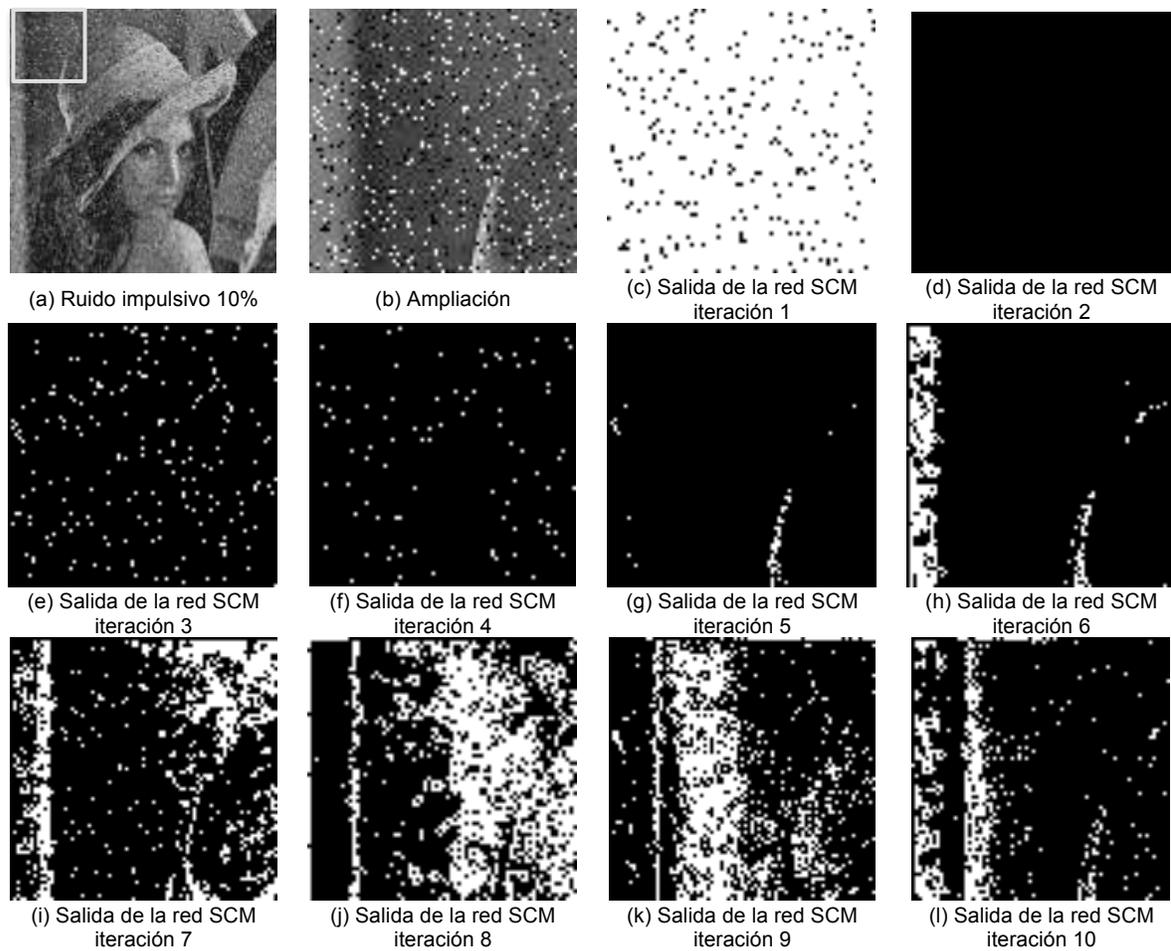


Figura 4.58 Imagen con 10% de ruido impulsivo procesada con la red SCM

Dado el comportamiento de la red SCM y su habilidad para destacar los píxeles ruidosos en distintas iteraciones, se observó que el ruido pimienta se destaca por píxeles inactivos y el ruido sal por píxeles activos.

Se aplicó el algoritmo de filtrado selectivo de la Tabla 4.2 modificado para utilizar el operador de mediana de 3x3 sobre los píxeles que no se activaron en la primera iteración y luego sobre los píxeles que sí se activaron en las iteraciones subsecuentes, obteniéndose imágenes filtradas de *Lena* con 10% de ruido impulsivo (Figura 4.59).

En la imagen se observa que el ruido disminuye en la primera iteración, hasta obtener la imagen de la cuarta iteración, a partir de entonces el operador de mediana afecta no solo a píxeles con ruido sino a bordes y líneas propias de la imagen contribuyendo a su deterioro y pérdida de nitidez.



Figura 4.59 Imágenes filtradas selectivamente por el operador de mediana utilizando la salida de la red SCM

Se repitió la prueba de filtrado selectivo a partir del procesamiento de la imagen con ruido mediante la red SCM incrementando la probabilidad de ocurrencia del ruido impulsivo de 0.1 a 0.9, se muestran los resultados del filtrado en las iteraciones con activaciones relevantes, que son la primera, tercera y cuarta iteraciones (Figura 4.60 y Figura 4.61).

Para una imagen sin ruido no hay activaciones que permitan aplicar el criterio de filtrado selectivo en las primeras cinco iteraciones, por lo que la imagen permanece sin cambios.

Se observó que para la imagen con 10% de ruido éste prácticamente ya no es visible, mientras que para la imagen con más de 20% de ruido impulsivo no se logran filtrar a todos los píxeles con ruido. Se obtuvo una imagen filtrada eficazmente hasta una probabilidad de ruido de 50%, para ruido mayor la recuperación es pobre.

La ventaja de filtrar selectivamente sin afectar los bordes y líneas se mantiene para las imágenes con los distintos niveles de ruido; y la utilización del operador de mediana para completar el filtrado hace notar su superioridad para eliminar el ruido impulsivo de probabilidad menor al 60%, a partir de ese punto se puede considerar otra técnica complementaria de filtrado.



Figura 4.60 Imágenes con ruido impulsivo con probabilidad de ocurrencia de 0 a 0.4 filtradas selectivamente con el operador de mediana utilizando la salida de la red SCM

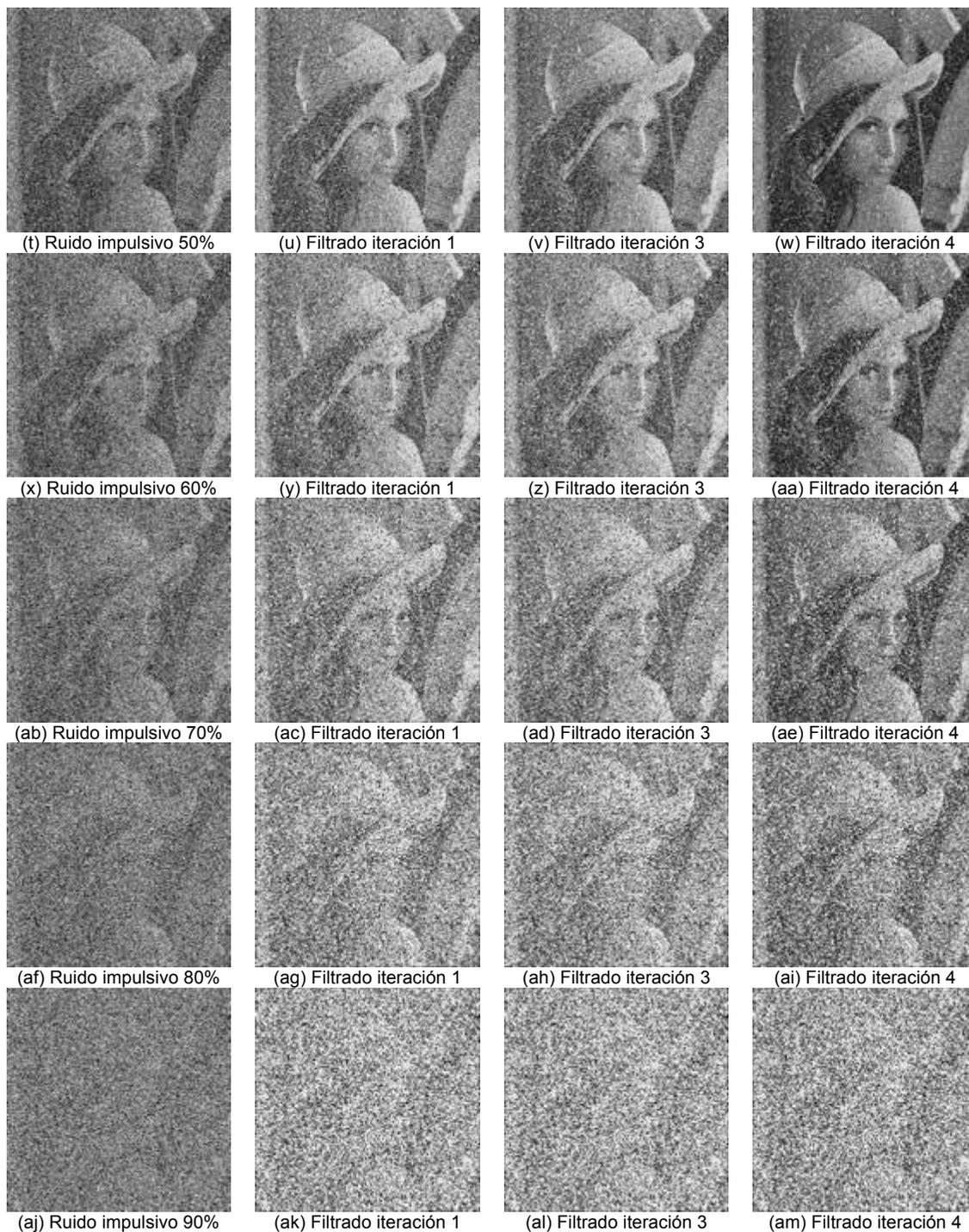


Figura 4.61 Imágenes con ruido impulsivo con probabilidad de ocurrencia de 0.5 a 0.9 filtradas selectivamente con el operador de mediana utilizando la salida de la red SCM

Se estudió el número de iteraciones necesarias para detectar el ruido impulsivo comparando las redes ICM y SCM utilizadas en combinación con el operador de mediana aplicado de manera selectiva. Se calculó el PSNR de las imágenes filtradas en cada iteración (Figura 4.62), utilizando las imágenes de *Lena* de 512x512 píxeles con 10% de ruido impulsivo.

En la gráfica se observa que dados los parámetros para cada tipo de red neuronal, la red ICM requiere nueve iteraciones para eliminar la mayor parte del ruido, cinco más que la red SCM y es capaz de restaurar mayor cantidad de píxeles con ruido impulsivo superando a la red SCM aproximadamente por 2dB, la imagen con ruido representa el valor de PSNR de la imagen sin filtrar.

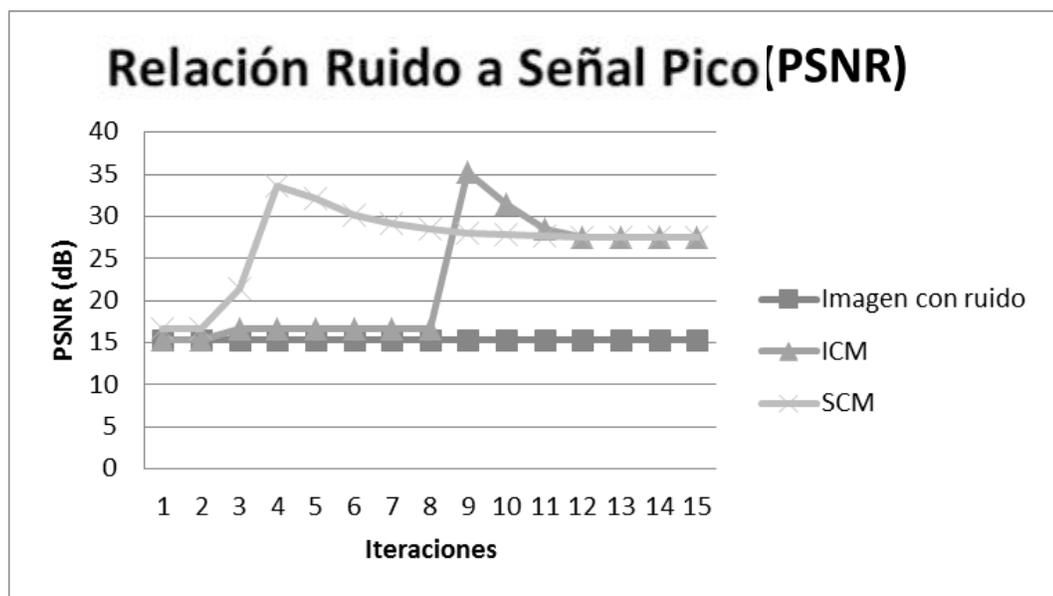


Figura 4.62 Relación entre iteraciones y recuperación de la imagen para ICM y SCM

Una vez que se identifica en qué iteración se puede detener el cálculo de las matrices de la red, se puede establecer un método de indexado de píxeles con ruido y realizar solo una operación de filtrado global para la imagen.

4.5.15 Implementación de la red PCNNNI

Finalmente se implementó la red PCNNNI y la matriz de tiempos en la plataforma de experimentación de acuerdo al algoritmo mostrado en Tabla 4.5 con los valores dados en [Ma, 2007] para $\alpha^F = 0.5$; $V_F = 0.5$; $\alpha^E = 0.5$; $V_E = 5.0$, $E_0 = 10$, $L = 255$, $\Delta = 15$.

Se hicieron pruebas con la imagen de *Lena* en escala de grises de 512x512 píxeles con ruido gaussiano de varianza 0.3 (Figura 4.63); inicialmente se cambió el rango dinámico de la imagen, ya que sin el cambio se requería un gran número de iteraciones para llenar la Matriz de Tiempos, una vez que se hizo el cambio sólo se requirieron 10 iteraciones.

Se observó que en las primeras iteraciones se activan todas o ninguna de las neuronas, posteriormente se generan patrones de activación no determinados, por lo que sólo con las salidas de la red no es posible aplicar una técnica directa de filtrado.

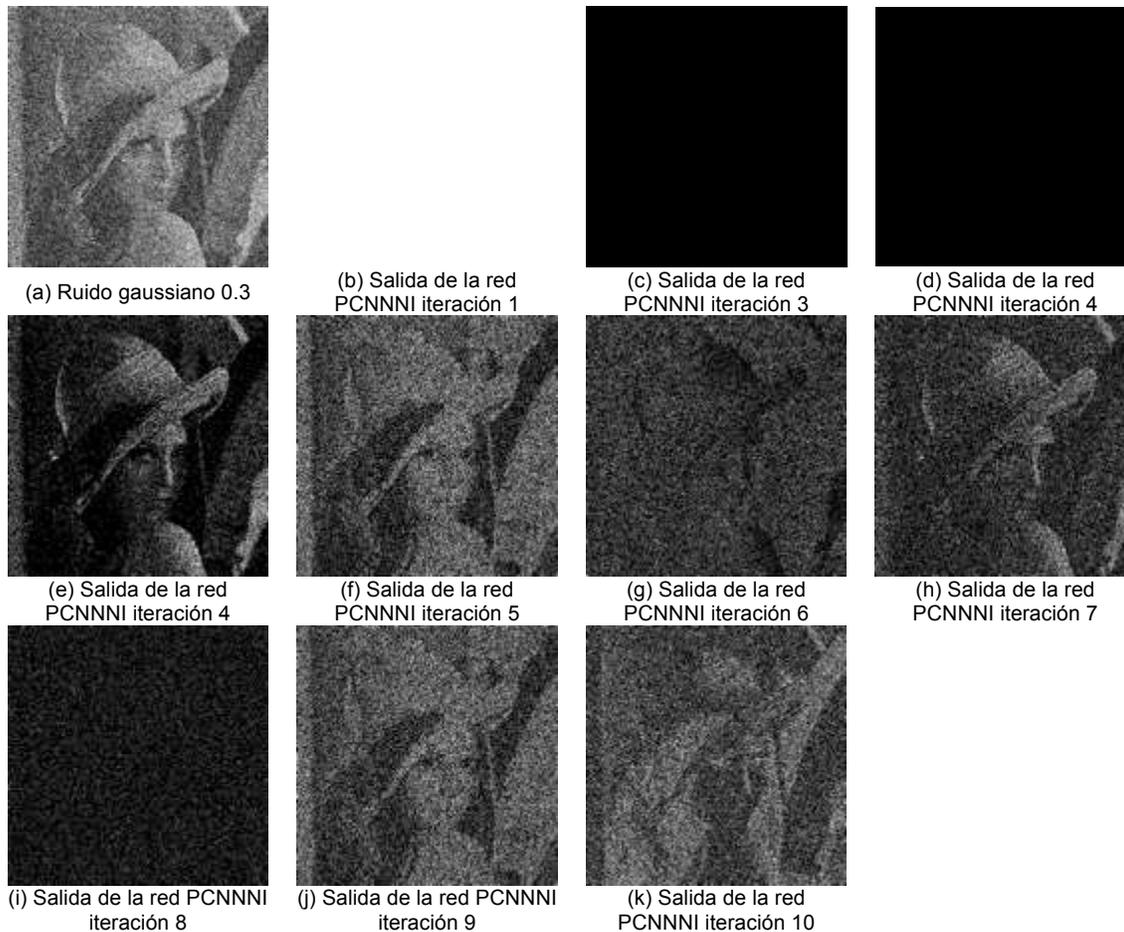


Figura 4.63 Salidas binarias de la red PCNNNI para una imagen con ruido gaussiano

Se realizó una prueba con la imagen de *Peppers* en escala de grises de 512x512 píxeles con el fin de ilustrar el proceso de filtrado de ruido gaussiano mediante el algoritmo basado en PCNNNI, para lo cual se simuló ruido gaussiano de $\mu = 0$ y $\sigma^2 = 0.01$. Se muestra una sección ampliada de una imagen sin ruido y con ruido (Figura 4.64).

