

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA
FACULTAD DE CIENCIAS QUIMICAS E INGENIERIA
MAESTRIA Y DOCTORADO EN CIENCIAS E INGENIERIA



NUEVOS MÉTODOS DE RECONOCIMIENTO DE
HUELLAS EMPLEANDO UN ENFOQUE HÍBRIDO
INTELIGENTE

TESIS

PARA OBTENER EL GRADO DE
DOCTOR EN CIENCIAS

PRESENTA:

MIGUEL ANGEL LOPEZ RAMIREZ

DIRECTOR DE TESIS

DRA. ELBA PATRICIA MELIN OLMEDA

CO-DIRECTOR DE TESIS

DR. GUILLERMO LICEA SANDOVAL

TIJUANA, B.C.

AGOSTO 2009

AGRADECIMIENTOS

A mis Codirectores de Tesis Doctores Elba Patricia Melin Olmeda y Guillermo Licea Sandoval por su apoyo y orientación para el desarrollo de este trabajo de Tesis.

En especial agradezco a los Doctores Oscar Castillo López y Elba Patricia Melin Olmeda por su apoyo incondicional, orientación científica, e intelectual durante mi formación en el Posgrado.

Agradezco al Doctor Antonio Rodríguez Díaz por su apoyo, y orientación.

A mis amigos y compañeros en especial a: Juan Ramón Castro Rodríguez, Arnulfo Alanís Garza, Oscar Humberto Montiel Ross, Roberto Sepúlveda Cruz, Fevrier Valdez Acosta, y Mario García Valdez.

Al personal de la Coordinación de Posgrado e Investigación de la FCQI por las facilidades y el apoyo otorgado en la estancia de mis estudios de Doctorado.

A la Coordinación de Posgrado e Investigación de la Universidad Autónoma de Baja California y a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Tijuana por el apoyo otorgado para concluir los estudios de Doctorado en Ciencias e Ingeniería.

DEDICATORIA

A mi esposa y a mis hijos por su amor y apoyo incondicional.

A mis padres por su dedicación y amor.

A mi familia.

RESUMEN

Se presenta un nuevo método híbrido inteligente Basado en un Integrador de respuesta de lógica difusa Tipo-1 (FIS) y lógica difusa Tipo-2 por intervalos (IT2FIS), para un Sistema de Reconocimiento de Patrones para Huellas Digitales, utilizando una arquitectura de Redes Neuronales Ensemble (ENN) para el entrenamiento de las huellas digitales. Se utilizaron algoritmos genéticos jerárquicos para la optimización de las funciones de membresía de los integradores de respuestas de lógica difusa tipo-1 y tipo-2 por intervalos, también se utilizaron algoritmos genéticos jerárquicos para optimizar las reglas de los integradores de respuesta de lógica difusa tipo-1 y lógica difusa tipo-2 por intervalos. El sistema ENN y los integradores de respuesta genético-difusos trabajan juntos en una arquitectura híbrida inteligente, donde las huellas digitales son entrenadas en la ENN y la respuesta es obtenida por los integradores genético-difusos. Se agregaron diferentes niveles de ruido a las huellas digitales para evaluar el sistema híbrido inteligente, primero se probó con el ruido borroso en movimiento lineal con niveles de 0 a 90 píxeles de distancia, después con el ruido borroso en movimiento radial con niveles de 0 a 90 píxeles de radio, y por último con niveles de ruido borroso con movimiento gaussiano de 0 a 10 píxeles de distancia. Para evaluar el sistema se utilizó la relación Señal a Ruido SNR en db para imágenes, y la fiabilidad para reconocimiento de patrones.

ABSTRACT

A new method based on hybrid intelligent systems is presented for response integration of the type-1 fuzzy logic (FIS) and type-2 fuzzy logic by intervals (IT2FIS), for a Pattern Recognition System of the fingerprints, using an architecture of the ensemble neural networks (ENN) for the training of the fingerprints, and Hierarchical genetic algorithms for the optimization of the membership functions of the type fuzzy logic for response integration of the logic Type-1 and Type-2 by intervals, also hierarchical genetic algorithms were utilized to optimize the fuzzy rules of the fuzzy logic response integration of the logic Type-1 and Type-2 by intervals. The ensemble neural network and response integration of the fuzzy logic Type-1 and Type-2 with the genetic algorithm work together in an hybrid intelligent architecture, where the fingerprints are the input to the ENN and the output is obtained for the response integration fuzzy-genetic. We added different noise levels to the fingerprints to evaluate the hybrid intelligent system, we first added the blur motion noise with levels from 0 to 90 pixels of distance, we later added the blur radial noise with levels from 0 to 90 pixels of radio, and we last added the blur Gaussian noise with levels 0 to 10 pixels of distance., to evaluate the hybrid intelligent system was used the relation signal to noise ratio SNR in decibels for images, and the reliability for pattern recognition systems.

AGRADECIMIENTOS.....	2
DEDICATORIA.....	3
RESUMEN	4
ABSTRACT	5
INDICE.....	6
LISTA DE FIGURAS	10
LISTA DE TABLAS.....	15
CAPÍTULO 1	16
INTRODUCCION.....	16
1.1 CONTEXTO GENERAL	16
1.2 DESCRIPCION DEL PROBLEMA.....	17
1.3 SOLUCION PROPUESTA	19
1.4 APORTACIONES DEL TRABAJO	21
CAPÍTULO 2	26
MODELO PROPUESTO SISTEMA HIBRIDO ENN-IT2FIS.....	26
2.1 DEFINICION DEL MODELO ENN-IT2FIS.....	26
2.1.1 ENN-IT2FIS (UTILIZANDO ALGORITMOS GENÉTICOS PARA OPTIMIZAR LAS FUNCIONES DE MEMBRESÍA)	33
2.1.1.1 DISEÑO DEL ALGORITMO GENÉTICO PARA OPTIMIZACIÓN DE LAS FUNCIONES DE MEMBRESÍA DEL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA	34

2.1.1.2 ESTRUCTURA DEL CROMOSOMA PARA LÓGICA DIFUSA TIPO-1.....	35
2.1.1.3 ESTRUCTURA DEL CROMOSOMA PARA LÓGICA DIFUSA TIPO-2.....	35
2.1.2 ENN-IT2FIS (UTILIZANDO ALGORITMOS GENÉTICOS PARA OPTIMIZAR LAS REGLAS DIFUSAS).....	36
2.1.2.1 ESTRUCTURA DEL CROMOSOMA PARA LÓGICA DIFUSA TIPO-1	36
2.1.2.2 ESTRUCTURA DEL CROMOSOMA PARA LÓGICA DIFUSA TIPO-2	34
CAPÍTULO 3	39
SIMULACION DE RESULTADOS	39
3.1 CONDICIONES GENERALES DE LOS EXPERIMENTOS.....	39
3.2 EXPERIMENTOS DE ENN-FIS (UTILIZANDO ALGORITMOS GENÉTICOS PARA OPTIMIZAR LAS FUNCIONES DE MEMBRESÍA).....	40
3.2.1 INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1.....	40
3.2.2 INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 (UTILIZANDO ALGORITMOS GENÉTICOS PARA OPTIMIZAR SUS FUNCIONES DE MEMBRESÍA).....	41
3.2.3 INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2.....	43

3.2.4 INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 (UTILIZANDO ALGORITMOS GENÉTICOS PARA OPTIMIZAR SUS FUNCIONES DE MEMBRESÍA).....	44
3.2.5 SIMULACIÓN DE RESULTADOS CON RUIDO BORROSO CON MOVIMIENTO LINEAL USANDO GAs (OPTIMIZAR FUNCIONES DE MEMBRESÍA)	46
3.2.6 SIMULACIÓN DE RESULTADOS CON RUIDO BORROSO EN MOVIMIENTO RADIAL USANDO GAs (OPTIMIZAR FUNCIONES DE MEMBRESÍA)	47
3.2.7 SIMULACIÓN DE RESULTADOS CON RUIDO BORROSO EN MOVIMIENTO GAUSSIANO USANDO GAs (OPTIMIZAR FUNCIONES DE MEMBRESÍA).....	50

3.3 EXPERIMENTOS DE ENN-IT2FIS (UTILIZANDO ALGORITMOS GENÉTICOS PARA OPTIMIZAR LAS REGLAS DIFUSAS).....	52
3.3.1 SIMULACIÓN DE RESULTADOS CON RUIDO BORROSO EN MOVIMIENTO LINEAL USANDO GAs (OPTIMIZAR REGLAS DIFUSAS).....	54
3.3.2 SIMULACIÓN DE RESULTADOS CON RUIDO BORROSO EN MOVIMIENTO RADIAL USANDO GAs (OPTIMIZAR REGLAS DIFUSAS).....	56
3.3.3 SIMULACIÓN DE RESULTADOS CON RUIDO BORROSO EN MOVIMIENTO GAUSSIANO USANDO GAs (OPTIMIZAR REGLAS DIFUSAS).....	58
CAPÍTULO 4	61
ANALISIS Y COMPARACION DE RESULTADOS.....	61
4.1 ANALISIS DE RESULTADOS OBTENIDOS CON FIS-ENN (UTILIZANDO ALGORITMOS GENÉTICOS PARA OPTIMIZAR LAS FUNCIONES DE MEMBRESÍA).....	61
4.2 ANALISIS DE RESULTADOS OBTENIDOS CON ENN-IT2FIS (UTILIZANDO ALGORITMOS GENÉTICOS PARA OPTIMIZAR LAS FUNCIONES DE MEMBRESÍA).....	62
4.3 ANALISIS DE RESULTADOS OBTENIDOS CON ENN-FIS-GA (UTILIZANDO ALGORITMOS GENÉTICOS PARA OPTIMIZAR LAS REGLAS DIFUSAS)	66