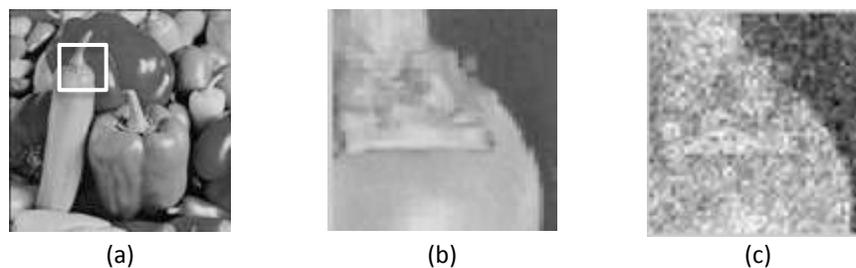


Figura 4.64 Imagen de a) *Peppers* sin ruido, b) ampliada, c) con ruido gaussiano de $\sigma^2 = 0.01$

En la Figura 4.65 se observan las salidas de las primeras diez iteraciones de la red ante la imagen sin ruido. En la primera iteración todas las neuronas se activan puesto que el valor de los píxeles es distinto de cero, por lo que se ignora ésta en la formación de la Matriz de Tiempos. En este conjunto de imágenes también se observa la formación de regiones y bordes.

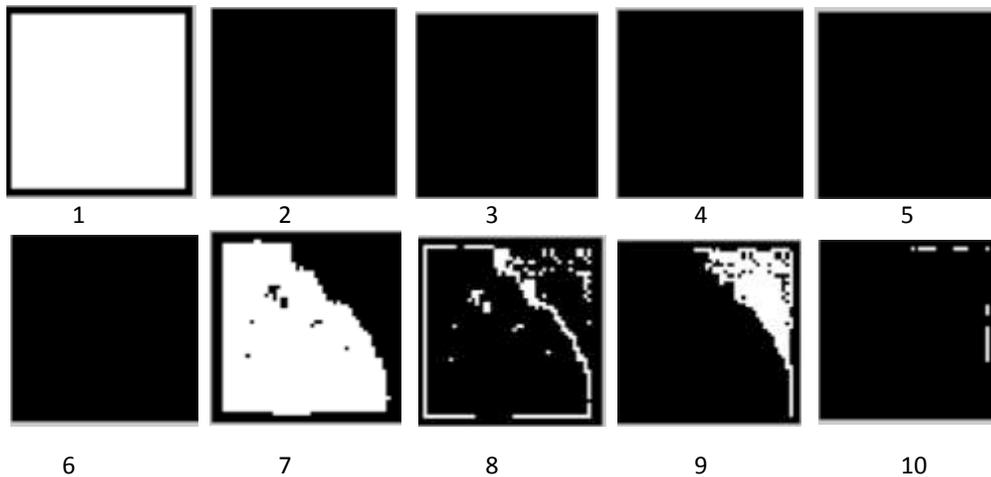


Figura 4.65 Imágenes de salida de la red PCNNNI, iteraciones 1 a 10 para imagen sin ruido

En la Figura 4.66 se muestran las salidas de la red PCNNNI correspondientes a las iteraciones 1 a 10 para la imagen con ruido gaussiano. Se observa que los patrones de activación son dispersos, por lo que ya no se forman regiones o bordes claros.

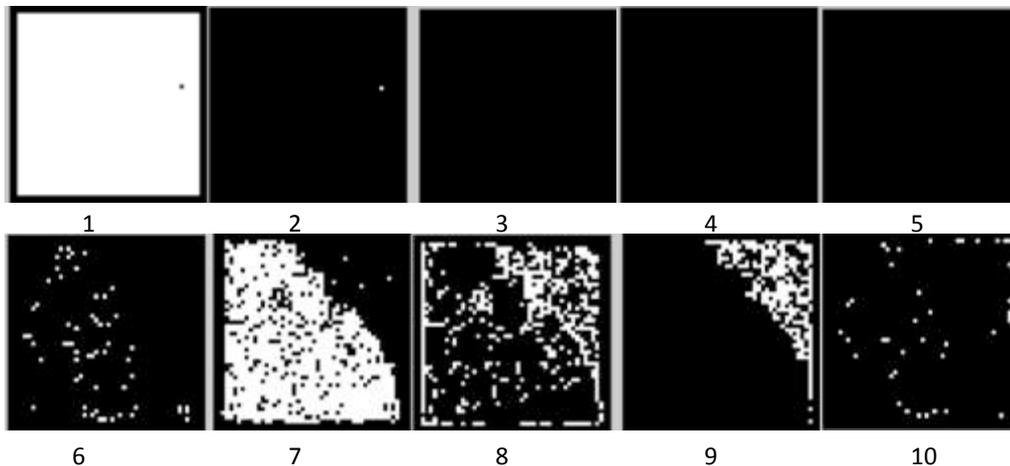


Figura 4.66 Imágenes de salida de la red PCNNNI, iteraciones 1 a 10 para imagen con ruido

4.4 Discusión

Una vez que se logró simular el ruido impulsivo y gaussiano en las imágenes se realizaron pruebas de filtrado con técnicas tradicionales, se obtuvieron las métricas NMSE, MAE y PSNR. Del mismo modo se determinó el tiempo de procesamiento de las distintas técnicas de filtrado tradicionales para las imágenes de prueba.

Se realizó la reimplementación de tres modelos de redes neuronales PCNN, ICM y SCM para procesar imágenes con ruido impulsivo y se implementó el modelo PCNNNI para imágenes con ruido gaussiano tal como están descritas en [Ma, 2010], donde se indican todos los parámetros de las redes implementadas pero no se manifiesta su capacidad de detección, la respuesta de los algoritmos a varianzas de ruido gaussiano superiores a 0.01 y no se indica el tiempo de procesamiento ni el criterio de paro.

En el siguiente capítulo se experimentará con los algoritmos de filtrado basados en PCNN para variar sus parámetros, determinar su tiempo de procesamiento y obtener las métricas de recuperación PSNR, MAE y NMSE para compararlas con otras técnicas de filtrado y de detección de impulsos.

Algunas investigaciones recientes apuntan hacia la utilización de memorias asociativas que reconstruyen imágenes o reconocen imágenes aunque exista ruido presente [Tang, 2007], no obstante el estado del arte y la experimentación indican que las Redes Pulso-Acopladas tienen potencial en la detección de los píxeles ruidosos, como se ha podido comprobar para el caso del ruido impulsivo, o bien proporcionan información temporal sobre la activación de los píxeles ruidosos, como se constató para el caso del ruido gaussiano.

Capítulo 5 Experimentación y resultados

En este capítulo se muestran las imágenes resultantes del filtrado de ruido impulsivo y gaussiano utilizando técnicas tradicionales y mediante técnicas basadas en PCNN, también se incluyen resultados sobre el efecto de la variación de parámetros en el filtrado apoyado en las redes PCNN, y finalmente pruebas de filtrado de ruido impulsivo y gaussiano mezclados.

5.1 Detalles y objetivos de la experimentación

Base de datos de imágenes utilizada:

Se seleccionaron imágenes estándar en formato BMP y escala de grises, de tamaño 512x512 píxeles como *Lena*, *Peppers*, *Baboon*, *Cameraman* y *Grass* [ImageProcessingPlace, 2016; USC-SIPI, 1977].

Características de los filtros implementados en la plataforma de pruebas:

- Filtro de mediana de 3x3 y de 5x5.
- Filtro promedio de 3x3.
- Filtro morfológico generado con base en [Myler, 1993].
- Filtro Wiener utilizando la función de Matlab `wiener2` [Matworks, 2014].
- Algoritmo de PCNN, con valores de $\alpha_F = 0.1$; $\alpha_L = 1.0$; $\alpha_E = 1.0$; $V_L = 0.2$; $V_E = 20.0$; $V_F = 0.5$; $\beta = 0.1$, $M_{ijkl} = W_{ijkl} = [0.5 \ 1 \ 0.5; \ 1 \ 0 \ 1; \ 0.5 \ 1 \ 0.5]$ [Ma, 2010].
- Algoritmo de ICM, con valores para las constantes de ajuste $f = 0.9$; $g = 0.8$; $h = 20$; $W_{ijkl} = [0.5 \ 1 \ 0.5; \ 1 \ 0 \ 1; \ 0.5 \ 1 \ 0.5]$ [Ma, 2010].
- Algoritmo de SCM, con valores de $f = 0.9$; $g = 0.8$; $h = 20$; $\gamma = 1$; $W_{ijkl} = [0.5 \ 1 \ 0.5; \ 1 \ 0 \ 1; \ 0.5 \ 1 \ 0.5]$ [Ma, 2010].
- Algoritmo de PCNNNI y la matriz de tiempos, con valores de $\alpha_F = 0.5$; $v_F = 0.5$; $\alpha_E = 0.5$; $v_E = 5.0$ [Ma, 2010].
- Filtro de ruido impulsivo basado en ICM combinado con el filtro de mediana [Ma, 2010].
- Filtro de ruido gaussiano basado en PCNNNI y la matriz de tiempos [Ma, 2010].
- Los valores iniciales de las matrices son cero, y las imágenes se normalizan de 0 a 1.

La realización de experimentos se dividió en técnicas para tratar ruido impulsivo y para ruido gaussiano. En cada categoría se efectuaron pruebas de variación de parámetros a las redes PCNN, respuesta al incremento del nivel de ruido y con distintas las imágenes de prueba.

5.2 Filtrado tradicional y basado en PCNN para tratar ruido impulsivo

5.2.1 Comparación entre técnicas de filtrado de ruido impulsivo

En la plataforma de experimentación se obtuvieron imágenes con ruido impulsivo, filtradas mediante técnicas tradicionales y basadas en PCNN. En el ejemplo de la Figura 5.1 se muestra una imagen de *Lena* con 10% de ruido impulsivo. La imagen se filtró mediante el filtro promedio que no eliminó el ruido impulsivo y sólo provocó un efecto de suavizado en la imagen; con el filtro de mediana se observó que si eliminó el ruido pero difuminó bordes y líneas; y por otro lado con el filtro morfológico que redujo el ruido impulsivo pero afectó los píxeles sin ruido. Se observó que las técnicas basadas en PCNN eliminaron el ruido y preservaron los detalles, en este caso la red ICM elimina mayor grado de ruido que la red SCM.



Figura 5.1 a) Imagen con ruido 10%, filtrada con b) filtro promedio, c) filtro de mediana, d) filtro morfológico, e) filtro con SCM, f) filtro con ICM

Comparando las técnicas de filtrado por medio de la métrica PSNR de las imágenes restauradas de *Lena* con distintas probabilidades de ocurrencia se observa que los métodos ICM y SCM filtran el ruido impulsivo de probabilidad menor al 30% con una recuperación de 2 dB mayor que los métodos tradicionales, y con recuperación de 20 dB con respecto a la imagen sin filtrar (Figura 5.2). Los métodos ICM y SCM alteran en menor grado una imagen sin ruido debido a que discriminan píxeles ruidosos y no ruidosos.

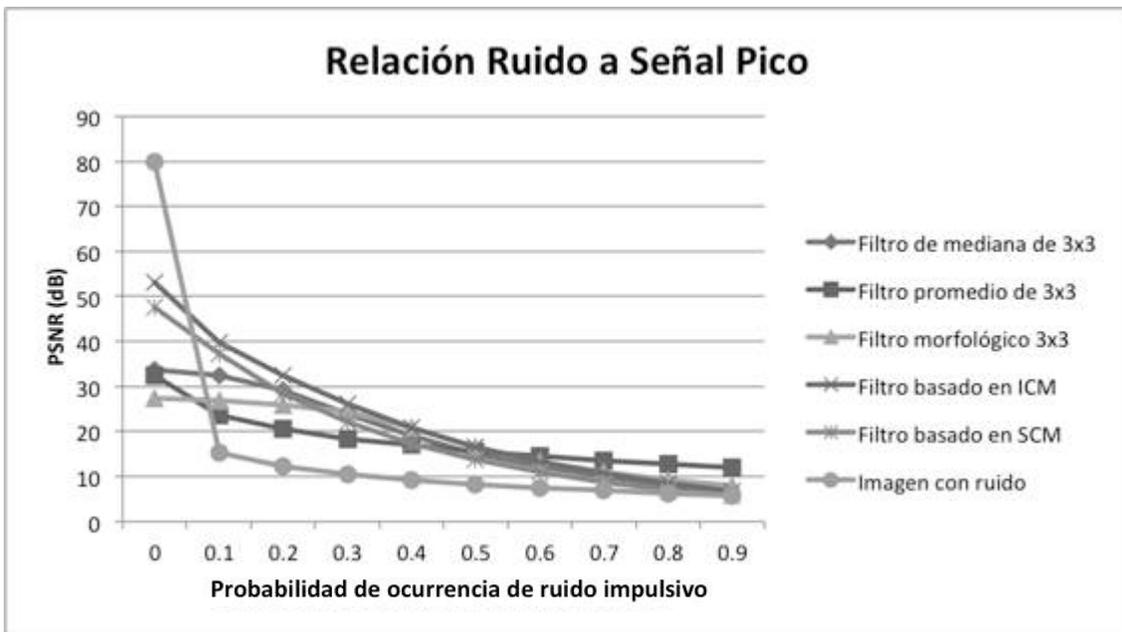


Figura 5.2 Relación entre probabilidad de ruido y PSNR

Se obtuvo de igual forma la métrica MAE para *Lena* con distintas probabilidades de ocurrencia (Figura 5.3). Se aprecia que para probabilidades de ruido superiores al 50% ningún método es suficientemente bueno para eliminarlo. El filtro promedio tiene una capacidad de recuperación lineal cuando la probabilidad del ruido aumenta.