4.4 ANALISIS DE RESULTADOS OBTENIDOS CON ENN-IT2FIS (UTILIZANDO ALGORITMOS GENÉTICOS PARA OPTIMIZAR LAS REGLAS DIFUSAS)	68
4.5 EVALUACIÓN DE RESULTADOS USANDO SNR (RELACIÓN SEÑAL A RUIDO)	71
4.6 FIABILIDAD DEL SISTEMA DE RECONOCIMIENTO DE PATRONES	78
4.7 PRUEBA ESTADÍSTICA T STUDENT PARA CONTRASTAR LOS RESULTADOS DE LOS INTEGRADORES DE RESPUESTA LÓGICA DIFUSA TIPO-2 Y TIPO-1	80
CAPÍTULO 5	82
CONCLUSIONES.....	82
REFERENCIAS	85
APENDICE	90

LISTA DE FIGURAS

FIGURA 2.1. CLASIFICACIÓN DE SISTEMAS HÍBRIDOS INTELIGENTES (HIS)	27
FIGURA 2.2. CLASIFICACIÓN MODULAR DE SISTEMAS HÍBRIDOS INTELIGENTES PROCESO SECUENCIAL (HIS)	27
FIGURA 2.3. CLASIFICACIÓN MODULAR DE SISTEMAS HÍBRIDOS INTELIGENTES ACOPLADOS PASIVAMENTE.....	29
FIGURA 2.4. MODELO PARA CADA UNO DE LOS TRES MÓDULOS DE LA RED NEURONAL ENSEMBLE	32
FIGURA 2.5 MODELO PROPUESTO DEL SISTEMA HIBRIDO INTELIGENTE UTILIZANDO UNA RED NEURONAL ENSEMBLE Y COMO MÉTODO DE INTEGRACIÓN UN SISTEMA DE LÓGICA DIFUSA TIPO-1 O TIPO-2.....	32
FIGURA2.6 ESTRUCTURA DEL CROMOSOMA PARA OPTIMIZAR LAS FUNCIONES DE MEMBRESÍA DEL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1	35
FIGURA2.7 ESTRUCTURA DEL CROMOSOMA PARA OPTIMIZAR LAS FUNCIONES DE MEMBRESÍA DEL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2.....	36
FIGURA 2.8 ESTRUCTURA DEL CROMOSOMA PARA OPTIMIZAR LAS REGLAS DIFUSAS DEL SISTEMA DE LÓGICA DIFUSA TIPO-1... ..	37
FIGURA 2.9 ESTRUCTURA DEL CROMOSOMA PARA OPTIMIZAR LAS REGLAS DIFUSAS DEL SISTEMA DE LÓGICA DIFUSA TIPO-2	38
FIGURA 3.1 BASE DE DATOS DE HUELLAS DIGITALES UTILIZADA PARA LOS EXPERIMENTOS FCV2000	39
FIGURA 3.2 FUNCIONES DE MEMBRESÍA ENTRADA MODULO1	41
FIGURA 3.3 FUNCIONES DE MEMBRESÍA ENTRADA MODULO2	41
FIGURA 3.4 FUNCIONES DE MEMBRESÍA ENTRADA MODULO3	41
FIGURA 3.5 FUNCIONES DE MEMBRESÍA DE LA VARIABLE DE SALIDA	41
FIGURA 3.6. RESULTADOS DE LA CONVERGENCIA DEL ALGORITMO GENÉTICO PARA EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1.....	42
FIGURA 3.7 FUNCIÓN DE MEMBRESÍA DE ENTRADA MODULO 1 USANDO EL GA.....	42
FIGURA 3.8 FUNCIONES DE MEMBRESÍA DE ENTRADA MODULO 2	42
FIGURA 3.9 FUNCIONES DE MEMBRESÍA DE ENTRADA MODULO 3	43

FIGURA 3.10 FUNCIONES DE MEMBRESÍA DE SALIDA.....	43
FIGURA 3.11 FUNCIONES DE MEMBRESÍA TIPO-2 ENTRADA MODULO1	43
FIGURA 3.12 FUNCIONES DE MEMBRESÍA TIPO-2 ENTRADA MODULO2	43
FIGURA 3.13 FUNCIONES DE MEMBRESÍA TIPO-2 ENTRADA MODULO3	43
FIGURA 3.14 FUNCIONES DE MEMBRESÍA TIPO-2 DE SALIDA.....	44
FIGURA 3.15 RESULTADOS DE LA CONVERGENCIA DEL ALGORITMO GENÉTICO PARA EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2	44
FIGURA 3.16 FUNCIONES DE MEMBRESÍA TIPO-2 ENTRADA MODULO1 USANDO GAs.....	45
FIGURA 3.17 FUNCIONES DE MEMBRESÍA TIPO-2 ENTRADA MODULO2 USANDO GAs.....	45
FIGURA 3.18 FUNCIONES DE MEMBRESÍA TIPO-2 ENTRADA MODULO3 USANDO GAs.....	45
FIGURA 3.19 FUNCIONES DE MEMBRESÍA TIPO-2 ENTRADA MODULO1 USANDO GAs.....	46
FIGURA 3.20 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO LINEAL DE 10 PÍXELES DE DESPLAZAMIENTO).....	46
FIGURA 3.21 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LOGICA DIFUSA TIPO-1 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO LINEAL DE 50 PÍXELES DE DESPLAZAMIENTO)	47
FIGURA 3.22 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO LINEAL DE 10 PÍXELES DE DESPLAZAMIENTO).....	47
FIGURA 3.23 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO LINEAL DE 50 PÍXELES DE DESPLAZAMIENTO).....	48

FIGURA 3.24 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO RADIAL DE 10 PÍXELES DE DESPLAZAMIENTO).....	48
FIGURA 3.25 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO RADIAL DE 50 PÍXELES DE DESPLAZAMIENTO).....	49
FIGURA 3.26 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO RADIAL DE 10 PÍXELES DE DESPLAZAMIENTO).....	49
FIGURA 3.27 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO RADIAL DE 50 PÍXELES DE DESPLAZAMIENTO).....	50
FIGURA 3.28 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO GAUSSIANO 5 PÍXELES DE DESPLAZAMIENTO).....	51
FIGURA 3.29 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO GAUSSIANO 10 PÍXELES DE DESPLAZAMIENTO).....	51
FIGURA 3.30 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO GAUSSIANO 5 PÍXELES DE DESPLAZAMIENTO).....	52
FIGURA 3.31 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO GAUSSIANO 10 PÍXELES DE DESPLAZAMIENTO).....	52

FIGURA 3.32	RESULTADOS DE LA CONVERGENCIA DEL ALGORITMO GENÉTICO PARA OPTIMIZAR LAS REGLAS DEL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1.....	54
FIGURA 3.33	RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO LINEAL DE 10 PÍXELES DE DESPLAZAMIENTO).....	54
FIGURA 3.34	RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LOGICA DIFUSA TIPO-1 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO LINEAL DE 50 PÍXELES DE DESPLAZAMIENTO).....	55
FIGURA 3.35	RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO LINEAL DE 10 PÍXELES DE DESPLAZAMIENTO).....	55
FIGURA 3.36	RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO LINEAL DE 50 PÍXELES DE DESPLAZAMIENTO).....	56
FIGURA 3.37	RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO RADIAL DE 10 PÍXELES DE DESPLAZAMIENTO).....	56
FIGURA 3.38	RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO RADIAL DE 50 PÍXELES DE DESPLAZAMIENTO).....	57
FIGURA 3.39	RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO RADIAL DE 10 PÍXELES DE DESPLAZAMIENTO).....	57
FIGURA 3.40	RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 (CON UN NIVEL DE	

RUIDO BORROSO CON MOVIMIENTO RADIAL DE 50 PÍXELES DE DESPLAZAMIENTO).....	58
FIGURA 3.41 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO GAUSSIANO 5 PÍXELES DE DESPLAZAMIENTO).....	58
FIGURA 3.42 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO GAUSSIANO 10 PÍXELES DE DESPLAZAMIENTO).....	59
FIGURA 3.43 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO GAUSSIANO 5 PÍXELES DE DESPLAZAMIENTO).....	59
FIGURA 3.44 RESULTADOS EXPERIMENTALES DE LAS HUELLAS USANDO EL INTEGRADOR DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 (CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO GAUSSIANO 10 PÍXELES DE DESPLAZAMIENTO).....	60
FIGURA 4.1. COMPARACIÓN DE RESULTADOS ENTRE TIPO-1 Y TIPO-2 USANDO GAS CON RUIDO RUIDO BORROSO CON MOVIMIENTO LINEAL.....	64
FIGURA 4.2. COMPARACIÓN DE RESULTADOS ENTRE TIPO-1 Y TIPO-2 USANDO GAS CON RUIDO RUIDO BORROSO CON MOVIMIENTO RADIAL	65
FIGURA 4.3 COMPARACIÓN DE RESULTADOS ENTRE LÓGICA DIFUSA TIPO-1 Y TIPO-2 USANDO GAS CON DIFERENTES NIVELES DE RUIDO BORROSO CON MOVIMIENTO LINEAL	67
FIGURA 4.4 COMPARACIÓN DE RESULTADOS ENTRE LÓGICA DIFUSA TIPO-1 Y TIPO-2 USANDO GAS CON DIFERENTES NIVELES DE RUIDO BORROSO CON MOVIMIENTO RADIA	68
FIGURA 4.5 COMPARACIÓN DE RESULTADOS ENTRE LÓGICA DIFUSA TIPO-1 Y TIPO-2 USANDO GAS CON DIFERENTES NIVELES DE RUIDO BORROSO CON MOVIMIENTO LINEA	68
FIGURA 4.6 COMPARACIÓN DE RESULTADOS ENTRE LÓGICA DIFUSA TIPO-1 Y TIPO-2 USANDO GAS CON DIFERENTES NIVELES DE RUIDO BORROSO CON MOVIMIENTO LINEAL	70

FIGURA 4.7	BASE DE DATOS HUELLAS DIGITALES ORIGINALES CLASIFICADAS POR PERSONAS.....	73
FIGURA 4.8	COMPARATIVO INTEGRADORES DE RESPUESTA DE LOGICA DIFUSA TIPO-1 Y TIPO-2 CON RESPECTO A LA RELACIÓN SEÑAL A RUIDO PROMEDIO ASNR PARA LAS IMÁGENES DE LA PERSONA 1.....	74
FIGURA 4.9	COMPARATIVO INTEGRADORES DE RESPUESTA DE LOGICA DIFUSA TIPO-1 Y TIPO-2 CON RESPECTO A LA RELACIÓN SEÑAL A RUIDO PROMEDIO ASNR PARA LAS IMÁGENES DE LA PERSONA 2.....	74
FIGURA 4.10	COMPARATIVO INTEGRADORES DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 Y TIPO-2 CON RESPECTO A LA RELACIÓN SEÑAL A RUIDO PROMEDIO ASNR PARA LAS IMÁGENES DE LA PERSONA 3.....	74
FIGURA 4.11	COMPARATIVO INTEGRADORES DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 Y TIPO-2 CON RESPECTO A LA RELACIÓN SEÑAL A RUIDO PROMEDIO ASNR PARA LAS IMÁGENES DE LA PERSONA 4.....	75
FIGURA 4.12	COMPARATIVO INTEGRADORES DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 Y TIPO-2 CON RESPECTO A LA RELACIÓN SEÑAL A RUIDO PROMEDIO ASNR PARA LAS IMÁGENES DE LA PERSONA 5.....	75
FIGURA 4.13	COMPARATIVO INTEGRADORES DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 Y TIPO-2 CON RESPECTO A LA RELACIÓN SEÑAL A RUIDO PROMEDIO ASNR PARA LAS IMÁGENES DE LA PERSONA 6.....	75
FIGURA 4.14	COMPARATIVO INTEGRADORES DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 Y TIPO-2 CON RESPECTO A LA RELACIÓN SEÑAL A RUIDO PROMEDIO ASNR PARA LAS IMÁGENES DE LA PERSONA 7.....	76
FIGURA 4.15	COMPARATIVO INTEGRADORES DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 Y TIPO-2 CON RESPECTO A LA RELACIÓN SEÑAL A RUIDO PROMEDIO ASNR PARA LAS IMÁGENES DE LA PERSONA 8.....	76
FIGURA 4.16	COMPARATIVO INTEGRADORES DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 Y TIPO-2 CON RESPECTO A LA RELACIÓN SEÑAL A RUIDO PROMEDIO ASNR PARA LAS IMÁGENES DE LA PERSONA 9.....	76
FIGURA 4.17	COMPARATIVO INTEGRADORES DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 Y TIPO-2 CON RESPECTO A LA RELACIÓN SEÑAL A RUIDO PROMEDIO ASNR PARA LAS IMÁGENES DE LA PERSONA 10.....	77
FIGURA 4.18	COMPARACIÓN DE LA FIABILIDAD DE LOS INTEGRADORES DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 Y TIPO-1	79

LISTA DE TABLAS

TABLA 4.1. COMPARACIÓN DE RESULTADOS ENTRE EL INTEGRADOR DE RESPUESTAS DE LÓGICA DIFUSA TIPO-1 USANDO GAs Y SIN OPTIMIZAR SUS FUNCIONES DE MEMBRESIA CON NIVELES DE RUIDO BORROSO CON MOVIMIENTO LINEAL	61
TABLA 4.2. COMPARACIÓN DE RESULTADOS ENTRE EL INTEGRADOR DE RESPUESTAS DE LÓGICA DIFUSA TIPO-1 USANDO GAs Y SIN OPTIMIZAR SUS FUNCIONES DE MEMBRESIA CON NIVELES DE RUIDO BORROSO CON MOVIMIENTO RADIAL.....	62
TABLA 4.3. COMPARACIÓN DE RESULTADOS ENTRE EL INTEGRADOR DE RESPUESTAS DE LÓGICA DIFUSA TIPO-1 USANDO GAs Y SIN OPTIMIZAR SUS FUNCIONES DE MEMBRESIA CON NIVELES DE RUIDO BORROSO MOVIMIENTO GAUSSIANO.....	62
TABLA 4.4. COMPARACIÓN DE RESULTADOS ENTRE TIPO-1 Y TIPO-2 USANDO GAS CON RUIDO RUIDO BORROSO CON MOVIMIENTO LINEAL	63
TABLA 4.5. COMPARACIÓN DE RESULTADOS ENTRE TIPO-1 Y TIPO-2 USANDO GAs CON RUIDO RUIDO BORROSO CON MOVIMIENTO RADIAL	64
TABLA 4.6. COMPARACIÓN ENTRE TIPO-1 Y TIPO-2 USANDO GAs CON RUIDO BORROSO CON MOVIMIENTO LINEAL	66
TABLA 4.7. COMPARACIÓN ENTRE TIPO-1 Y TIPO-2 USANDO GAS CON RUIDO BORROSO CON MOVIMIENTO RADIAL	67
TABLA 4.8. COMPARACIÓN ENTRE TIPO-1 Y TIPO-2 CON LOS PARÁMETROS MODIFICADOS POR CADA MODULO USANDO NIVELES DE RUIDO BORROSO CON MOVIMIENTO LINEAL	69
TABLA 4.9. COMPARACIÓN ENTRE TIPO-1 Y TIPO-2 CON LOS PARÁMETROS MODIFICADOS POR CADA MODULO USANDO NIVELES DE RUIDO BORROSO CON MOVIMIENTO RADIAL	69
TABLA 4.10. RELACIÓN SEÑAL A RUIDO PROMEDIO ASNR PARA CADA UNA DE LAS IMÁGENES DE LAS HUELLAS DIGITALES DE LAS PERSONAS CON UN NIVEL DE RUIDO BORROSO CON MOVIMIENTO LINEAL DE 10 A 90 PÍXELES DE DISTANCIA PARA LOS INTEGRADORES DE RESPUESTA DE LÓGICA DIFUSA TIPO-1 Y TIPO-2	72

TABLA 4.11. COMPARACION DE LA FIABILIDAD PARA LOS INTEGRADORES DE RESPUESTA DE LOGICA DIFUSA TIPO-1 Y TIPO-2 CON LOS PARAMETROS MODIFICADOS POR CADA MODULO DE LA RED NEURONAL ENSEMBLE, USANDO NIVELES DE RUIDO BORROSO CON MOVIMIENTO LINEAL 79

TABLA 4.12. COMPARACIÓN DEL VALOR T STUDENT, MEDIA, DESVIACIÓN ESTÁNDAR Y VARIANZA ENTRE LOS INTEGRADORES DE RESPUESTA DE LÓGICA DIFUSA TIPO-2 Y TIPO-1 81

CAPÍTULO 1. INTRODUCCIÓN

1.1 Contexto General

En este trabajo de Tesis se aborda la problemática para reconocimiento de patrones utilizando modelación híbrida neuro-genética difusa tipo-2 por intervalos, como metodología de computación suave (*Soft Computing*), aprovechando a los sistemas difusos tipo-2 por intervalos como integradores de respuestas para sistemas de reconocimiento de patrones, usando una arquitectura de una red neuronal artificial ensemble, y algoritmos genéticos para optimizar las reglas y las funciones de membresía del sistema de lógica difusa tipo-1 y tipo-2 por intervalos. El trabajo se divide en cinco capítulos y un apéndice, a continuación se resumen los contenidos de cada uno.