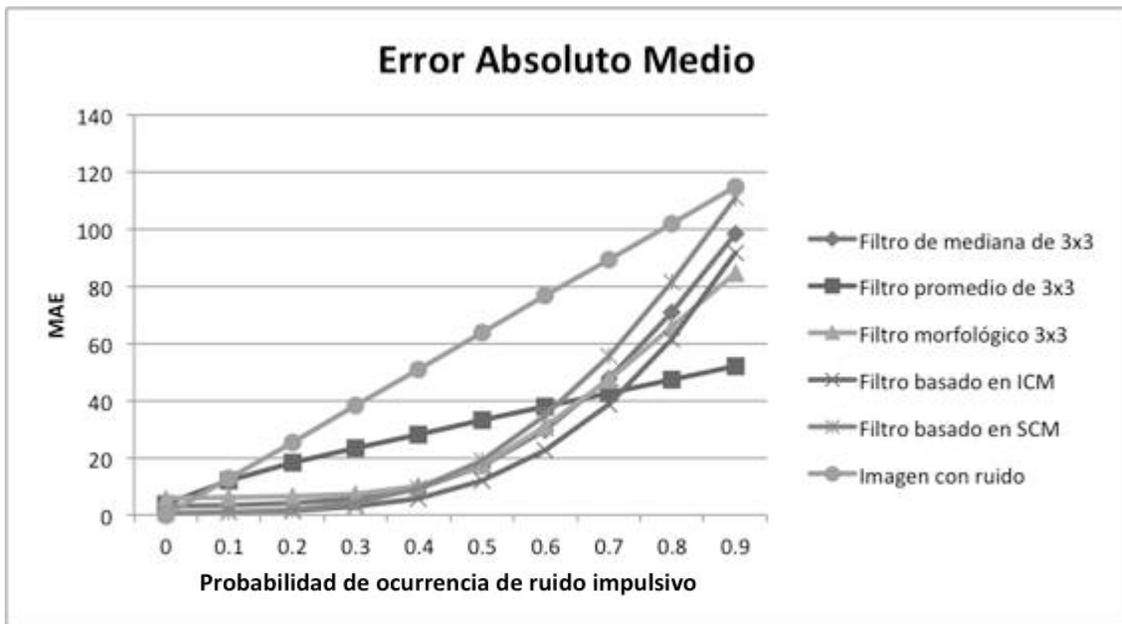


Figura 5.3 Relación entre probabilidad de ruido y MAE

También se calculó el NMSE para *Lena* con distintas probabilidades de ocurrencia de ruido impulsivo (Figura 5.4). Se hace notar que el filtro promedio generó menor error que otras técnicas conforme la probabilidad de ocurrencia del ruido aumenta.

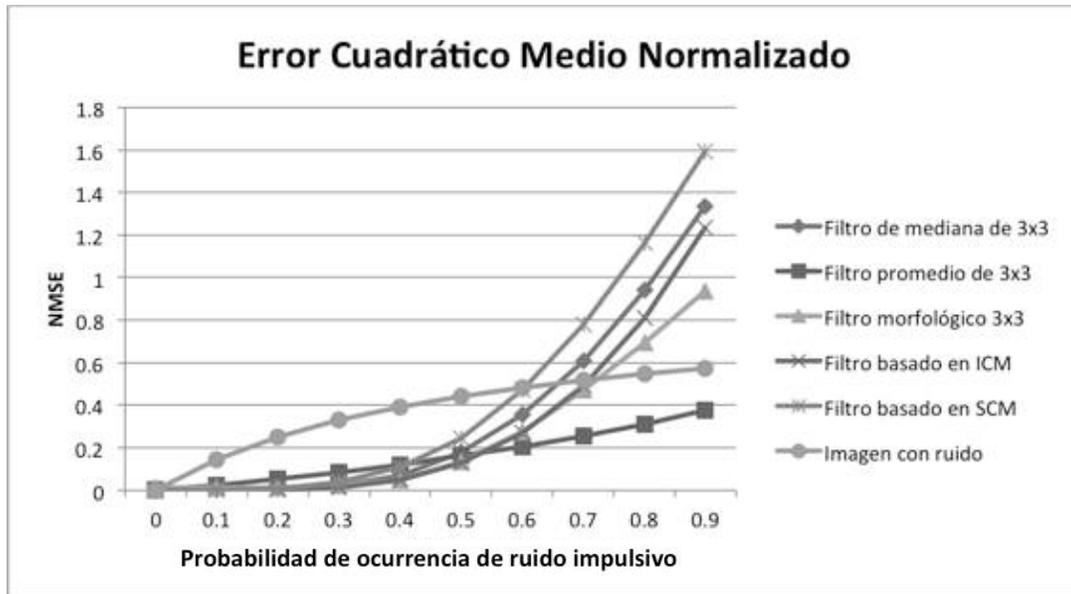


Figura 5.4 Relación entre probabilidad de ruido y NMSE

La medición del tiempo de procesamiento de los filtros con la imagen de *Lena* de 512x512 píxeles (Figura 5.5) arrojó que los filtros promedio y mediana tardan menos de 40 ms, el filtro morfológico tarda en promedio 300 ms, de las técnicas basadas en Redes Neuronales SCM tardan aproximadamente 235 ms y las redes ICM tardan hasta 20000 ms.

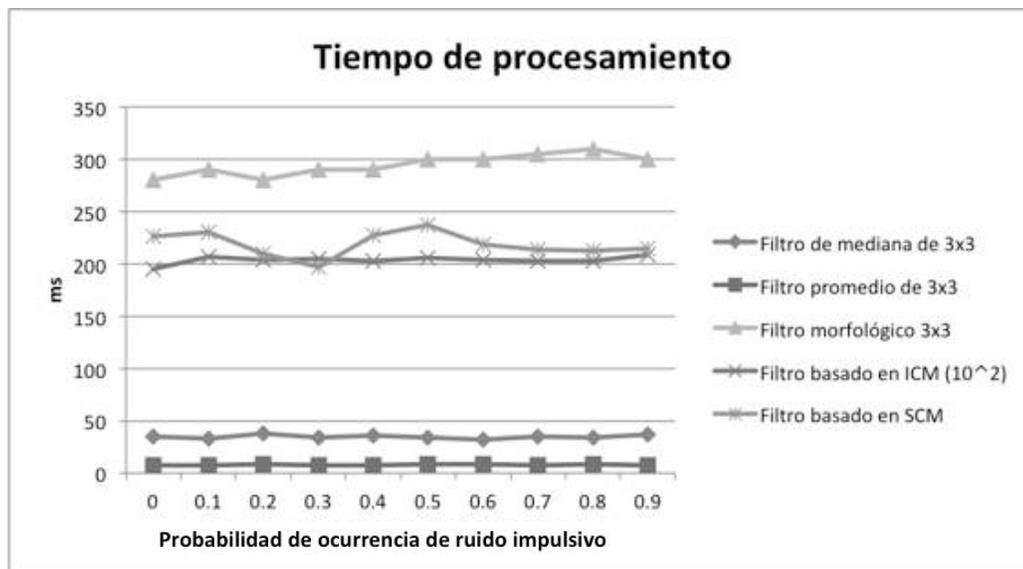


Figura 5.5 Relación entre probabilidad de ruido impulsivo y tiempo de procesamiento

En la Tabla 5.1 se observa un ejemplo de la recuperación de los filtros en términos del PSNR para imágenes con ruido impulsivo de probabilidad de ocurrencia 0.1 y 0.6, comparando numéricamente las técnicas tradicionales de filtrado contra las técnicas basadas en PCNN, la técnica basada en ICM es superior a todas las técnicas comparadas para las probabilidades consideradas, mientras que la técnica SCM supera al filtro de mediana para la probabilidad de ocurrencia de 10%.

Tabla 5.1 Capacidad de recuperación PSNR [dB] de los filtros de ruido impulsivo

Imagen, probabilidad de ocurrencia de ruido impulsivo	Imagen original	Filtro Mediana de 3x3	Filtro Promedio de 3x3	Filtro Morfológico	Filtro con ICM	Filtro con SCM
	PSNR (dB)					
<i>Lena</i> , 0.1	15.320	32.435	23.654	26.816	39.799	37.305
<i>Lena</i> , 0.6	7.507	12.214	14.561	13.320	13.380	10.943
<i>Baboon</i> , 0.1	15.605	23.041	20.936	20.061	31.454	36.810
<i>Baboon</i> , 0.6	7.771	11.989	15.101	9.094	13.206	11.803
<i>Peppers</i> , 0.1	15.303	33.571	23.745	26.963	40.714	38.475
<i>Peppers</i> , 0.6	7.513	12.223	14.602	9.715	13.385	11.544
<i>Camán.</i> , 0.1	15.062	34.606	23.477	24.813	40.630	38.222
<i>Camán.</i> , 0.6	7.267	11.938	13.841	11.341	13.089	11.242
<i>Grass</i> , 0.1	14.738	17.974	16.314	12.997	24.214	22.235
<i>Grass</i> , 0.6	6.938	9.907	11.797	8.031	10.965	9.874

5.2.2 Características del filtrado de ruido impulsivo al variar la imagen

Como experimento adicional se incluyó el filtrado de ruido impulsivo utilizando la imagen de *Lena* modificada en escala (Figura 5.6). El resultado es que la calidad de la imagen restaurada aumenta hasta en 17 dB conforme aumenta el tamaño de la imagen procesada de 32 a 1024 píxeles por lado para el caso del filtro de mediana, luego para las técnicas basadas en Redes Neuronales ICM y SCM aumenta en 14 dB para el mismo cambio de tamaño, por lo tanto el filtrado tradicional y basado en PCNN es sensible al escalado de la imagen.

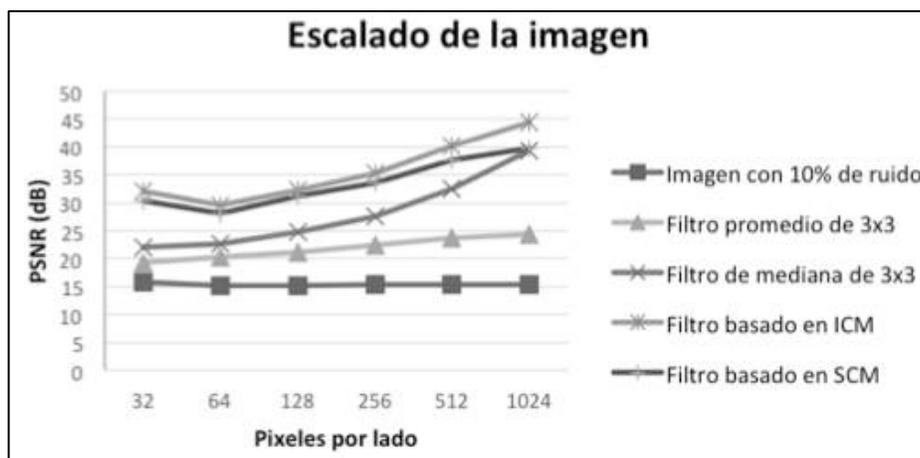


Figura 5.6 Respuesta de los filtros al escalado de la imagen

Se determinó la calidad del filtrado cuando la imagen es rotada, desplazada y puesta en espejo (Figura 5.7). Aquí se observa que la recuperación de la imagen denotada por PSNR se mantiene aun cuando la imagen procesada es rotada, trasladada y reflejada tanto para las técnicas tradicionales como para las técnicas basadas en Redes Neuronales.

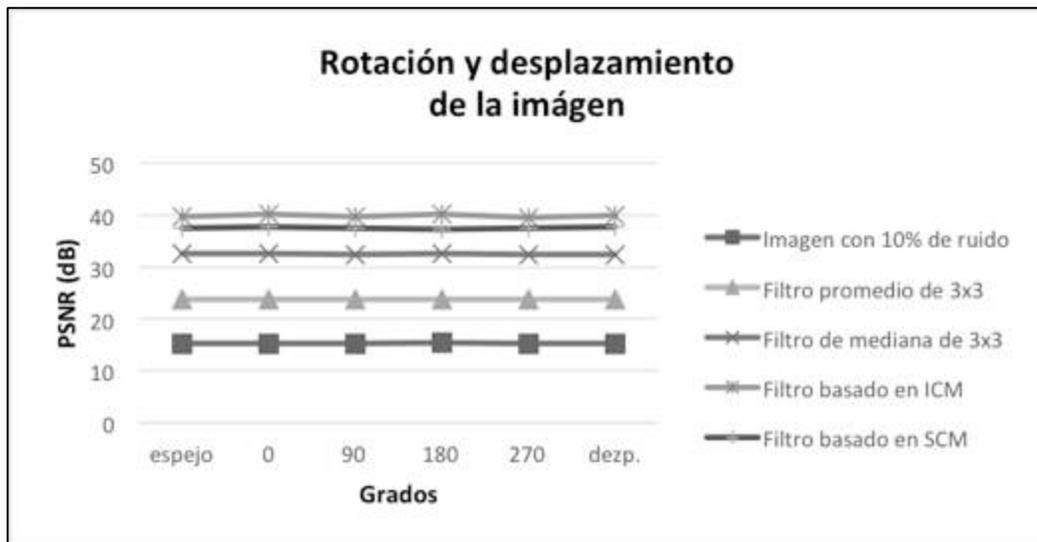


Figura 5.7 Respuesta de los filtros a la rotación de la imagen

5.2.3 Características del filtrado al variar los parámetros de las redes PCNN

Se realizó la variación de los tres parámetros de la red ICM para ajustarlos a una mejor detección del ruido impulsivo, como se muestra en la Tabla 5.2.

Tabla 5.2 Porcentaje de detección del ruido impulsivo de la red ICM

Parámetros			% de detección de ruido sal	% de detección de ruido pimienta	% total detección	Velocidad de detección (número de iteraciones)
f	g	h				
0.9	0.8	20	38.4	100	69	6
0.7	0.8	20	100	100	100	10
0.5	0.8	20	0	100	50	15
0.2	0.8	20	0	100	50	15
0.9	0.9	20	38.4	100	69	10
0.9	0.6	20	69.2	100	85	4
0.9	0.2	20	0	100	50	2
0.9	0.8	40	38.4	100	69	9
0.9	0.8	20	38.4	100	69	10
0.9	0.8	10	38.4	100	69	3
0.9	0.8	5	0	100	50	15

Se determinó experimentalmente que para el modelo ICM el parámetro f que atenúa la señal *feeding* y el parámetro g que atenúa el umbral deben ser cercanos a uno para una mejor detección de pixeles ruidosos, mientras que el parámetro h que amplifica la salida para el umbral debe ser mayor a 20 para una mejor detección reducción del número de iteraciones necesarias.

5.2.4 Comparación con otros detectores de impulsos

Como parte complementaria de trabajo se realizó el filtrado de ruido impulsivo bipolar utilizando el operador de mediana de 3x3 sobre pixeles ruidosos detectados tanto por la técnica con PCNN como por la técnica ERID utilizando la imagen de Lena de 512x512 pixeles, con lo cual se conformó la Tabla 5.3 que indica el PSNR. Se observa que el filtrado con ICM sólo supera a la técnica ERID cuando la probabilidad de ocurrencia del ruido impulsivo es menor de 0.3.

Tabla 5.3 Comparación en PSNR (dB) de filtrado entre detectores de impulsos

Probabilidad de ocurrencia de ruido impulsivo	Con ruido	Filtrado ICM	ERID
	PSNR		
0	∞	46.095	27.386
0.1	30.115	31.111	26.889
0.2	24.844	25.842	24.963
0.3	22.027	23.027	23.277
0.4	20.173	21.172	21.977
0.5	18.833	19.832	20.958

5.2.5 Discusión sobre filtrado de ruido impulsivo

En el caso de [Ma, 2010] se tratan probabilidades de ocurrencia del ruido impulsivo menores a 0.1, por lo que se reprodujeron los algoritmos de filtrado mediante redes PCNN para corroborar los resultados y experimentar otras probabilidades de ruido mas altas.

Los resultados obtenidos del filtrado se compararon con el estado del arte de 2014 cuando fue elaborada la propuesta, no obstante se hizo una nueva revisión documental en la que se encontraron propuestas que alternan entre operadores de mediana, promedio y morfológico [Zhang, 2015a; Zhang, 2015b], filtros adaptativos [Venkatachalam, 2015], o basados en descomposición en sub-bandas [Liu, 2015; Lei, 2015; Gao, 2015], en los que se ha superado a los filtros tradicionales considerados en la tesis; la comparación con dichas técnicas puede tratarse como trabajo futuro.