El capítulo uno corresponde a la introducción, donde se presentan los conceptos básicos sobre los que se fundamenta el trabajo de investigación realizado. Se Comenzará presentando el entorno de la problemática que se pretende resolver, seguido de las alternativas de solución que se proponen en esta investigación.

En el capítulo dos se presentan las principales aportaciones de la tesis. En primer lugar se describe la arquitectura híbrida inteligente para reconocimiento de patrones, posteriormente se presenta el modelo de la red neuronal ensemble utilizada, el modelo del integrador de respuestas de lógica difusa tipo-1 y tipo-2 por intervalos (IT2FIS). Posteriormente se describen los métodos de algoritmos genéticos jerárquicos (HGA) utilizados para optimizar las funciones de membresía de los sistemas de inferencia lógica difusa tipo-1 y tipo-2 por intervalos, y en seguida se presentan los métodos de algoritmos genéticos jerárquicos (HGA) para optimizar las reglas difusas de los integradores de respuesta de los sistemas de inferencia de lógica difusa tipo-1 y tipo-2 por intervalos.

El capítulo tres se describe las condiciones en que se llevaron a cabo los experimentos para modelar el sistema híbrido, haciendo uso de la arquitectura ENN-FIS-GA (Funciones de Membresía), ENN-IT2FIS-GA (Funciones de Membresía), ENN-FIS-GA (Reglas Difusas), ENN-IT2FIS-GA (Reglas Difusas) expuestos en el capítulo dos; en el primero se describe la

forma de trabajar con las huellas digitales que utiliza el sistema híbrido inteligente, después se entrena la red neuronal ensemble buscando la mejor arquitectura moviendo los diferentes parámetros de los módulos de la red neuronal ensemble, número de capas, número de neuronas, funciones de entrenamiento, método de aprendizaje para encontrar la mejor arquitectura de entrenamiento, posteriormente se creó una base de datos de las huellas digitales utilizadas agregándoles diferentes niveles de ruido borroso con movimiento lineal, ruido borroso con movimiento radial y gaussiano para evaluar el sistema híbrido inteligente.

En el capítulo cuatro se hace un análisis de los resultados con la relación señal a ruido S/N para imágenes con la finalidad de evaluar el sistema híbrido inteligente, al final se muestran los resultados de los modelos ENN-FIS-GA (Funciones de Membresía), ENN-IT2FIS-GA (Funciones de Membresía), ENN-FIS-GA (Reglas Difusas), ENN-IT2FIS-GA (Reglas Difusas) para los diferentes tipos de ruido. Los resultados del análisis se organizan en tablas comparativas entre los diferentes modelos de integradores de respuesta.

En el capítulo cinco, se presentan las conclusiones y aportaciones del presente trabajo. Se propondrán también algunas líneas de trabajo futuras.

En el apéndice A se documentan algunos desarrollos auxiliares del capítulo dos.

1.2 Descripción del problema

En general la propuesta de un nuevo método de reconocimiento de patrones para huellas con un enfoque híbrido inteligente es uno de los problemas de mayor interés en numerosas ramas de la ciencia. El método involucra la combinación de las tres áreas más importantes dentro de Soft Computing (Computación Suave), como son Redes Neuronales, Lógica Difusa y Algoritmos Genéticos.

El problema abordado en la investigación es proponer un Nuevo Método de Reconocimiento de Patrones para Huellas con un enfoque Híbrido Inteligente, primero se encontró la mejor Arquitectura de la Red Neuronal Ensemble para la Base de Datos de las Huellas Digitales

Utilizadas, después la investigación se enfoca en el método de integración de respuestas como parte del Sistema Híbrido Inteligente utilizando un Integrador de Respuestas de Lógica Difusa Tipo-1 y Lógica Difusa Tipo-2, posteriormente se utilizaron algoritmos genéticos para Optimizar las funciones de Membresía del Integrador de Respuestas del Sistema de Lógica Difusa Tipo-1 y Lógica Difusa Tipo-2. Después se utilizaron los Algoritmos Genéticos para optimizar las reglas del Integrador de Respuestas del Sistema de Lógica difusa Tipo-1 y Lógica difusa tipo-2.

Cuando el ser humano trata de describir los fenómenos naturales o hacer inferencias a partir de su estudio, se enfrenta a diversos tipos de incertidumbre originadas por la información imperfecta, por los modelos usados para hacer inferencias o por la interpretación dada a esos modelos.

En la práctica ningún sistema de Adquisición de imágenes es perfecto y las imágenes que son obtenidas la mayoría están degradadas, por ejemplo con ruido borroso con movimiento lineal, turbulencia atmosférica, y difracción óptica [1]. Muchos métodos para restauración de imágenes se han desarrollado, incluyendo filtros de error cuadrático medio, la técnica de descomposición de valor singular [2]. En todos esos métodos, el sistema de adquisición de imágenes es modelado como un sistema lineal, y los sistemas son caracterizados por una función de dispersión. Las técnicas de restauración tratan con la inversión del proceso de degradación. La imperfección en los datos al adquirir las imágenes pueden ser debido a la incertidumbre, la imprecisión, la vaguedad y a la inconsistencia de los datos. Normalmente el nivel de incertidumbre varía dependiendo del contexto y problema a tratar. Esta imprecisión o incertidumbre puede venir dada por la inclusión de ruido en las imágenes a tratar. Los sistemas de lógica difusa tipo-1 y tipo-2 por intervalos (IT2FLS), combinados con técnicas como las redes neuronales (NN) y los algoritmos genéticos (GA) han demostrado su habilidad para resolver diferentes tipos de problemas tales como control [3,4,5], predicción [6,7,8], procesamiento de señales e imágenes [9,10], procesos industriales [11,12,13], etc. En muchos de los casos, la clave del éxito era la incorporación del conocimiento experto humano. En la última década, ha existido un gran interés en incluir

capacidad de aprendizaje a los IT2FLS [14, 15, 16,17]. Esto se ha logrado por medio del desarrollo de técnicas híbridas que incluyen los IT2FLS junto con técnicas complementarias como lo son las redes neuronales, los algoritmos evolutivos o los métodos probabilísticos [17, 18, 19,20].

Existe consenso entre los investigadores en que se pueden obtener sistemas más inteligentes por medio de la hibridación de metodologías de computación suave [21,22,23,24,25], haciendo de este modo que las debilidades de unos sistemas se compensen con las bondades de otros. Los sistemas ENN-IT2FIS (utilizando algoritmos genéticos para optimizar las funciones de membresía para un caso y optimizando las reglas difusas para el otro) combinan los sistemas de inferencia difusos tipo-2 por intervalos (IT2FIS), y las redes neuronales ensemble, aprovechando las características sobresalientes de cada modelo.

1.3 Solución propuesta

En este trabajo de tesis se presentan dos metodologías de un sistema híbrido neurodifuso tipo-2 por intervalos (ENN-IT2FIS) para un sistema de reconocimiento de patrones. En la primera, la red neuronal ensemble que cuenta con tres módulos (redes neuronales monolíticas) se entrena con una base de datos de huellas para encontrar su mejor arquitectura, y las activaciones de salida de cada módulo son integrados con un sistema difuso tipo-2 por intervalos, utilizando algoritmos genéticos para optimizar las funciones de membresía del sistema difuso tipo-2 por intervalos.

En la segunda el algoritmo genético se utiliza para optimizar las reglas difusas del integrador de respuesta difusas (ENN-IT2FIS). Para evaluar la validez de la solución propuesta, se agregaron tres tipos de niveles de ruido a la base de datos de huellas ruido borroso con movimiento lineal, ruido borroso con movimiento radial, y ruido borroso movimiento gaussiano, los niveles del ruido utilizados fueron de 10 a 90 píxeles de distancia, se comprobó el comportamiento del sistema de reconocimiento de patrones para los diferentes integradores de respuestas difusos ENN-FIS, y ENN-IT2FIS. Se evaluó el sistema de Reconocimiento de

Patrones utilizando la relación señal a ruido (SNR) en decibels para imágenes [43,44] y se obtuvo la fiabilidad del sistema híbrido para los integradores de respuesta de lógica difusa tipo-2 y tipo-1[45].

El alcance del presente trabajo se enfocó al análisis y comportamiento de un sistema de reconocimiento de patrones para huellas por medio de una arquitectura híbrida inteligente, utilizando la combinación de diferentes métodos de computación suave como lo son redes neuronales, lógica difusa tipo-1 y tipo-2 y algoritmos genéticos jerárquicos (HGA).

1.4 Aportaciones del trabajo

Esta investigación ha proporcionado varias contribuciones de interés para la comunidad científica internacional a continuación se presentan las publicaciones que se han derivado del presente trabajo:

Revistas Internacionales:

1. **Response Integration in Ensemble Neural Networks using The Sugeno Integral and Difuso Inference System for Pattern Recognition**, Miguel Lopez, Patricia Melin; *Journal of Automation, Mobile Robotics and Intelligent Systems*, Vol. 2 No. 1, ISSN 1897-8649.

Capítulos de libro:

1. **Optimization of Response Integration with Difuso Logic in Ensemble Neural Networks Using Genetic Algorithms**. Miguel Lopez, Patricia Melin, Oscar Castillo.; In *Soft Computing for Hybrid Intelligent Systems*. Book Chapter, Volume 154/2008: 129-150, September 2008. Springer Berlin / Heidelberg. ISBN: 978-3-540-70811-7, ISSN: 1860-949X (Print) 1860-9503 (Online).
2. **A Method for Creating Ensemble Neural Networks Using a Sampling Data Approach**, Miguel Lopez, Patricia Melin, Oscar Castillo:. *Analysis and Design of Intelligent Systems using Soft Computing Techniques* 2007: 355-364, *Theoretical Advances and Applications of Difuso Logic and Soft Computing*, Selection of Papers from IFSA 2007. Advances in Soft Computing Vol 42 Springer 2007, ISBN 978-3-540-72433-9.

Congresos Internacionales:

1. **Pattern Recognition in Ruido borroso con movimiento lineal Noisy using Difuso Methods for Response Integration in Ensemble Neural Networks**, Miguel Lopez, Patricia Melin, NAFIPS 2009, Portugal.

2. **Topology Optimization of Difuso Systems for Response Integration in Ensemble Neural Networks: The Case of Fingerprint Recognition**, Miguel Lopez, Patricia Melin, NAFIPS 2008, New York, ISBN: 978-1-4244-2352-1: 61002-61007.
3. **Response Integration in Ensemble Neural Networks Using Interval Tipo-2 Difuso Logic**, World Congress on Computational Intelligence (WCCI 2008, IEEE.), Hong Kong, IJCNN 2008: 1503-1508.
4. **Analysis of sampling methods in the learning process of ensemble neural networks**, Miguel Lopez, Patricia Melin, In Proceedings NAFIPS 2007, Number 1, pages CD Rom, San Diego, USA, June 2007. ISBN: 1-4244-1214-5, Library of Congress: 2007924575: 604-609.

CAPÍTULO 2. MODELO PROPUESTO SISTEMA HIBRIDO ENN-IT2FIS

2.1 Definición del Modelo ENN-IT2FIS

Un sistema neuro-difuso tipo-2 por intervalo es un modelo híbrido-cooperativo que integra los métodos de razonamiento aproximado de los sistemas de lógica difusa tipo-2 por intervalos (IT2FLS) con las capacidades de aprendizaje de las redes neuronales (NNs) y robustecido cooperativamente por un proceso de aprendizaje basado en algoritmos genéticos jerárquicos (HGAs), los cuales presentan la capacidad de explorar los espacios de búsqueda complejos. Los tres métodos son complementarios y como consecuencia, el enfoque híbrido proporciona un marco de razonamiento aproximado poderoso que tiene capacidades de aprendizaje y generalización, El uso de un método computacional sencillo tiene buenos resultados en la mayoría de las aplicaciones, para aplicaciones mas especializadas se utilizan dos o más métodos de computación inteligente, también llamados métodos híbridos. Dentro de las Arquitecturas de Sistemas Híbridos Inteligentes (HIS) existe una mezcla de modelos, por ejemplo;

- Combinación de de diferentes modelos de Conocimiento y conocimiento de ingeniería.
- Combinación de modelos de conocimiento y redes neuronales.
- Combinación de redes neuronales y lógica difusa, redes neuronales y algoritmos genéticos, lógica difusa y algoritmos genéticos, redes neuronales, lógica difusa y algoritmos genéticos.

Dentro de las arquitecturas de los Sistemas Híbridos Inteligentes (HIS) [26] se encuentra la siguiente clasificación que se muestra a continuación en la figura 2.1:

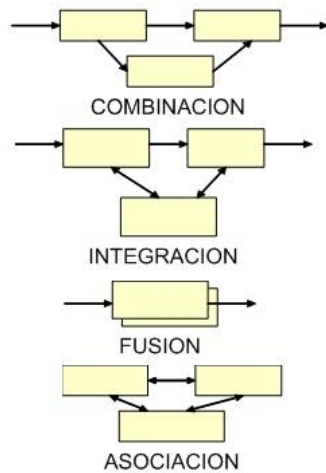


Figura 2.1 Clasificación de Sistemas Híbridos Inteligentes (HIS).

Otra Clasificación de los Sistemas Híbridos Inteligentes es en forma modular como se muestra a continuación en la figura 2.2.

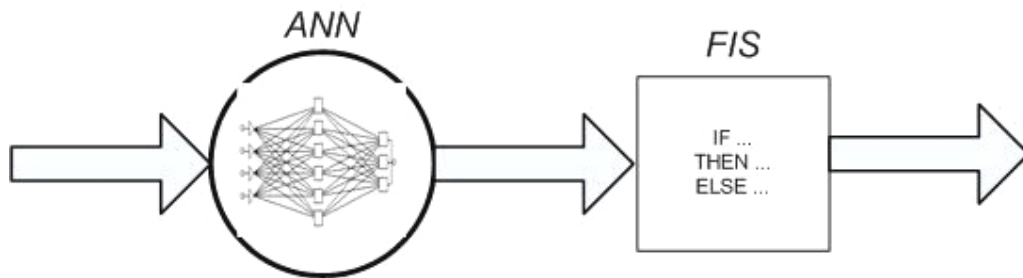


Figura 2.2 Clasificación Modular de Sistemas Híbridos Inteligentes Proceso Secuencial (HIS).

El uso de un método computacional sencillo tiene buenos resultados en la mayoría de las aplicaciones, para aplicaciones mas especializadas se utilizan dos o más métodos de computación inteligente, también llamados métodos híbridos.

CLASIFICACION DE SISTEMAS HIBRIDOS INTELIGENTES MODULAR

- **Configuración secuencial:** La principal característica de esta configuración es el procesamiento de los datos en serie como son enviados de un modulo al próximo. Un modulo actúa como un preprocesador de datos extrayendo las características requeridas dentro de una forma apropiada para el próximo

- Una red neuronal artificial actúa como un preprocesador para un sistema basado en reglas convirtiendo el nivel de la señal de la información dentro de una forma más apropiada para un nivel simbólico de decisiones.

El sistema híbrido inteligente propuesto utiliza un modelo secuencial- combinacional, dentro del modelo secuencial esta una red neuronal artificial ensemble y un sistema difuso (Integrador de Respuesta de Lógica difusa tipo-1 y tipo-2), la red neuronal ensemble que le entrega secuencialmente la información al integrador de respuestas de lógica difusa tipo-1 y tipo-2, y un modelo combinacional entre un algoritmo genético jerárquico (HGA) y el integrador de respuesta de lógica difusa tipo-1 y tipo-2, el HGA optimiza sus funciones de membresía en un caso y las reglas difusas en otro caso, dando un modelo híbrido inteligente secuencial-combinacional.

El modelo secuencial-combinacional utilizado es un modelo híbrido inteligente modular pasivo acoplado, esto es porque es el método más simple de integración debido a la existencia autónoma de los componentes de la red neuronal y el sistema difuso. El método de comunicación es usualmente por el datos compartidos por los componentes, después de que el primer modulo termina su cálculo (red neuronal ensemble) guarda el resultado (activaciones de la red neuronal ensemble) para que sea leído por el segundo modulo (integrador de respuestas difuso tipo-1 y tipo-2). El acoplamiento pasivo no requiere ningún protocolo de control sofisticado para sincronizar los módulos ya que la comunicación es normalmente unidireccional y todos los datos usualmente son depositados de una manera sencilla. Una configuración típica debe involucrar una red neuronal artificial procesando un vector de entrada de datos y guardando las activaciones de salida de la red neuronal, como se muestra en la figura 2.3.

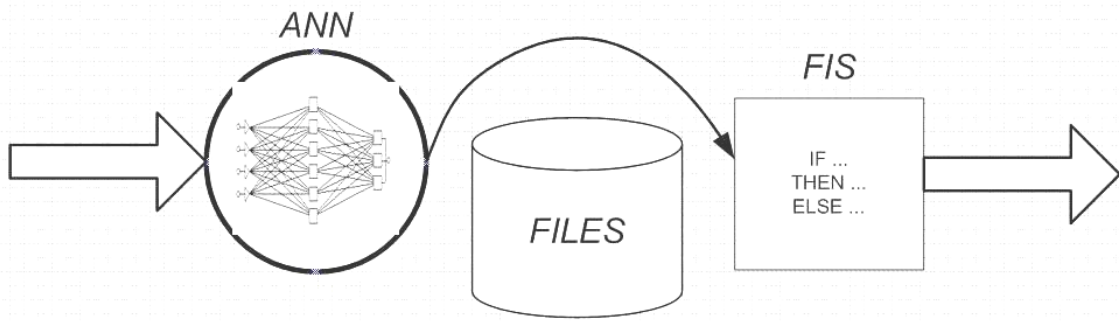


Figura 2.3 Clasificación Modular de Sistemas Híbridos Inteligentes Acoplados Pasivamente.