Mediante la plataforma de experimentación se realizaron pruebas de detección de ruido con imágenes de prueba, se decidió que la red PCNN original no es apta para llevar a cabo el proceso de filtrado ya que sólo detecta el 50% de píxeles ruidosos, por lo tanto sólo se consideraron los modelos ICM y SCM para la comparación con las técnicas tradicionales.

Las Redes Neuronales Pulso-Acopladas se utilizan para detectar los píxeles con ruido impulsivo, el filtrado selectivo de estos píxeles se lleva a cabo de manera complementaria con la información que revela la red [Jiang, 2010; Yuan, 2011; Wen, 2013; Ling, 2013].

En la literatura revisada se encontró un método para determinar el criterio de paro en la iteración de las redes que consiste en calcular el PSNR para cada iteración y determinar la cantidad de iteraciones que maximizara esta cifra [Chenchen, 2008]. Al realizar el experimento para distintas probabilidades de ruido y para diversas imágenes, la repetitividad de los resultados indicó que para la red ICM eran necesarias nueve iteraciones, mientras que para la red SCM bastaban tres manteniendo los parámetros dados.

Por otro lado también se pudo observar que la detección del ruido sal y pimienta se lleva a cabo por separado, ya que en las primeras iteraciones destaca el ruido pimienta y en una segunda serie de activaciones destaca el ruido sal, fenómeno que hasta ahora no ha sido reportado en la literatura, no obstante un análisis numérico de las ecuaciones permitió ver que la detección de los píxeles con ruido pimienta se debe a que todos los píxeles con valor distintos de cero destacan por superar el umbral inicial de cero.

La detección del ruido sal obedece a que el valor de dichos píxeles hace que sean superiores al umbral en la iteración correspondiente, lo que permite suponer que el efecto de las redes puede ser implementado por otros medios de manera equivalente, no obstante comprobar esto ya no fue parte del alcance de la tesis.

También se encontró que puede variar la inclusión de los píxeles cercanos a los márgenes de la imagen en el proceso de detección dependiendo del valor de los parámetros de las redes, lo cual no ha sido comentado en la literatura.

En cuanto a la tarea de filtrado selectiva, se optó por aplicar el operador de mediana, ya que de acuerdo a la literatura la naturaleza del operador permite eliminar los valores extremos en la ventana de la imagen considerada para el filtro, no obstante se ha reportado que también puede ser empleado el operador promedio o morfológico [Ma, 2010], también es importante señalar que los filtros considerados como tradicionales afectan a la imagen aún y cuando ésta no contiene ruido, lo cual no ocurre al emplear la red neuronal con las iteraciones limitadas como se dijo anteriormente.

La determinación del tiempo de procesamiento fue una forma de comparar los métodos basados en Redes Neuronales Pulso-Acopladas, ya que dicho dato no se ha reportado en otros trabajos, así como tampoco las características de la plataforma de pruebas, excepto en [Zhang, 2010].

Se han obtenido tiempos más largos de procesamiento con los algoritmos basados en las Redes Neuronales Pulso-Acopladas, no obstante es posible hacerlos más eficientes por ejemplo al indexar y paralelizar los cálculos para poder trabajar incluso con imágenes de mayor tamaño y a color.

Finalmente se consideró el papel de los parámetros del modelo de Redes Pulso-Acopladas, que en su forma más sencilla (ICM) consta de tres parámetros, experimentalmente se encontró una combinación que produce una detección de todo el ruido presente, no obstante la eliminación del ruido depende también del algoritmo complementario de filtrado por lo que aunque la detección sea buena no se puede garantizar que la imagen se restablezca completamente. El proceso de selección de parámetros óptimos de las Redes Neuronales ha sido afrontada de diversas formas [Peralta, 2010; Almeida, 2008], especialmente en el caso de las Redes tipo Pulso-Acopladas se han encontrado referencias al uso de Algoritmos Genéticos [Shao-Fa, 2010], no obstante el implementarlos excede los alcances de la tesis.

En la Tabla 5.4 se resumen los resultados del filtrado de las técnicas con PCNN y los filtros tradicionales, donde se muestra el rango de ruido en el cual es más alta la capacidad de restauración de la imagen para cada algoritmo, de cuánto es el máximo de capacidad de restauración en PSNR, la habilidad cualitativa de preservar bordes, la invariancia a la escala y rotación de la imagen y el tiempo de procesamiento.

Tabla 5.4 Resumen de características de filtrado de ruido impulsivo con PCNN y filtros tradicionales

Técnica de filtrado	Rango de filtrado efectivo (probabilidad)	Máxima capacidad de restauración de la imagen PSNR (dB)	Capacidad de preservación de detalles finos MAE	Invariancia a la escala	Invariancia a la rotación	Tiempo de procesamiento (ms)
Filtro de mediana de 3x3	0-0.3	20	5	Media	Alta	40
Filtro promedio de 3x3	0.1-0.3	10	20	Alta	Alta	10
Filtro morfológico de 3x3	0-0.3	15	10	No determinado	No determinado	300
ICM	0-0.5	30	3	Baja	Alta	20000
SCM	0-0.5	25	4	Baja	Alta	225

5.3 Filtrado tradicional y basado en PCNN para tratar ruido gaussiano

5.3.1 Comparación entre técnicas de filtrado de ruido gaussiano

Se realizó la comparación entre filtros tradicionales como el filtro de mediana y el filtro *Wiener* y las técnicas basadas en PCNNNI. En el ejemplo con la imagen de *Lena* se observa que la técnica PCNNNI elimina poco ruido preservando algunos detalles, el filtro de mediana suaviza la imagen difuminando los bordes pero disminuye el efecto visual del ruido, finalmente el filtro *Wiener* elimina aún mas el efecto visual del ruido y preserva bien los bordes y líneas (Figura 5.8).

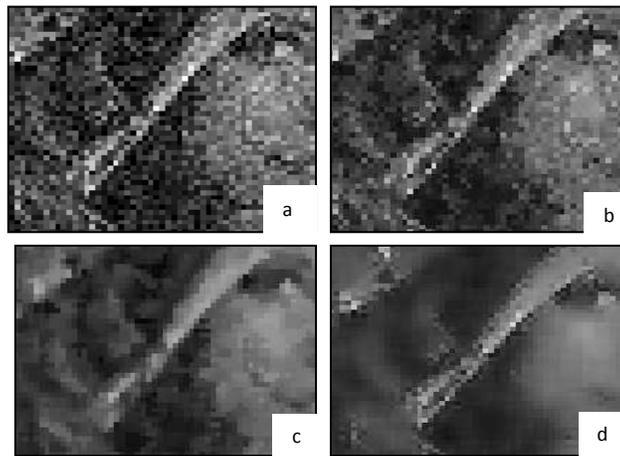


Figura 5.8 Ampliación de a) imagen ruido gaussiano de media= 0 y $\sigma^2 = 0.01$, b) filtrada con PCNNNI, c) con filtro de mediana y d) con filtro Wiener.

Se procesaron distintas imágenes con ruido gaussiano de media 0 aplicando los filtros de mediana, *Wiener* y el filtro basado en Redes Neuronales tipo PCNNNI de lo cual se obtuvieron los resultados que se muestran en la Tabla 5.5, se destaca que el filtro *Wiener* supera a las otras técnicas mejorando la imagen por 7dB, mientras que el filtro de mediana recupera la imagen por 6 dB y la red PCNNNI por 2 dB.

Tabla 5.5 Capacidad de recuperación en NMSE y PSNR de los filtros de ruido gaussiano

Imagen con ruido gaussiano, (σ^2)	Imagen original	PCNNNI		F. Mediana		F. Wiener	
	PSNR	NMSE	PSNR	NMSE	PSNR	NMSE	PSNR
<i>Lena</i> , 0.08	12.15	0.224	14.22	0.080	18.70	0.069	19.31
<i>Lena</i> , 0.09	11.80	0.245	13.83	0.088	18.29	0.075	18.97
<i>Baboon</i> , 0.08	11.93	0.143	13.90	0.064	17.39	0.050	18.47
<i>Baboon</i> , 0.09	11.58	0.155	13.53	0.069	17.05	0.053	18.21
<i>Peppers</i> , 0.08	12.15	0.140	14.27	0.050	18.76	0.044	19.31
<i>Peppers</i> , 0.09	11.76	0.154	13.85	0.056	18.26	0.048	18.91

Para cuantificar los resultados de las técnicas de filtrado se obtuvo la métrica PSNR, se observó su comportamiento conforme aumenta la varianza del ruido gaussiano en la imagen de *Lena* (Figura 5.9). En esta serie de experimentos se incluyeron los algoritmos de filtrado de ruido impulsivo basados en ICM y SCM.

Se observó que para varianzas menores a 0.02 los filtros basados en PCNN tuvieron mejor desempeño que el filtro de mediana hasta por 10 dB, no obstante la técnica PCNNNI no mostró mejoras sustanciales en la imagen filtrada. Por su parte el filtro *Wiener* se perfila como el que produce mejor recuperación de la imagen para varianzas de ruido superiores a 0.02 seguido por el filtro de mediana.

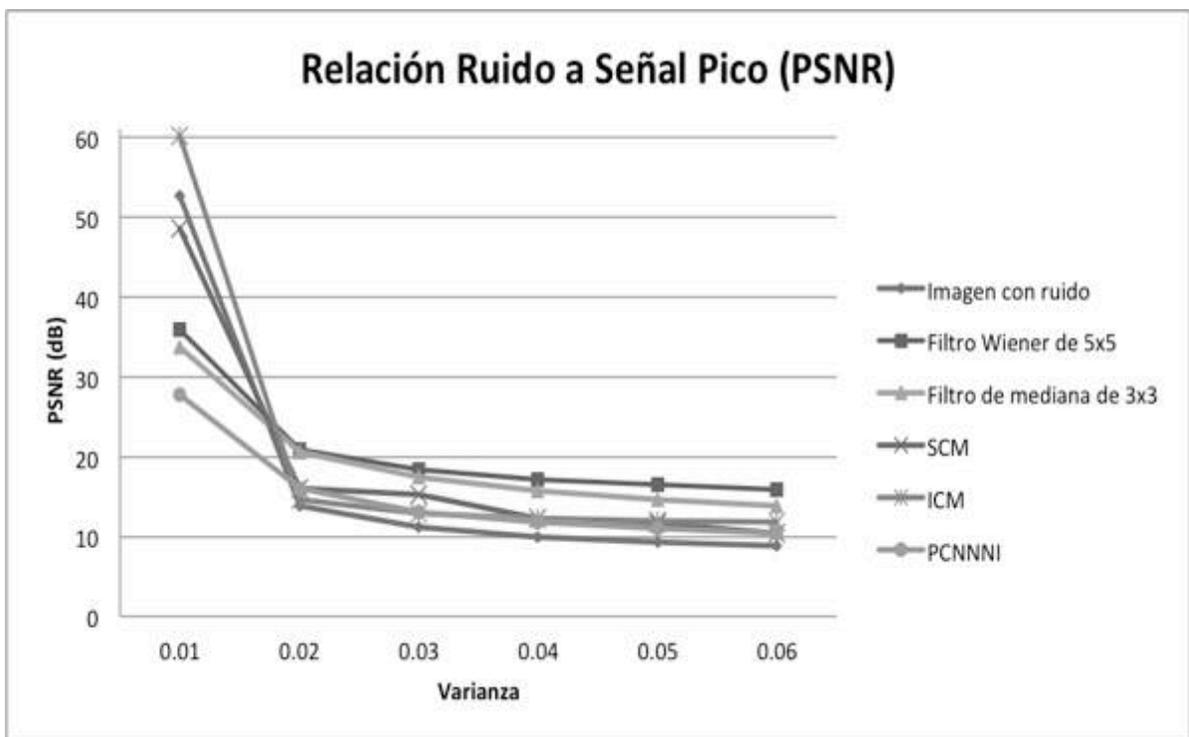


Figura 5.9 Relación entre la varianza y el PSNR para filtros de ruido gaussiano

También se obtuvo el Error Absoluto Medio para *Lena* (Figura 5.10), se observa que la recuperación de la imagen sólo se logra con varianzas de ruido superiores al 0.01, para varianzas menores las técnicas *Wiener*, PCNNNI y el filtro de mediana tienden a empeorar la imagen y para varianzas mayores a 0.01 el filtro *Wiener* mejora la imagen por 50 puntos, el filtro de mediana por 40 puntos y las técnicas de PCNN por 20 puntos.

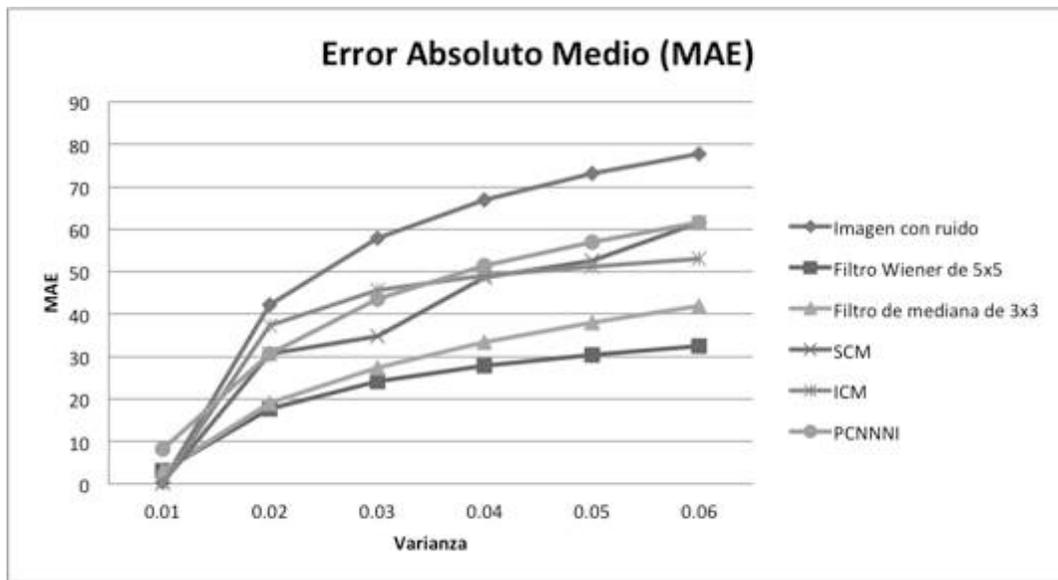


Figura 5.10 Relación entre la varianza y el MAE

Por su lado el cálculo del NMSE (Figura 5.11) muestra que las técnicas basadas en PCNN tienen un efecto similar y en general producen una recuperación de la imagen inferior a la del filtro *Wiener* y el de mediana para varianzas de ruido gaussiano superiores a 0.02.

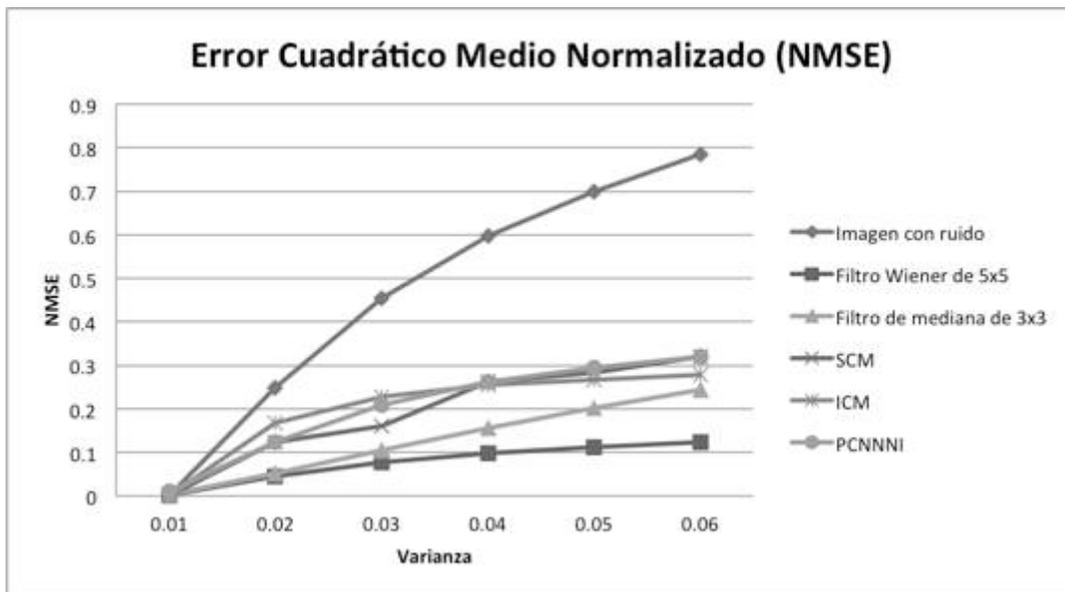


Figura 5.11 Relación entre la varianza y el NMSE

También se midió el tiempo de procesamiento de las técnicas tradicionales como el filtro *Wiener* y el filtro de mediana y de las técnicas basadas en PCNN utilizando la imagen de *Lena*, se destacó que el tiempo de procesamiento de éstas últimas es diez veces mayor que el tiempo de procesamiento de las técnicas tradicionales (Figura 5.12).