Usando las redes neuronales se intenta modelar una función desconocida f para un par de datos de entrada salida. Los algoritmos existentes para ese tipo de problemas de regresión son modelos de regresión lineal, redes neuronales y teoría de wavelet. Una red neuronal puede ser recordada como una representación de una función determinada por los factores de sus pesos y su arquitectura [26]. El mapeo completo es caracterizado por una función compuesta relacionando las entradas feed-forward con las salidas.

Esto es $O = f_{\text{compuesta}}(x)$ usando p -capas de mapeo dentro de una red $p+1$ capas feed forward como se muestra a continuación en la ecuación 2.1.

$$O = f^{Lp}(f^{Lp-1} \dots (f^{L1}(x), \dots)) \quad (2.1)$$

Usualmente se entrena una red neuronal con un conjunto de datos de entrenamiento, entrando a la red neuronal, e interpreta la salida de acuerdo a las reglas lógicas del conjunto de entrenamiento [27, 28, 29, 30]. Las técnicas más comúnmente usadas ajustan los parámetros de los pesos de la red neuronal usando el método de back propagation con aprendizaje de mínimos cuadrados (Aprendizaje Supervisado en redes neuronales multicapa) definido en la ecuación 2.2 como:

$$(2.2)$$

Donde k = numero de neuronas de salida.

$$w_{ji}^i(n+1) = w_{ji}^i(n) \dots + \eta \delta^i(n) y_i^{i-1}(n) \quad (2.3)$$

Primero se calcula los pesos de las entradas x_j , usando la ecuación 2.4.

$$X_j = \sum_i y_i W_{ij} \quad (2.4)$$

Donde y_i es el nivel de actividad de la unidad i -ésima de la capa previa y W_{ij} es el peso de la conexión entre la unidad i -ésima y la j -ésima. En seguida, se calcula la actividad de y_j usando la función del total de los pesos de entrada. Típicamente usando la función sigmoide como se muestra en la ecuación 2.5:

$$y_i = \frac{1}{1+e^{-x_j}} \quad (2.5)$$

Una vez que las actividades de todas las unidades de salida han sido determinadas, la red neuronal calcula el error E , el cual está definido por la ecuación 2.6:

$$E = \frac{1}{2} \sum_i (y_i - d_i)^2 \quad (2.6)$$

Donde y_i es el nivel de actividad de la i -ésima unidad en la última capa y d_i es la salida deseada de la i -ésima unidad. El algoritmo de back-propagation consiste de cuatro pasos:

1. Calcula los cambios de error como la actividad de los cambios de unidad de salida. La derivada del error (EA) es la diferencia entre la actividad actual y la deseada como se muestra en la ecuación 2.7.

$$EA_i = \frac{\partial E}{\partial y_i} = y_i - d_i \quad (2.7)$$

2. Calcula el cambio de error total de la entrada recibida con respecto al cambio de señal de unidad de salida. Esta cantidad EI es la respuesta del primer paso multiplicada por la razón a la cual la unidad de salida cambia con respecto al cambio total de la señal de entrada como se muestra en la ecuación 2.8.

$$EI_i = \frac{\partial E}{\partial x_i} = \frac{\partial E}{\partial y_i} x \frac{\partial y_i}{\partial x_i} = EA_i y_i (1 - y_i) \quad (2.8)$$

3. Calcula los cambios de error con los pesos respecto a la conexión de los cambios de la unidad de salida. Esta cantidad EW es la respuesta del paso 2 multiplicada por el nivel de actividad de donde procede la conexión como se muestra en la ecuación 2.9.

$$EW_{ij} = \frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial x_j} x \frac{\partial x_j}{\partial W_{ij}} = El_i y_i \quad (2.9)$$

4. Se calcula la actividad de los cambios de error de la unidad a la entrada de la capa anterior. Este paso permite que la propagación se aplique a toda la red multicapas. Cuando la actividad de la entrada de las capas anteriores afectan la actividad de las unidades de salida al cual están conectadas. Por lo que para calcular el efecto del error total, se tienen que sumar todos los efectos separados en las unidades de salida. Pero cada uno de los efectos es un simple cálculo. Esto es la respuesta en el paso 2 es multiplicada por el peso de la conexión a la unidad de salida como se muestra en la ecuación 2.10.

$$EA_i = \frac{\partial E}{\partial y_i} = \sum_j \frac{\partial E}{\partial x_j} x \frac{\partial x_j}{\partial y_i} = \sum_j El_j W_{ij} \quad (2.10)$$

Usando los pasos 2 y 4, se puede convertir los EAs de una capa de la unidad dentro de las EAs de las unidades anteriores. Este procedimiento es repetido para obtener las EAs para las capas anteriores deseadas. Una vez que se conoce la EA de la unidad, se pueden usar los pasos 2 y 3 para calcular los EWs de todas sus conexiones.

Las redes neuronales utilizadas se basan en el modelo de back-propagation y es el mismo que utilizan los tres módulos de la red neuronal ensemble, cada una de ellas tiene dos capas ocultas. La base de datos utilizada contiene 10 personas con 8 muestras por persona donde k corresponde al número de personas y m corresponde al número de muestras. La red neuronal ensemble propuesta tiene dos capas ocultas por cada

modulo, en la primer capas tiene $2*(k + m)$ neuronas y en la segunda capa oculta tiene $(k + m)$ neuronas, y k neuronas de salida.

La entrada de las huellas digitales proviene de archivos individuales 1.tif, 2.tif, 8.tif, localizados en 10 directorios nombrados del 1 al 10. Cada archivo .TIF es de 300x300 pixeles, las imágenes tienen 256 niveles de tonos de gris, 0 para el tono de negro y 255 para el tono de blanco, se redimensiono la imagen a 100x100 pixeles y se mapeo dentro de un vector característico que entra a cada uno de los módulos de la red neuronal ensemble como se muestra en la figura 2.4.

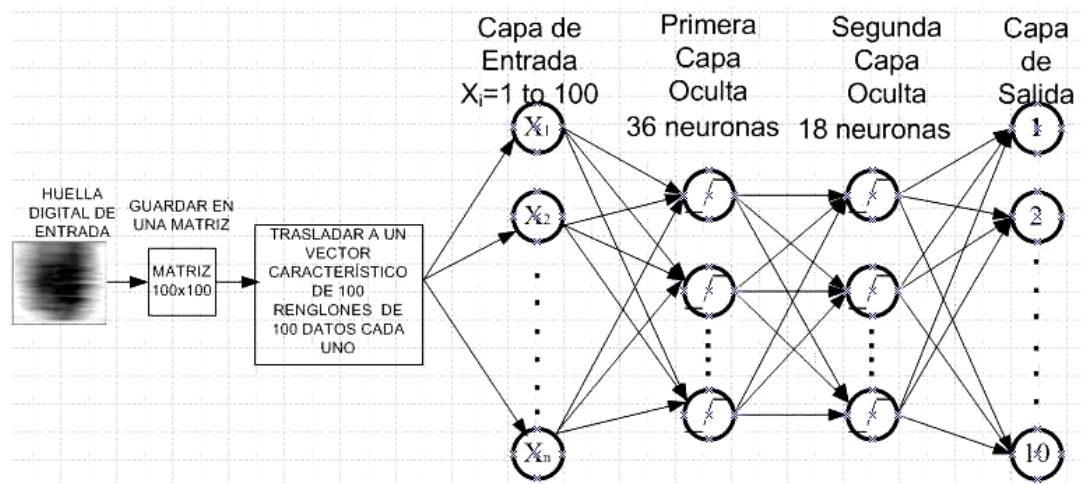
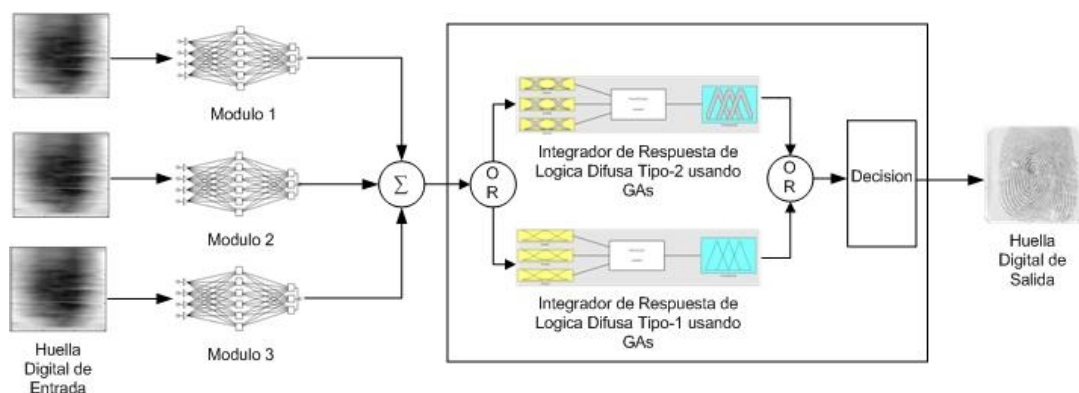


Figura 2.4 Modelo para cada uno de los tres módulos de la red neuronal ensemble.