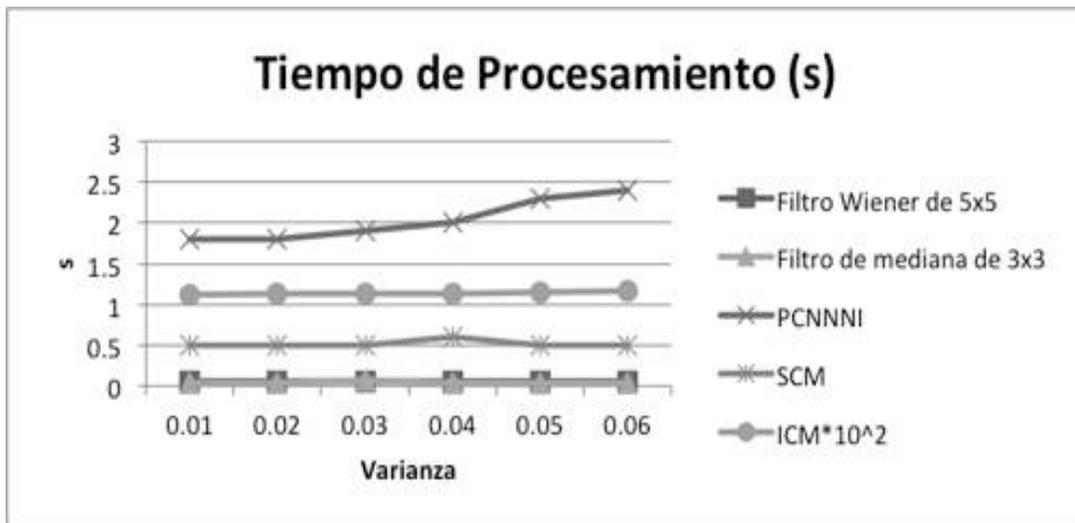


Figura 5.12 Relación entre la varianza del ruido gaussiano y el tiempo de procesamiento

5.3.2 Características del filtrado de ruido gaussiano al variar los parámetros de las PCNN

Se realizó la variación de los parámetros de la red PCNNNI con la imagen de *Lena* de 512x512 píxeles para determinar los mejores valores para el filtrado a través de la medición del PSNR. Dichos parámetros variables son las constantes de ajuste a_F y V_F de la señal *feeding* en donde se observó que el coeficiente de atenuación a_F debía estar en un valor cercano a 0.7, el coeficiente de amplificación V_F debe ser pequeño y menor a uno para obtener un mayor PSNR (Figura 5.13, Figura 5.14).

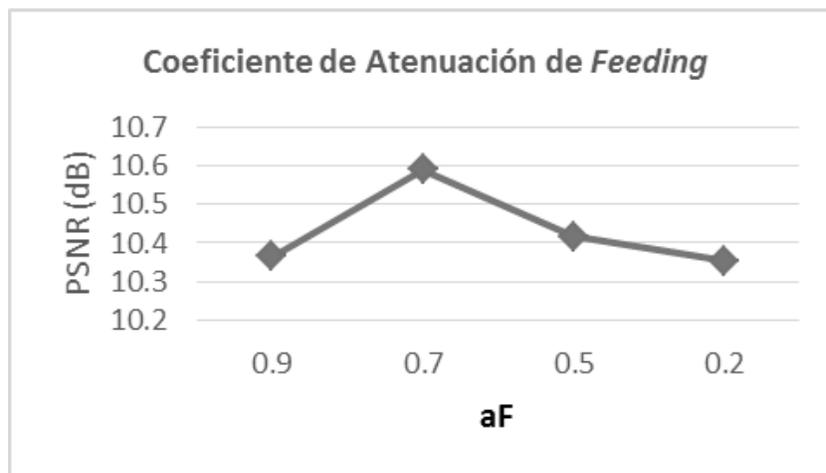


Figura 5.13 Relación entre el coeficiente de atenuación de feeding y el PSNR

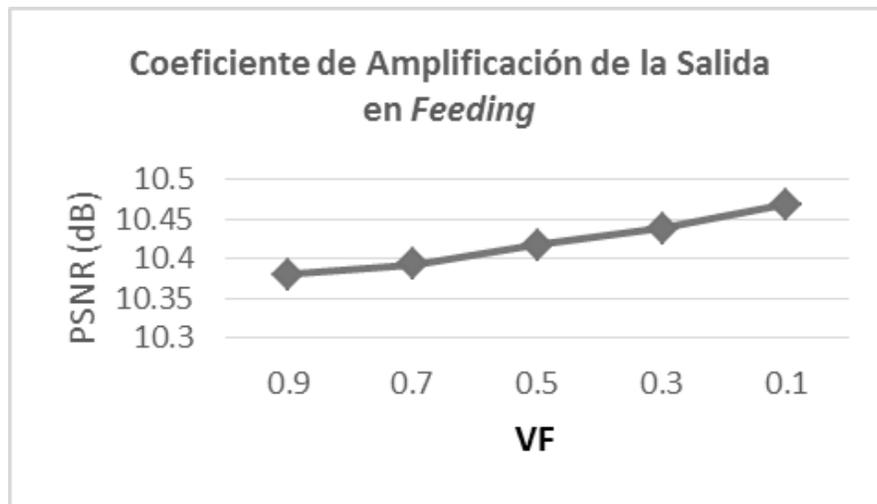


Figura 5.14 Relación entre el coeficiente de amplificación de feeding y el PSNR

Por otra parte al variar las constantes de ajuste de la señal de umbral se observó que el coeficiente de atenuación α_E debía tener un valor cercano a 0.1 para lograr una mayor recuperación en PSNR (Figura 5.15), y por otro lado el coeficiente de amplificación V_E debía ser cercano a seis para obtener un mayor valor de PSNR (Figura 5.16).

Se hace notar que para esta red el cambio de parámetros no produjo mejoras sustantivas al filtrado, pues los cambios son máximo de 0.2 dB.

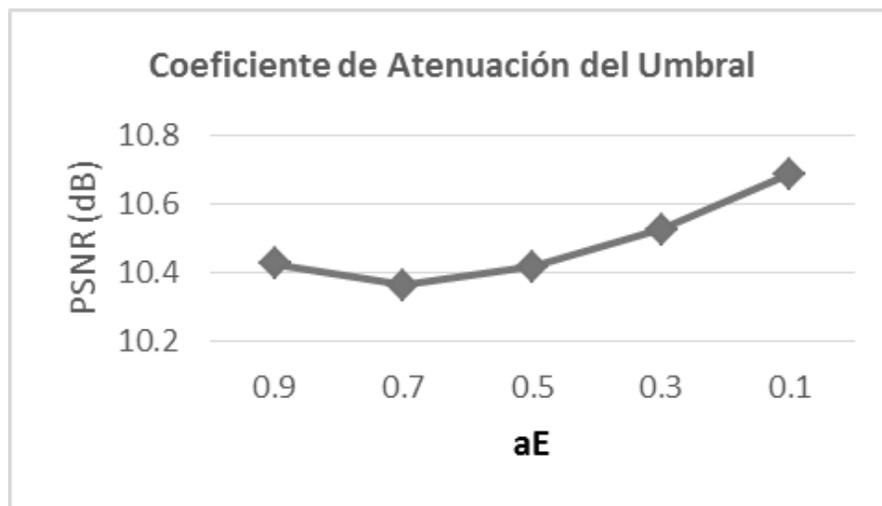


Figura 5.15 Relación entre las constantes de umbral y el PSNR

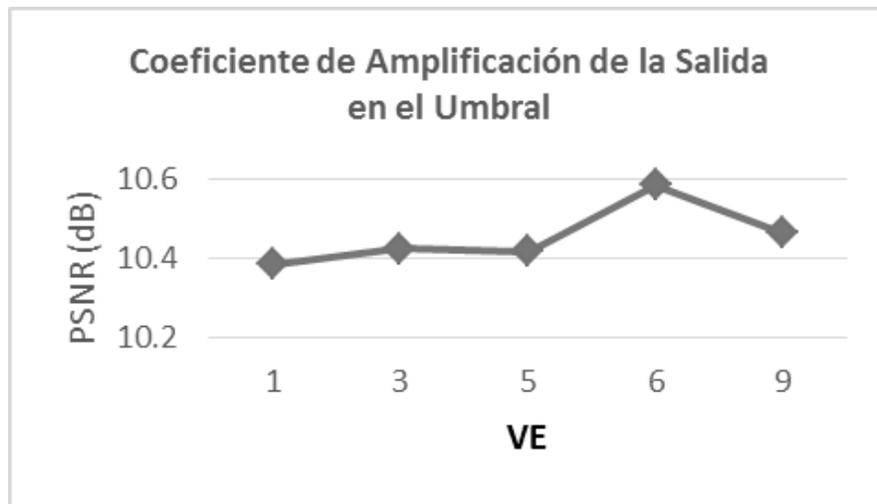


Figura 5.16 Relación entre las constantes de umbral y el PSNR

5.3.3 Discusión sobre filtrado de ruido gaussiano

Para realizar las pruebas de filtrado de ruido gaussiano se optó por el modelo de ruido aditivo de media 0 y varianzas entre 0.01 y 0.3 ya que los trabajos encontrados al respecto tratan estas varianzas o su equivalente en desviación estándar, que es la raíz cuadrada de la varianza normalizada para el manejo de imágenes. De aquí se hace notar que algunas técnicas propuestas en la literatura tratan varianzas menores a 0.01 [Ma, 2010], algunos de estos métodos se reprodujeron para observar su comportamiento con mayores valores de varianza.

También es importante mencionar que cuando no se indica el PSNR o error de las imágenes con ruido sin filtrar no es posible asegurar que con alguna técnica se esté mejorando la imagen pues la puede alterar en menor grado y aparentemente superar a otros filtros [Zhang, 2010; Shao-Fa, 2010].

Como en el caso del tratamiento de ruido impulsivo, se consideraron los trabajos anteriores a la aceptación de la propuesta, no obstante se ha realizado una nueva revisión en busca de técnicas actuales y más eficaces para comparar los filtros basados en PCNN dentro de las cuales se han detectado las siguientes [Zhang, 2016; Palacios, 2016; Liu, 2016; Panetta, 2016; Kumar, 2015; Chang, 2015], que van desde técnicas difusas, transformadas en frecuencias, *Wavelets* y estimación por gradientes y que si bien muestran tener un desempeño eficiente también implican mayor complejidad de cálculo que las PCNN.

Para procesar imágenes con ruido gaussiano es necesario obtener una matriz donde se guarde el número de iteración de la primera activación de las neuronas, esta técnica es frecuentemente utilizada al obtener los datos de una Red Pulso-Acoplada con Interconexión Nula o bien a partir de los modelos SCM e ICM.

Es importante mencionar que la red neuronal no filtra la imagen por sí misma, sino que se utiliza una técnica complementaria que puede ser el filtro *Wiener* [Ma, 2007], los operadores promedio y mediana [Ma, 2010], entre otros.

En el caso de la técnica propuesta en [Ma, 2007] se detectó que al eliminar la interacción entre neuronas se desacoplan los potenciales y la modificación del rango dinámico de la imagen con ruido, para asegurar el llenado de la Matriz de Tiempos en diez iteraciones, afecta negativamente el filtrado.

Los experimentos muestran que para las redes ICM y SCM, la Matriz de Tiempos se llena en menos de 15 iteraciones sin necesidad de cambiar el rango dinámico, aunque se detectó que es necesario ignorar los resultados de la primera iteración en la formación de la matriz, ya que los pixeles con valor distinto de cero se activan en esta iteración.

Mediante el análisis visual de las imágenes filtradas resultantes se hizo evidente que las métricas aplicadas no pueden reflejar la conservación de los detalles finos de la imagen como bordes y líneas, se han revisado otras métricas que detectan los bordes [Zhang, 2010] y la relación entre los valores de los pixeles [Gayathri, 2014] que también pueden ser aplicadas para valorar los resultados, no obstante ello excede el alcance de la tesis.

Al igual que en el caso del ruido impulsivo la red no filtra la imagen, como todos los pixeles tienen ruido en algún grado se busca que los patrones de activación revelen información sobre el nivel de ruido que afecta a los pixeles para aplicar una técnica selectiva de filtrado.

El tiempo de procesamiento se determinó como métrica adicional, las técnicas basadas en PCNN demoran más tiempo que los métodos tradicionales, no obstante también se puede considerar la optimización de los algoritmos y la paralelización de los cálculos para reducir el tiempo.

Se realizaron experimentos sencillos para mejorar el filtrado de la imagen mediante PCNN, se varió cada parámetro por separado para visualizar su efecto en el resultado final y a pesar de que se han considerado técnicas para el cálculo automático de dichos parámetros [Yonekawa, 2009], los experimentos muestran que los cambios no producen mejoras substanciales en el filtrado de ruido gaussiano.

Finalmente se incluye la Tabla 5.6 con los principales resultados que permiten comparar las técnicas tradicionales de filtrado y las técnicas basadas en PCNN para tratar el ruido gaussiano, se incluye el rango de varianza de ruido que filtra mejor cada técnica, el grado en que mejora la imagen (PSNR), su capacidad visual de recuperar los bordes y el tiempo de procesamiento.

Tabla 5.6 Características de filtrado de ruido gaussiano con PCNN y filtros tradicionales

Filtrado de ruido impulsivo	Rango de filtrado efectivo (varianza)	Capacidad de restauración de la imagen PSNR (dB)	Capacidad de preservación de bordes MAE	Tiempo de procesamiento (s)
Filtro Wiener de 5x5	0.02-0.09	10	20	0.06
Filtro de mediana de 3x3	0.02-0.06	9	21	0.03
ICM	0.01-0.02	8	35	112
SCM	0.01-0.03	5	30	0.5
PCNNNI	0.02-0.06	2	30	1.8

5.4 Filtrado de ruido impulsivo y gaussiano mezclados

5.4.1 Simulación de ruido impulsivo y gaussiano mezclados en imágenes digitales

Como pruebas adicionales se realizó la determinación de la habilidad de las redes PCNN como parte de algoritmos de filtrado de ruido mezclado, en donde se consideraron dos casos, en el primero se tiene una imagen de *Lena* de 512x512 píxeles con 10% de ruido impulsivo a la que se agregó ruido gaussiano que va aumentando de varianza de 0.01 a 0.3 (Figura 5.17).

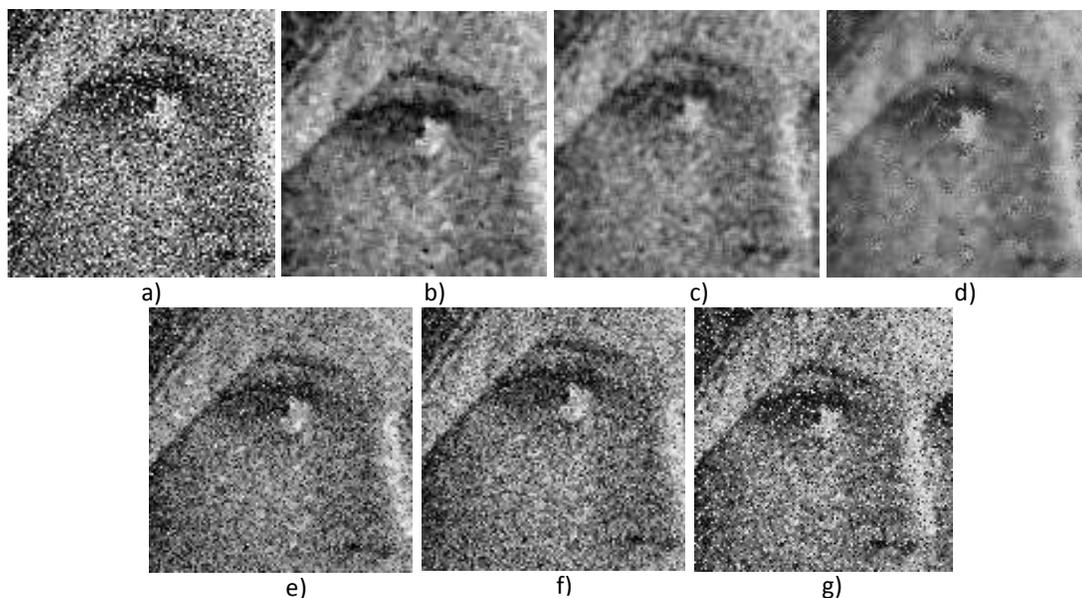


Figura 5.17 a) Ampliación de imagen con 10% ruido impulsivo + ruido gaussiano de $\mu = 0$, $\sigma^2 = 0.01$, filtrada con b) mediana, c) promedio, d) filtro Wiener, e) SCM, f) ICM y g) PCNNNI

En el segundo caso se realizó el experimento de filtrado de una imagen de *Lena* de 512x512 píxeles con ruido gaussiano con $\mu = 0$, $\sigma^2 = 0.01$ a la que se agregó ruido impulsivo de distintas probabilidades de ocurrencia de ruido de 0.1 a 0.9 (Figura 5.18).

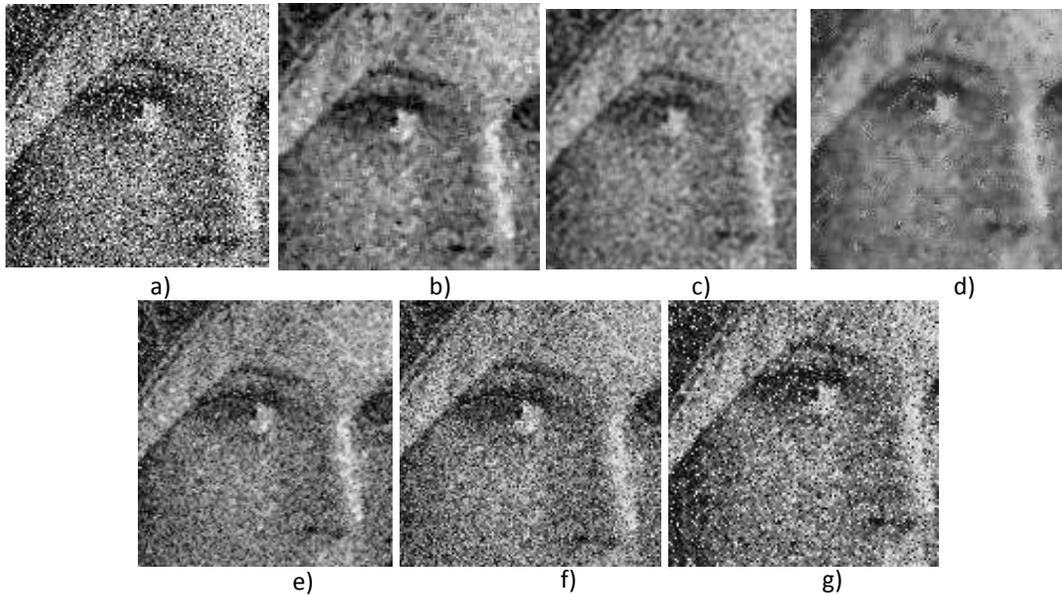


Figura 5.18 a) Ampliación de imagen con ruido gaussiano con $\mu = 0$, $\sigma^2 = 0.01$ + 10% ruido impulsivo, filtrada con b) mediana, c) promedio, d) filtro Wiener, e) SCM, f) ICM y g) PCNNNI.

El comportamiento de los filtros aplicados en el caso de la imagen con ruido impulsivo y con ruido gaussiano variable (Figura 5.19) hace notar que el filtro *Wiener* produce una mayor recuperación de la imagen (10 dB), seguido por el filtro promedio (8dB) y de mediana (7dB), donde al final se ubican los filtros basados en PCNN como son el modelo ICM (3dB), SCM (3dB) y PCNNNI (2dB).

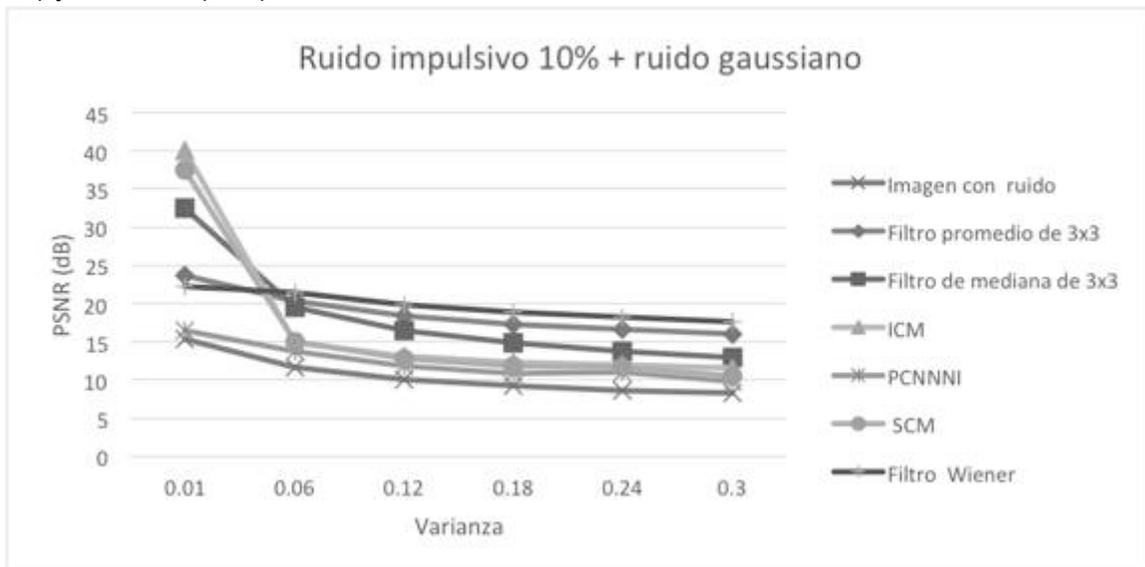


Figura 5.19 Relación PSNR con ruido mezclado cuando aumenta la varianza

Para el caso de la imagen con ruido gaussiano y ruido impulsivo variable se compararon los resultados en PSNR (Figura 5.20), se puede notar que los filtros *Wiener* y promedio, por 10 y 8dB respectivamente, superan a los otros métodos, y se observa que los métodos basados en Redes Neuronales recuperan la imagen con 50% menos efectividad que los filtros tradicionales.

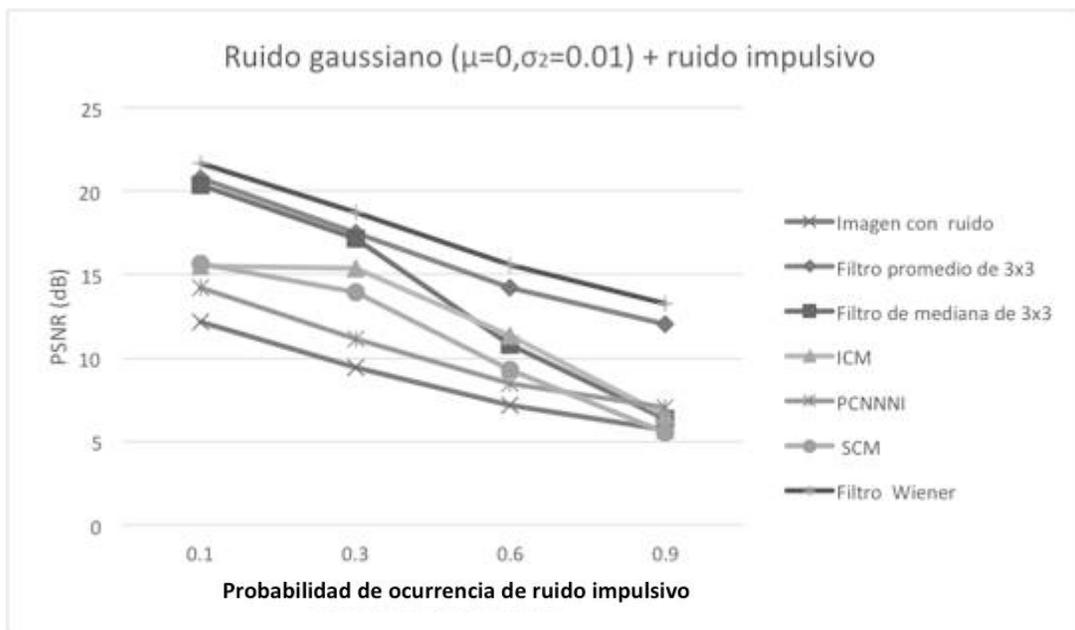


Figura 5.20 Relación PSNR con ruido mezclado cuando aumenta la probabilidad

5.4.2 Discusión sobre filtrado de ruido impulsivo y gaussiano mezclados

En la búsqueda de un algoritmo que permita el filtrado de ruido de cualquier tipo se consideraron los experimentos adicionales para tratar ruido impulsivo y gaussiano mezclados, se han encontrado trabajos que reportan este tipo de filtrado [Shao-Fa, 2010; Yuan, 2011; Tu, 2008; Kai, 2010; Yongqiu, 2008; Ji, 2008; Ma, 2007], no obstante no existe un análisis de la capacidad de los algoritmos basados en PCNN como sistemas dinámicos para realizar la comparación.

Mediante los experimentos realizados se destaca que el filtro promedio puede constituir una herramienta útil para el tratamiento del ruido mezclado, por lo que se le consideró como parte de un algoritmo de filtrado en conjunto con el operador de mediana y se integraron al procesamiento con la red ICM con lo cual se logró superar el rendimiento del filtro *Wiener* como se reportó en el artículo presentado en COMIA 2015.

Capítulo 6 Conclusión

Mediante este capítulo final del escrito se pretende mostrar el logro de los objetivos y alcances mediante actividades y productos específicos, los artículos redactados, las aportaciones y las principales conclusiones del trabajo realizado.

6.1 Objetivos logrados

Se realizaron las actividades de la Tabla 6.1 que dan cumplimiento a los objetivos propuestos en la tesis y la Tabla 6.2 muestra cómo se lograron los alcances.

Tabla 6.1 Actividades realizadas para el logro de los objetivos

OBJETIVOS	ACTIVIDADES
1. Analizar los tipos de ruido en imágenes digitales.	Se realizaron dos programas en lenguaje C para la simulación de los tipos de ruido impulsivo y gaussiano.
2. Implementar algoritmos para simular tipos de ruido específicos en imágenes digitales	Se implementó una plataforma de experimentación para la simulación de ruido impulsivo y gaussiano.
3. Revisar los problemas típicos de las técnicas de filtrado tradicionales.	Se revisaron 32 documentos sobre el tema que fueron generados de 2010 a 2014, de los cuales se seleccionaron ocho para el estado del arte.
4. Implementar algoritmos tradicionales de filtrado.	Se realizaron tres programas en lenguaje C para implementar los filtros de mediana, promedio y morfológico.
5. Revisar los problemas típicos de las RNAs en el dominio de Visión Artificial.	Se detectaron seis problemas típicos, que son: el ajuste de parámetros de la red, el criterio de paro, el procesamiento completo de la imagen, el tiempo y la complejidad de procesamiento, la capacidad de tratar cualquier tipo de imagen y la interpretación de la información interna de la red.
6. Implementar la RNA Pulso-Acoplada y variantes para problemas de filtrado de ruido de imágenes digitales.	Se realizaron tres programas en lenguaje C para el filtrado de imágenes basado en ICM, SCM y PCNNNI.
7. Experimentar con bases de datos de imágenes digitales conocidas en CENIDET y de uso común en la comunidad de Visión Artificial.	Se integró la plataforma de experimentación que admite cualquier imagen en formato bmp. Base de datos de más de 740 imágenes generadas mediante la plataforma de experimentación a partir de cinco imágenes muestra.
8. Obtener los valores de las métricas de restauración de imágenes.	Se agregó a la plataforma de experimentación una sección con la que se pueden generar y guardar archivos con las métricas calculadas en formato xls.
9. Analizar los resultados obtenidos.	Sección 5.2.4, 5.3.3 y 5.4.2 de la presente tesis sobre experimentos, resultados y discusión.
10. Realizar un estudio comparativo.	Sección 6.5 de la presente tesis sobre conclusiones.

Tabla 6.2 Actividades realizadas para el logro de los alcances

ALCANCES	ACTIVIDADES
1. Analizar por lo menos el ruido impulsivo bipolar y el gaussiano en imágenes digitales.	Se analizaron 50 imágenes con ruido impulsivo, 50 imágenes con ruido gaussiano, diez imágenes con ruido impulsivo y gaussiano mezclados y diez con cambios de escala y rotación.
2. Implementar algoritmos para la simulación de ruido impulsivo y gaussiano en imágenes digitales.	Se realizaron dos programas en lenguaje C para la simulación de ruido impulsivo y gaussiano, el ruido gaussiano utilizado para las pruebas se generó en Matlab dado que no fue igualado.
3. Analizar por lo menos tres técnicas tradicionales para el filtrado de imágenes.	Se analizaron 50 imágenes con ruido impulsivo y 50 con ruido gaussiano filtradas con filtro de mediana, 50 imágenes con ruido impulsivo y diez con ruido gaussiano filtradas con el filtro promedio, 20 imágenes con ruido impulsivo filtradas con filtro morfológico y 50 imágenes con ruido gaussiano filtradas con filtro <i>Wiener</i> .
4. Implementar por lo menos tres algoritmos tradicionales para el filtrado de imágenes.	Se realizaron tres programas en lenguaje C para implementar los filtros de mediana, promedio y morfológico, el filtro <i>Wiener</i> utilizado para las pruebas se generó en Matlab dado que no fue igualado.
5. Elegir e implementar por lo menos tres variantes de RNA Pulso-Acoplada para problemas de filtrado de ruido de imágenes digitales.	Se generaron 20 imágenes con ruido impulsivo y diez con gaussiano filtradas con ICM, 20 imágenes con ruido impulsivo y diez con gaussiano filtradas con SCM y 20 imágenes con ruido gaussiano filtradas con PCNNNI.
6. Experimentar con al menos cinco bases de datos de imágenes digitales.	Se utilizó la plataforma de experimentación para tratar imágenes (bmp) y guardar las imágenes generadas y los datos en xls. Sólo se emplearon <i>Lena, Baboon, Peppers, Grass y Cameraman</i> .
7. Aplicar por lo menos dos métricas para evaluar la restauración de las imágenes.	Se programó el cálculo automático de las métricas PSNR, MAE, NMSE y tiempo de procesamiento.
8. Realizar el estudio comparativo de la calidad de la imagen y tiempo de procesamiento.	Se obtuvieron los datos y las gráficas de PSNR, MAE, NMSE y tiempo de procesamiento de las secciones 5.2, 5.3 y 5.5 del presente documento.
9. Las comparaciones se realizaron con respecto a lo reportado en la literatura y utilizando software desarrollado de uso libre.	Recopilación de artículos y documentos del estado del arte. Elaboración de la plataforma de experimentación en lenguaje C.
10. No se busca mejorar resultados sino detectar claramente las ventajas y desventajas de la implementación de algoritmos de filtrado con PCNN.	Se mejoró el tiempo de procesamiento de imágenes con ruido impulsivo mediante ICM en 50% superando al filtro de mediana. Se mejoró la capacidad de filtrado de ruido gaussiano del algoritmo PCNNNI en 6dB superando al filtro <i>Wiener</i> . Se publicaron tres artículos escritos para ICMEAE 2015, COMIA 2016, MCP2016 y para la revista CyS.