El modelo completo propuesto del sistema híbrido inteligente para reconocimiento de patrones se muestra a continuación en la figura 2.5.



2.1.1 ENN-IT2FIS (Utilizando Algoritmos Genéticos para Optimizar las funciones de Membresía).

La optimización es el proceso de hacer las cosas mejor, un científico trabaja una nueva idea y optimiza lo mejor de esa idea. La optimización consiste en intentar variaciones del concepto inicial y utilizar la información resultante para mejorar la idea. Una computadora es una herramienta perfecta para la optimización tanto como la idea o la variable pueda entrar en un formato electrónico, es decir alimentar la computadora de algunos datos y obtener la solución a la salida. Esta es la única solución, a veces no lo es, ¿es esta la mejor solución?, esa es una gran pregunta. La optimización es una herramienta matemática con la que se cuenta para conseguir esas respuestas.

El algoritmo genético (GA) es una técnica de optimización y de búsqueda basada en los principios de selección genéticos y naturales [32]. El GA permite una población integrada por muchos individuos para evolucionar bajo una selección de reglas especificada por el estado de los individuos con una mejor aptitud. El método fue desarrollado por John Holland (1975) entre los años 60's y 70's, que se popularizo entre sus estudiantes, David Goldberg quien fue capaz de resolver la dificultad de un problema de control, algunas de las ventajas del GA incluyen:

- Optimizar variables continuas o discretas.
- No requiere información derivada.
- Simultáneamente busca dentro de un muestreo amplio de la superficie de costo.
- Trata con grandes números o variables.
- Es muy apropiado para procesamiento en paralelo.
- Optimiza variables con superficies de costo extremadamente complejas (pueden saltar afuera de un mínimo local).
- Proporcionan una lista de variables óptimas, no solamente una solución.

- Pueden codificar las variables ya que la optimización es hecha con las variables codificadas.

- Trabaja con datos generados numéricamente, datos experimentales o funciones analíticas.

2.1.1.1 Diseño del algoritmo genético para optimización de las funciones de membresía del integrador de respuesta de lógica difusa

El GA trata con la codificación problema usando cromosomas para optimizar las funciones de membresía de los sistemas de lógica difusa tipo-1 y tipo-2, usados como un método de integración de respuestas en las redes neuronales ensemble. Las características del GA son las siguientes:

- 1 Los GAs son procedimientos de búsquedas en paralelo que se pueden implementar dentro de computadoras que manejan procesamiento paralelo para que masivamente a altas velocidades realicen sus operaciones.
- 2 Los GAs se aplican a ambos problemas de optimización continuos y discretos.
- 3 Los GAs son estocásticos y con una menor probabilidad ser atrapados dentro de un mínimo local, lo cual inevitablemente está presente en cualquier aplicación práctica de optimización.
- 4 Los GAs tienen flexibilidad en la identificación de la estructura y los parámetros de modelos complejos tales como las redes neuronales y los sistemas de lógica difusa.

En cada generación, el GA construye una nueva población usando los operadores genéticos de cruce y mutación, esto es los miembros con los valores con más alta aptitud y son los que sobreviven para participar en las próximas operaciones de cruce. Después de un número de generaciones, la población contiene los miembros con los valores de más alta aptitud esto es análogo a los modelos de Darwin de la evolución por la mutación aleatoria y selección natural. Los GAs son algunas veces referidos como

métodos de optimización basados en poblaciones, mejoran el funcionamiento de las poblaciones, esto lo realizan actualizando todas las poblaciones y no solamente los individuos. En el presente trabajo de Tesis el GA se utilizó para optimizar las funciones de membresía del integrador de respuesta de lógica difusa tipo-1 y tipo-2 en una red neuronal ensemble [33,34]. El GA trata con la codificación del problema usando un cromosoma para optimizar las funciones de membresía de los sistemas de lógica difusa tipo-1.

2.1.1.2 Estructura del Cromosoma para lógica difusa tipo-1

Los primeros 9 bits representan la activación de las funciones de membresía para el módulo 1, cada uno de los módulos tiene tres variables lingüísticas, las cuales son Activación Baja, Activación Media, y Activación Alta, los bits del 10 al 18 son para el módulo 2 y del 19 al 27 son para el módulo 3 y los bits del 28 al 36 representan la variable de salida del módulo ganador para las tres funciones de membresía ganador módulo1, ganador módulo2 y ganador módulo3. Las funciones de membresía para optimizar son de tipo triangular para las variables de entrada y de salida.

La estructura del cromosoma usado para optimizar las funciones de membresía se muestra a continuación en la figura 2.6.

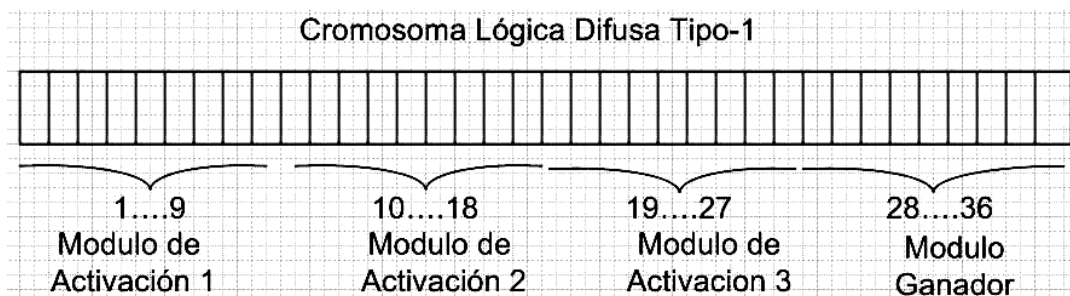


Figura 2.6 Estructura del cromosoma para optimizar las funciones de membresía del integrador de respuesta de lógica difusa tipo-1.

Los primeros 12 bits representan la activación de las funciones de membresía, cada una tiene tres variables lingüísticas, las cuales son activación baja, activación media y activación alta. Los bits del 13 al 24 son para el modulo 2, los bits del 25 al 36 son para el modulo 3, y los bits del 37 al 48 son representan la variable de salida del modulo ganador, ganador modulo1, ganador modulo2 y ganador modulo. Las funciones de membresía para optimizar son de tipo gaussiano para las variables de entrada y de salida. La estructura del cromosoma usado para optimizar las funciones de membresía del integrador de respuestas de lógica difusa tipo-2 se muestra a continuación en la figura 2.7.

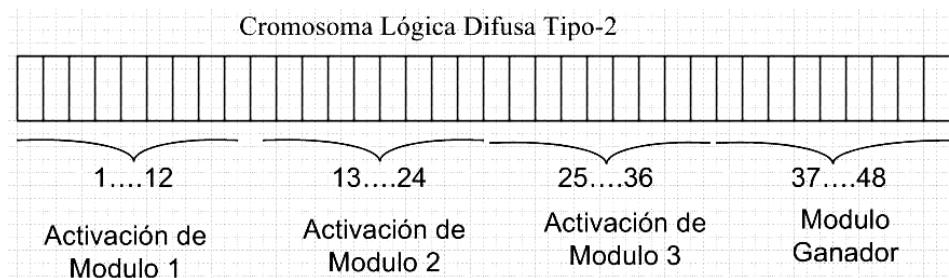


Figura 2.7 Estructura del cromosoma para optimizar las funciones de membresía del integrador de respuesta de lógica difusa tipo-2.

2.1.2 ENN-IT2FIS (Utilizando Algoritmos Genéticos para Optimizar las Reglas Difusas).

El GA también se utilizó en el presente trabajo de tesis para optimizar el número de reglas del integrador de respuesta del sistema difuso de lógica difusa tipo-1 y tipo-2 de en una red neuronal ensemble [36]. El GA trata la codificación del problema usando un cromosoma para optimizar las reglas de los sistemas de lógica difusa tipo-1 y tipo-2 [37].

2.1.2.1 Estructura del cromosoma para el sistema de lógica difusa tipo-1, los primeros 27 bits representan las reglas difusas del sistema de inferencia difuso. El

funciones de membresía y una salida con tres funciones de membresía. Las funciones de membresía de salida son Activación baja, Activación Media y Activación Alta. Los bits ubicados de la posición 28 a la 39 representan la activación de la primera regla difusa para cada una de las tres entradas en la cual cada una cuenta con tres funciones de membresía, de manera similar los bits de la posición 40 a la 51 la activación para la segunda regla, y los últimos bits de la 340 a la 351 representan la activación de la ultima regla difusa (regla 27).