6.2 Productos

6.2.1 Plataforma de experimentación

Se desarrolló una plataforma de experimentación con interfaz, en lenguaje C y se redactó un manual de instalación de los componentes necesarios para ejecutarla.

6.2.2 Bases de datos de imágenes con ruido y filtradas

Se usaron cinco imágenes originales a color en formato bmp y de ahí se generaron las variantes con ruido y filtradas (Tabla 6.3).

Tabla 6.3 Descripción de la base de datos generada

	<i>Lena</i>						<i>Baboon</i>		<i>Camerman</i>	<i>Peppers</i>		<i>Grass</i>
	Ruido impulsivo	Ruido gaussiano	Ruido mezclado	Procesadas	Cambio de parámetros	Variación de imagen	Ruido impulsivo	Ruido gaussiano	Ruido impulsivo	Ruido impulsivo	Ruido gaussiano	Ruido impulsivo
Número de imágenes	10	50				12	10	2	10	10	2	2
Filtro Mediana de 3x3	10	50	10			12	10	2	10	10	2	2
Filtro Promedio de 3x3	10	10	10			12	10	2	10	10	2	2
Filtro Morfológico de 3x3	10						2		2	2		2
Filtro Wiener		50	10					2			2	
Red ICM	80	10	10	10	10	12	2	2	2	2	2	2
Red SCM	80	10	10	40		12	2	2	2	2	2	2
Red PCNNNI		30	10	10	20			2			2	
Total	200	210	60	60	30	60	36	14	36	36	14	12

6.2.3 Bases de datos de imágenes de prueba

Se recolectaron imágenes de prueba estándar de 5 bases de datos distintas, con imágenes en blanco y negro, en escala de grises y a color, de varios tamaños desde 32 hasta 2048 píxeles por lado y en formato tiff, bmp, jpeg y ljpeg.

6.2.4 Artículos elaborados

A. Se realizó una publicación para el evento *International Conference on Mechatronics, Electronics and Automotive Engineering* (ICMEAE) de la IEEE celebrado en noviembre de 2015 en Cuernavaca, Morelos, donde constan los resultados obtenidos de la implementación de ICM combinada con el filtro de mediana para mejorar el filtrado del ruido impulsivo mediante un algoritmo basado en Redes Neuronales Pulso-Acopladas, dicho artículo está registrado como ISBN 978-1-4673-8329-5/15 DOI 10.1109/ICMEAE.2015.21, pp.41.44 y cuyo fragmento se muestra en la Figura 6.2.

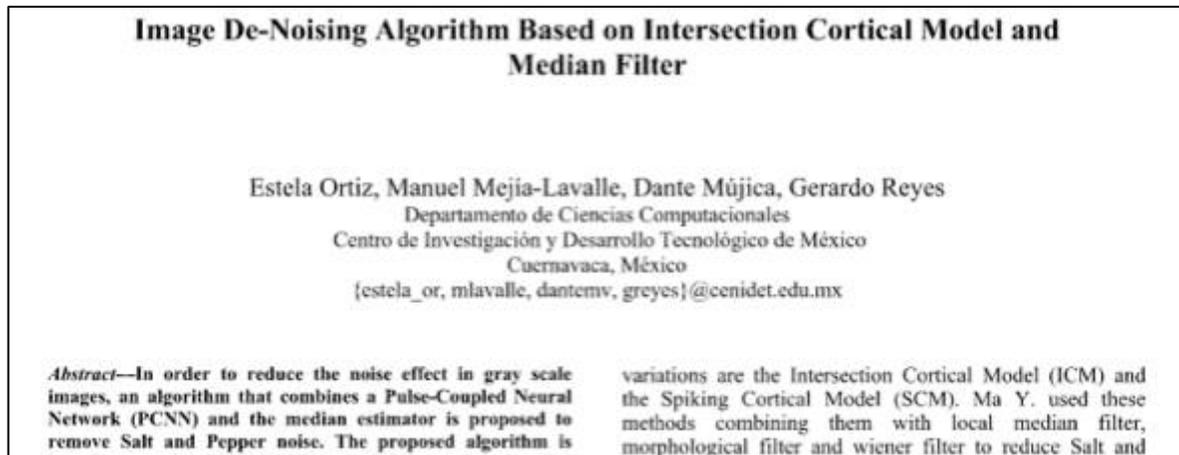


Figura 6.1 Publicación realizada para ICMEAE 2015

B. Se presentó un segundo artículo en inglés sobre el filtrado de ruido impulsivo mediante Redes Neuronales Pulso-Acopladas para el octavo evento del *Mexican Conference on Pattern Recognition* (MCPR2016) del INAOE presentado en junio de 2016 en la ciudad de Guanajuato, el cual fue aprobado para su publicación en *The MCPR 2016, LNCS 9703 proceedings*. Un fragmento del artículo se muestra en la Figura 6.3.

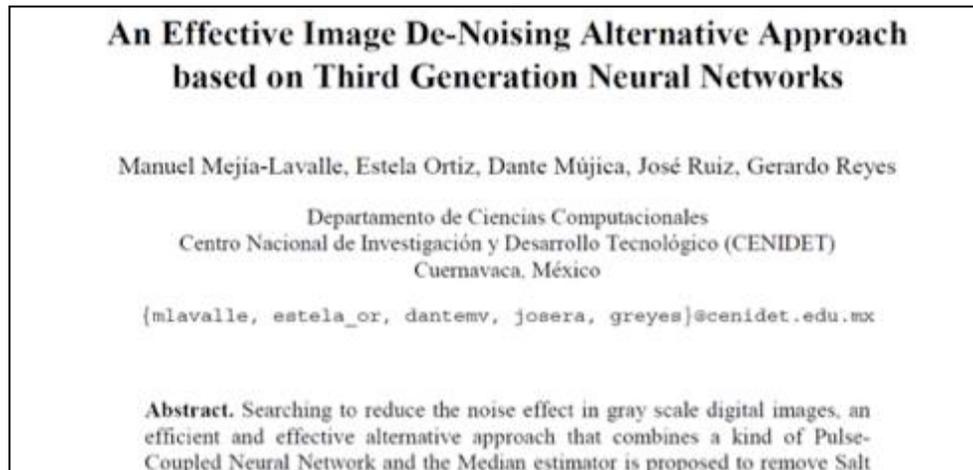


Figura 6.2 Publicación realizada para MCPR2016

C. Se escribió un trabajo en español sobre el mejoramiento del filtrado de ruido gaussiano mediante Redes Neuronales tipo Pulso-Acopladas y fue aprobado por el octavo Congreso Mexicano de Inteligencia Artificial (COMIA), celebrado en la ciudad de Puebla del 23 al 28 de mayo de 2016 organizado por el SMIA y el INAOE. La primera página del escrito se muestra en la Figura 6.4, cabe mencionar que el trabajo fue realizado como parte de la estancia de investigación que se llevó a cabo en el CIC del IPN.

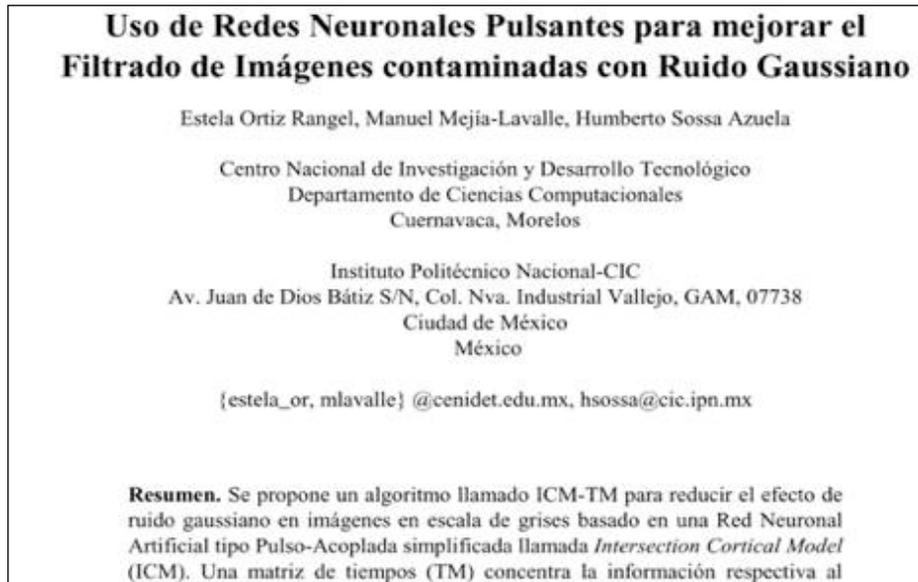


Figura 6.3 Publicación realizada para COMIA 2016

D. Finalmente se escribió un trabajo sobre el filtrado de ruido gaussiano mediante Redes Neuronales tipo Pulso-Acopladas para la revista *Computación y Sistemas* del CIC-IPN ISSN: 2007-9737, elaborado con la colaboración de Doctor Humberto Sossa (Figura 6.5). Dicho artículo se encuentra en revisión para ser aprobado para su publicación.

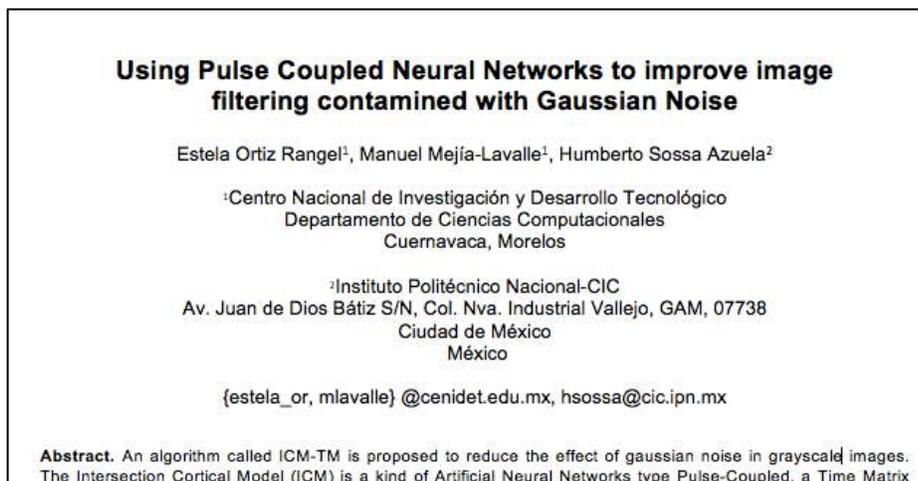


Figura 6.4 Publicación realizada para la revista *CyS*

6.3 Aportaciones

1. La determinación de criterio de paro para las redes PCNN al tratar con ambos tipos de ruido:
 - a) Para ruido impulsivo, basado en la maximización del PSNR.
 - b) Para ruido gaussiano, basado en el llenado total de la matriz de tiempos.
2. Determinación de parámetros de las PCNN en los algoritmos de filtrado de imágenes por medio de la plataforma de pruebas que mejoren la calidad del filtrado.
 - a) Para la red ICM en el filtrado de ruido impulsivo, se propone que los parámetros queden: $f = 0.7$, $g = 0.8$, $h = 20$ para detectar 100% de los píxeles ruidosos.
 - b) Para la red PCNNNI en el filtrado de ruido gaussiano, se propone que los parámetros sean $a_F = 0.7$, $V_F = 0.1$, $a_E = 0.1$, $V_E = 6$ por separado.
3. Se propuso un algoritmo al que se denominó ICMD-M (incluido en el artículo para ICMEAE) para eliminar el ruido impulsivo por medio de la selección de píxeles ruidosos con el modelo ICM y el operador de mediana, el cual tiene un desempeño superior al filtro de mediana por al menos 7 dB, y consta de dos redes paralelas que procesan la imagen y su inversa.
4. Se propuso un algoritmo al que se denominó ICM-TM (incluido en el artículo para COMIA) para eliminar el ruido gaussiano por medio de la obtención de una matriz de tiempos a partir del modelo ICM y la aplicación de un criterio selectivo de filtrado que combina la información de la matriz con los operadores promedio y mediana el cual tiene un desempeño superior al filtro *Wiener* por 2dB, al filtro de mediana por 2.2 dB y a la red PCNNNI por 7 dB.

6.4 Lecciones aprendidas

La realización de la tesis permite el desarrollo de diversas capacidades como el análisis, la abstracción, el planteamiento de objetivos, la identificación de problemas y la organización del tiempo. Un aspecto importante a tomar en cuenta es la selección del estado del arte y la necesidad de seguir actualizando la información en busca de lograr una mayor flexibilidad de la investigación, la posible adaptación de los objetivos planteados y una mayor utilidad de los hallazgos encontrados.

La realización de la estancia es una de las experiencias más enriquecedoras al momento de investigar, ya que brinda diversos aprendizajes como la importancia de preparar las actividades que se van a realizar, la comunicación con el asesor y la oportunidad de aprovechar cursos que se ofrecen en el centro al que se acuda.

6.5 Conclusiones

Al final de este trabajo de maestría se llegó a identificar claramente las ventajas y desventajas de las diversas técnicas de filtrado consideradas (Tabla 6.4, 6.5).

Tabla 6.4 Ventajas y desventajas del filtrado de ruido impulsivo y gaussiano mediante filtros tradicionales

FILTROS TRADICIONALES		
	VENTAJAS	DESVENTAJAS
Ruido impulsivo	<p>El tiempo de procesamiento del filtro de mediana y promedio es corto comparado con los basados en PCNN en un orden de 1:5 en el mejor de los casos.</p> <p>El filtrado es invariante a la rotación y escalamiento de la imagen.</p>	<p>Los filtros de mediana, promedio y morfológico afectan a la imagen aun cuando no hay ruido presente por lo menos en 15 dB más, cinco puntos de MAE. El filtrado es sensible al escalado de la imagen.</p> <p>El filtro de mediana y promedio afecta líneas y bordes indistintamente.</p> <p>Ni el filtro promedio ni el morfológico eliminan el ruido impulsivo de probabilidad mayor a 0.3 de manera efectiva.</p> <p>El filtro morfológico fue descartado para el filtrado de ruido impulsivo porque su tiempo de procesamiento es largo (300ms) con respecto al filtro de mediana (40 ms) sin producir una mejor recuperación de la imagen.</p>
Ruido gaussiano	<p>Las imágenes con ruido gaussiano restauradas con el filtro <i>Wiener</i> tienen mayor calidad (6dB) que con PCNN.</p> <p>El tiempo de procesamiento es de 0.06 segundos (10 veces menor) comparado con PCNNNI).</p> <p>El filtro <i>Wiener</i> es capaz de filtrar una imagen con ruido impulsivo y gaussiano mezclados en 10dB, de modo que es superior a los filtros de mediana, promedio, ICM, SCM y PCNNNI.</p>	<p>Para varianzas inferiores a 0.01 los filtros <i>Wiener</i> y de mediana deterioran la imagen en 18dB.</p> <p>El filtro <i>Wiener</i> depende de la estimación de una función de degradación y de la imagen no contaminada, así como de la estimación de las características estadísticas del ruido y representa una gran carga computacional debido al cálculo de la transformada en frecuencia.</p>

Tabla 6.5 Ventajas y desventajas del filtrado de ruido impulsivo y gaussiano mediante PCNN

FILTROS BASADOS EN PCNN		
	VENTAJAS	DESVENTAJAS
Ruido impulsivo	<p>El filtro basado en SCM detecta el ruido impulsivo en 4 iteraciones de la red (200ms).</p> <p>El tiempo de procesamiento con ICM es susceptible de mejora al indexar datos. Los filtros con ICM y SCM permiten el filtrado selectivo de ruido impulsivo sin afectar líneas y bordes (3-4dB mejor que el filtro de mediana).</p> <p>La red ICM permite restaurar la imagen mejor que la red SCM por al menos 2dB.</p> <p>La red ICM y la red SCM detectan más del 80% del ruido impulsivo.</p> <p>El filtrado es invariante a la rotación y escalamiento de la imagen</p> <p>Es posible determinar experimentalmente los valores de los parámetros de la red ICM para detectar el 100% de los pixeles ruidosos ($f=0.7$, $g=0.8$, $h=20$).</p>	<p>El filtro basado en ICM requiere por lo menos nueve iteraciones de la red para detectar el ruido impulsivo (20 mil ms).</p> <p>La red PCNN original fue descartada para el filtrado de ruido impulsivo ya que sólo detecta el 50% del ruido.</p> <p>El filtrado es sensible al escalado de la imagen.</p>
Ruido gaussiano	<p>Los filtros con ICM y SCM recuperan mejor la imagen para varianzas de ruido gaussiano menores a 0.02.</p> <p>Es posible proponer un algoritmo de filtrado de ruido gaussiano mediante la red ICM que filtre ruido impulsivo y gaussiano mezclados, sin necesidad de modificar la red, sólo el filtro complementario.</p>	<p>El filtrado de ruido gaussiano con PCNN aún no supera el desempeño del filtro <i>Wiener</i> de 5x5.</p> <p>Los algoritmos basados en ICM y SCM superan al de PCNNNI en el filtrado de ruido gaussiano por al menos 2dB.</p> <p>La variación de parámetros de la red PCNNNI no produce mejoras sustantivas al filtrado (0.2dB).</p>

6.6 Trabajo futuro

El trabajo futuro de la investigación puede centrarse en puntos como el ajuste automático de los parámetros de operación de las redes PCNN, SCM y PCNNNI para filtrar ruido impulsivo y gaussiano, por ejemplo, por medio de algoritmos evolutivos.

Llevar a cabo la comparación de los filtros basados en PCNN optimizados con técnicas actuales de filtrado mencionadas en la discusión de ruido impulsivo y ruido gaussiano, ya sea mediante la re-implementación de las técnicas o bien mediante la comparación de las cifras.

Experimentar con imágenes a color y de distintos tamaños por medio de la plataforma implementada en lenguaje C para verificar su respuesta de filtrado.

Llevar a cabo un análisis de las redes PCNN como sistemas dinámicos para la tarea de filtrado.

Implementar en la plataforma de experimentación otras técnicas de simulación de ruido, de filtrado y de procesamiento de imágenes de mayor tamaño.

Realizar el filtrado de ruido *Speckle* por medio de los algoritmos de filtrado propuestos y proponer uno nuevo que pueda filtrar cualquier tipo de ruido sin información anterior.

Disminuir el tiempo de procesamiento de los algoritmos con PCNN mediante la optimización del código y la paralelización del cálculo.

Análizar y mejorar el desempeño de las PCNN como detectores de impulsos y de ruido impulsivo no bipolar que permita su comparación con otras técnicas de detección.

Seleccionar otra técnica complementaria de filtrado alterna a los operadores de mediana, promedio y morfológico.

Bibliografía

- [Aizenberg, 2003]** Aizenberg I., Astola, T. Detectors of the Impulsive Noise and new Effective Filters for the Impulsive Noise Reduction, Image Processing, Algorithms and Systems II, SPIE VOL. 5014, 419-428, 2003.
- [Almeida, 2008]** Almeida, L.M., Ludermir, T.B., An Evolutionary Approach for Tuning Artificial Neural Network Parameters. , pp.156–163. 2008
- [Barbu, 2012]** Barbu T. Novel linear image denoising approach based on a modified gaussian filter kernel. Numer Funct Anal Optim. 2012
- [Bovik, 2009]** Bovik, A. The Essential Guide to Image Processing. Image. Elsevier Inc. (2009).
- [Cárdenas, 2015]** Cárdenas, S. Implementación y Evaluación de Redes Neuronales Artificiales tipo “Pulse-Coupled Neural Networks” (PCNN) Aplicadas a Visión Artificial. Tesis de Maestría en Ciencias de la Computación, CENIDET. 2013
- [Chang, 2015]** Chang, J.R., Wang, Y.F., Propagated Image Filtering. IEEE, 978-1-4673(2), pp.10–18. 2015
- [Chenchen, 2008]** Chenchen, L., Zhang, Z., Sonar Images De-noising Based on Pulse Coupled Neural Networks. Congress on Image and Signal Processing, pp.403–406. 2008
- [Gao, 2015]** Gao, G., Liu, Y., An efficient three-stage approach for removing salt & pepper noise from digital images. Optik - International Journal for Light and Electron Optics, 126(4), pp.467–471. 2015
- [Gayathri, 2014]** Gayathri, A., Srinivasan, A., An Efficient Algorithm for Image Denoising using NLM and DBUTM Estimation. IEEE, 978-1-4799. 2014
- [González, 2002]** González R.C.: Digital Image Processing, 2nd Ed. Prentice Hall (2002)
- [ImageProcessing Place, 2016]** Imageprocessingplace.com. (2016). Image Databases. Available at: http://www.imageprocessingplace.com/root_files_V3/image_databases.htm [Accessed 11 Oct. 2016].
- [Ji, 2008]** Ji, L., Yi, Z., A mixed noise image filtering method using weighted-linking PCNNs. Neurocomputing, 71, pp.2986–3000. 2008
- [Jiang, 2010]** Jiang J, Shen J. An Effective Adaptive Median Filter Algorithm for Removing Salt & Pepper Noise in Images. Symposium on Photonics and Optoelectronic, SOPO 2010 - Proceedings. IEEE. 2010.
- [Kai, 2010]** Kai, H.E., Cheng, W., Shao-Fa, L., Modified PCNN Model and Its Application to Mixed-noise Removal. International Conference on Innovative Computing and Communication and 2010 Asia-Pacific Conference on Information Technology and

- Ocean Engineering, 213(2), pp.213–216. 2010
- [Kang, 2013]** Kang D, Lim H. Efficient noise reduction in images using directional modified sigma filter. Springer Sci Media New York. 2013
- [Kumar, 2015]** Kumar A. V., Barjinder Singh Saini. Alexander fractional integral filtering of wavelet coefficients for Image Denoising. Signal & Image Processing: An International Journal (SIPIJ), 6(3), pp.43–54. 2015
- [Lei, 2015]** Lei, Y. Technique for image de-noising based on non-subsampled shearlet transform and improved intuitionistic fuzzy entropy. Optik - International Journal for Light and Electron Optics, 126(4), pp.446–453. 2015.
- [Lindblad, 2005]** Lindblad T., Kinser J.M.: Image processing using pulse-coupled neural networks. In: Springer, 2nd. Ed. (2005)
- [Ling, 2013]** Ling C. Adaptive image de-noising algorithm in intersecting cortical model. Journal of Multimedia. 2013
- [Liu, 2015]** Liu, L. et al. A new weighted mean filter with a two-phase detector for removing impulse noise. Information Sciences, 315, pp.1–16. 2015
- [Liu, 2016]** Liu, J. et al. Image denoising with multidirectional shrinkage in directionlet domain. Signal Processing, 125, pp.64–78. 2016
- [Ma, 2010]** Y. Ma, K. Zhan, and Z. Wang, Applications of Pulse-Coupled Neural Networks. Springer, 2010, pp. 11–25.
- [Ma, 2007]** Ma, Y. et al. A novel algorithm of image gaussian noise filtering based on pcnn time matrix. IEEE International Conference on Signal Processing and Communications (ICSPC), pp.1499–1502. 2007
- [Matworks, 2014]** <http://www.mathworks.com/products/matlab/examples.html> (25-03-14)
- [Myler, 1993]** Myler, Hr Harley R, Weeks, Arthur R, The pocket handbook of image processing algorithms in C. Prentice Hall, 1993.
- [Pajares, 2008]** Pajares, G. *Visión por computador-Imágenes digitales y aplicaciones*. 2da Edición. Ed. Alfaomega, Ra-Ma, 2008
- [Palacios, 2016]** Alfredo E. Enriquez, Ponomaryov, V., Image Denoising using Block Matching and Discrete Cosine Transform with Edge Restoring. IEEE CONFERENCE PROCEEDINGS. CONIELECOMP, 978-1-5090, pp.140–147. 2016
- [Panetta, 2016]** Panetta, K., Bao, L., Agaian, S. Sequence - to - Sequence Similarity - Based Filter for Image Denoising. IEEE Sensors Journal. 2016
- [Peralta, 2010]** Peralta, J., Gutierrez, G., Sanchis, A. Automatic Design of Artificial Neural Networks to Forecast Time Series. III Simposio de Inteligencia Computacional,

pp.238–242. 2010

- [Pratt, 2001]** Pratt, W. Generalized Wiener Filtering Computation Techniques. IEEE Transactions on Computers. C-21, 636-641(1972)
- [Ramírez, 2012]** Ramírez Quintana J. A., Chacon Murguia M. I. and Chacon Hinojos J. F., “Artificial Neural Image Processing Applications: A Survey”, Engineering Letters, 20:1, pp. 68-80, 2012.
- [Sabia.tic.udc.es, 2016]** Sabia.tic.udc.es. (2016). Visión artificial e interacción sin mandos. Available at:<http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/3D/VisionArtificial/iteracion.html> [Accessed 11 Oct. 2016].
- [Schulte, 2006]** S. Schulte, M. Nachtgeael. A fuzzy impulse noise detection and reduction method. IEEE Transactions on Image Processing. 2006, pp. 1153-1162
- [Shao-Fa, 2010]** Shao-Fa L, Kai HE, Cheng W,. Modified PCNN Model and Its Application to Mixed-noise Removal. Int Conf Innov Comput Commun 2010 Asia-Pacific Conf Inf Technol Ocean Eng. 2010;213(2):213-216. doi:10.1109/CICC-ITOE.2010.61.
- [Shao-Fa, 2008]** Shaofa, Y. Tu, and M. Wang, Mixed-noise Removal for Color Images Using Modified PCNN Model, Second Int. Symp. Intell. Inf. Technol. Appl., pp. 347–351, 2008
- [Shao-Fa, 2007]** Tu Yongqiu, Li Shaofa, Wang Minqin (2007). *Modified PCNN Model and Its Application to Mixed-noise Removal*. Proceeding of ROBIO, pp. 1341-1347.
- [Sreenivasulu, 2014]** Sreenivasulu P, Chaitanya NK. Removal of Salt and Pepper Noise for Various Images Using Median Filters : A Comparative Study. IUP J Telecommun.;VI(2):54-70. 2014
- [Tang, 2007]** Tang, H., Tan, K. C., & Yi, Z. . *Neural Networks: Computational Models and Applications*. Springer-Verlag. 2007.
- [Tu, 2008]** Tu, Y., Li, S., Wang, M. Mixed-noise Removal for Color Images Using Modified PCNN Model. Second International Symposium on Intelligent Information Technology Application, pp.347–351. 2008
- [USC-SIPI,1977]** <http://sipi.usc.edu/database> (11-05-16)
- [Varela, 2006]** Varela J. L. Filtro de orden estadístico para supresión de ruido en imágenes digitales. 2006, pp. 30-31
- [Venkatachalam, 2015]** Venkatachalam, K., Mohan, E. A Novel Algorithm for Image Denoising using Modified Adaptive Median Filter. Research Journal of Applied Sciences, Engineering and Technology, 10(4), pp.373–375. 2015
- [Wang, 2010]** Z. Wang, Y. Ma, F. Cheng, L. Yang “Review of pulse-coupled neural networks”, Image and Vision Computing Journal 28, Elsevier, pp. 5-13, 2010.
- [Wen, 2013]** Wen, H., Wen, J. Image Denoising and Restoration Using Pulse Coupled Neural Networks. 6th International Congress on Image and Signal Processing, (Cisp),

pp.282–287. 2013

- [Yonekawa, 2009]** Yonekawa, M. Kurokawa, H. An automatic parameter adjustment method of pulse coupled neural network for image segmentation. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 834-843. 2009
- [Yongqiu, 2008]** Yongqiu, T., Shaofa, L., Wang, M., Modified PCNN Model and Its Application to Mixed-noise Removal. International Conference on Robotics and Biomimetics, pp.1333–1338. 2008
- [Yuan, 2011]** Yuan-yuan, C. et al. A new method of denoising mixed noise using Limited Grayscale Pulsed Couple Neural Network. Cross Strait Quad-Regional Radio Science and Wireless Technology Conference, pp.1410–1413. 2011.
- [Zárate, 2015]** Zárate, M. Extracción de Características de Imágenes Digitales mediante una Red Neuronal Artificial Pulsante. Tesis de Maestría en Ciencias de la Computación, CENIDET. 2014
- [Zhang, 2007]** H. Zhang, Z. Zhang, Y. Ma, A novel image de-noising algorithm combined PCNN with morphology, Processing and Communication Systems, 2007.
- [Zhang, 2010]** D. Zhang, S. Mabu, and K. Hirasawa, Noise Reduction using Genetic Algorithm Based PCNN Method, IEEE, pp. 2627–2633, 2010.
- [Zhang, 2015a]** Zhang, C., Wang, K. A switching median–mean filter for removal of high-density impulse noise from digital images. Optik - International Journal for Light and Electron Optics, 126(9-10), pp.956–961. 2015a.
- [Zhang, 2015b]** Zhang, C., Wang, K. Removal of high-density impulse noise based on switching morphology-mean filter. AEU - International Journal of Electronics and Communications, 69(1), pp.226–235. 2015b.
- [Zhang, 2016]** Zhang, Y. Fuzzy density weight-based support vector regression for image denoising. Information Sciences, 339, pp.175–188. 2016.

A. Acrónimos

AD	Anisotropic Diffusion
AGF	Adaptive Gaussian Filter
AMF	Adaptive Median Filter
AWA	Adaptive Weighted Algorithm
AWGN	Additive White Gaussian Noise
BNE	Block Based Noise Estimation
CENIDET	Centro Nacional de Investigación y Desarrollo Tecnológico
CIC	Centro de Investigación en Computación
CO	Filtro Morfológico close-open
COMIA	Congreso Mexicano de Inteligencia Artificial
DBA	Decision-Based Algorithm
DSP	Digital Signal Processor
ERID	Enhanced Rank Impulse Detector
FIDRM	Detector Difuso de Ruido Impulsivo
GA	Genetic Algorithm
ICM	Intersection Cortical Model
IICM	Modified Intersection Cortical Model
ICMD-M	Intersection Cortical Model and Denoising by Median
ICMEAE	International Conference on Mechatronics, Electronics and Automotive Engineering
INAOE	Instituto Nacional de Astrofísica, Óptica y Electrónica
IPN	Instituto Politécnico Nacional
KVT	Método de Keny de Mediana Adaptativo Difuso
L&A	Linear-Attenuate
MAE	Mean Absolute Error
MCPR	Mexican Conference on Pattern Recognition

MDBUTMF	Modified Decision-Based Unsymmetric Trimmed Median Filter
MF	Standard Median Filter
MSE	Mean Square Error
NMSE	Normalized Mean Square Error
OC	Filtro Morfológico open-close
PCM	Pixel Correspondance Metric
PCNN	Pulse-Coupled Neural Network
PCNNNI	Pulse-Coupled Neural Network with Null Interaction
PDE	Partial Differential Equation
PSNR	Peak Signal-to-Noise Ratio
RNA	Red Neuronal Artificial
RGB	Formato de Color Red Green Blue
SMIA	Sociedad Mexicana de Inteligencia Artificial
SCM	Spiking Cortical Model
SNR	Signal Noise Ratio
SMF	Switching Median Filter
TM	Time Matrix