La estructura del cromosoma usado para optimizar las reglas del sistema de lógica difusa tipo-1 se muestra en la figura 2.8.

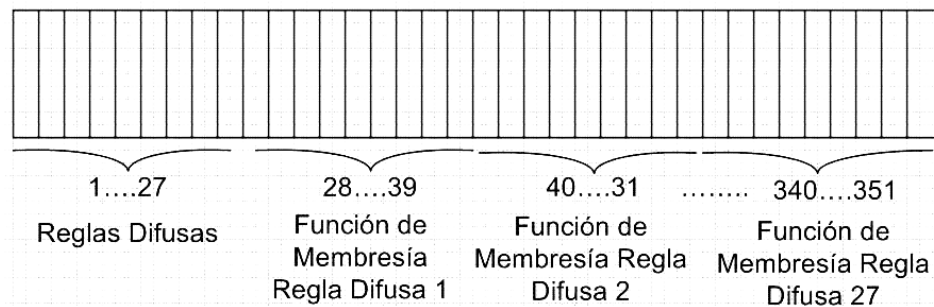


Figura 2.8 Estructura del cromosoma para optimizar las reglas difusas del sistema de lógica difusa tipo-1.

La estructura del cromosoma usada para optimizar el número de reglas difusas del integrador de respuesta de lógica difusa tipo-2, se muestra en la siguiente figura 2.9.

2.1.2.2 Estructura del cromosoma para el sistema de lógica difusa tipo-2, los primeros 27 bits representan las reglas difusas del sistema de inferencia difuso. El sistema de inferencia difuso tiene tres variables lingüísticas, y cada una con seis funciones de membresía (Alta y Baja) y una salida con seis funciones de membresía (Alta y Baja). Las funciones de membresía de salida son Activación baja, Activación Media y Activación Alta. Los bits ubicados de la posición 28 a la 51 representan la activación de la primera regla difusa para cada una de las tres entradas en la cual cada

a la 75 representan la activación para la segunda regla, y los últimos bits de la 625 a la 675 representan la activación de la ultima regla difusa (regla 27).

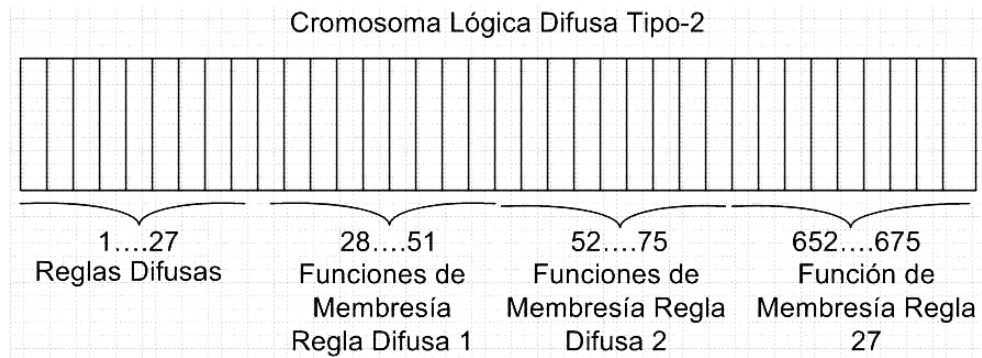


Figura 2.9 Estructura del cromosoma para optimizar las reglas difusas del sistema de lógica difusa tipo-2.

CAPÍTULO 3. SIMULACION DE RESULTADOS

3.1 CONDICIONES GENERALES DE LOS EXPERIMENTOS

Para los experimentos ejecutados en esta investigación, se utilizó la base de datos de huellas digitales de la Competición de Verificación de Huellas FCV2000 [37,38]. El tamaño de la imagen es de 300 píxeles de ancho por 300 píxeles de alto con una resolución de 500 píxeles por pulgada cuadrada (ppi), y con una escala de tonos de grises (0 negro, y 255 blanco). Las huellas digitales fueron adquiridas con un sensor óptico de bajo costo, se tomaron muestras de cuatro dedos por persona, dos de cada mano (dedo índice y dedo medio). La base de datos es de 10 personas, con 8 muestras por persona (4 dedos, una muestra por cada dedo), para un total de 80 huellas digitales. Las huellas fueron adquiridas fueron analizadas manualmente para asegurar que la máxima rotación este aproximadamente en un rango de -15° a $+15^\circ$, y cada par de impresiones del mismo dedo no tienen ninguna área que se traslape. Las muestras de las huellas de la base de datos FCV2000 se muestran a continuación en la figura 3.1.

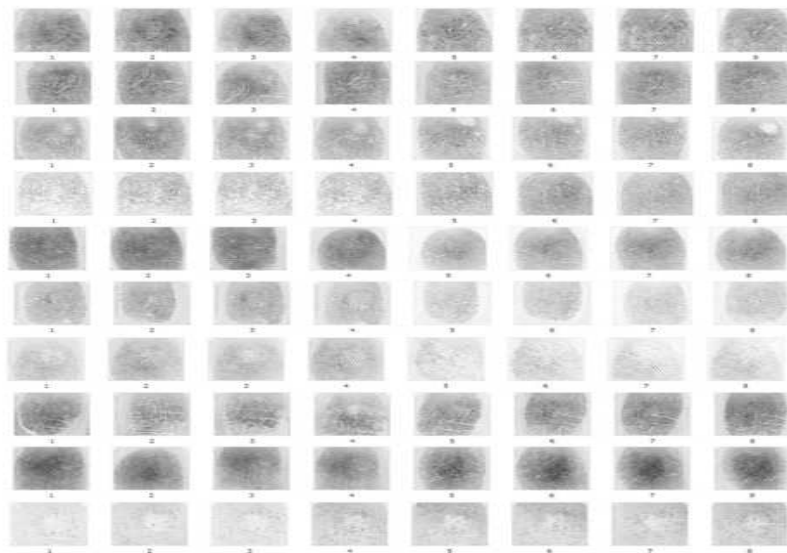


Figura 3.1 Base de datos de huellas digitales utilizada para los experimentos FCV2000.

3.2 EXPERIMENTOS DE ENN-FIS (Utilizando Algoritmos Genéticos para Optimizar las funciones de Membresía).

En la década pasada, los sistemas difusos han desplazado la tecnología convencional en diferentes aplicaciones científicas y de ingeniería, especialmente en reconocimiento de patrones y sistemas de control. Las áreas de aplicación de los sistemas difusos son en, tecnología difusa, en razonamiento aproximado, en tecnología de la información, donde ha dado un soporte a la toma de decisiones de los sistemas expertos con una cantidad limitada de reglas [39]. Para el caso de redes neuronales modulares, el sistema difuso se puede usar como un integrador de respuestas.

3.2.1 Integrador de Respuesta de lógica difusa tipo-1

Se diseño un sistema de inferencia de lógica difusa tipo-1 tipo Mamdani con tres variables lingüísticas, Activación Baja, Activación Media, y Activación Alta, y una variable lingüística de salida Activación Ganadora con tres funciones de membresía. Las tres funciones de membresía utilizadas para la entrada y la salida son de tipo triangular, y su rango de valores es de 0 a 1. A continuación se muestran algunas de las reglas utilizadas para el sistema de inferencia de lógica difusa.

If (Module1 is ActMod1Low) and (Module2 is ActMod2Medium) and (Module3 is ActMod3Medium) then (Winner Module is Module3)

If (Module1 is ActMod1High) and (Module2 is ActMod2Low) and (Module3 is ActMod3High) then (Winner Module is Module1)

If (Module1 is ActMod1Medium) and (Module2 is ActMod2High) and (Module3 is ActMod3Low) then (Winner Module is Module2)

If (Module1 is ActMod1Low) and (Module2 is ActMod2Medium) and (Module3 is ActMod3Medium) then (Winner Module is Module2)

En las figuras 3.2 a 3.5 se muestran las funciones de membresía de entrada y de salida, diseñadas usando el editor de lógica difusa de la toolbox de MATLAB [40].

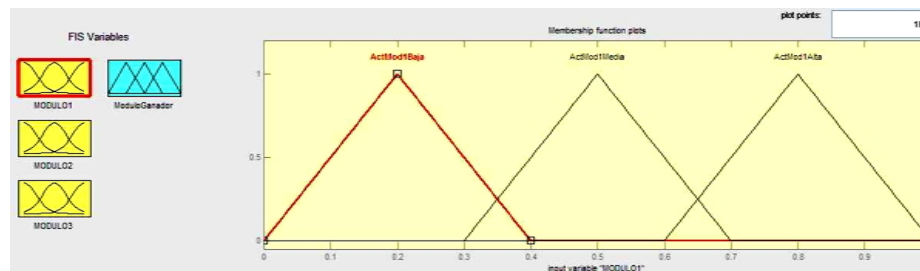


Figura 3.2 Funciones de Membresía Entrada Modulo1.

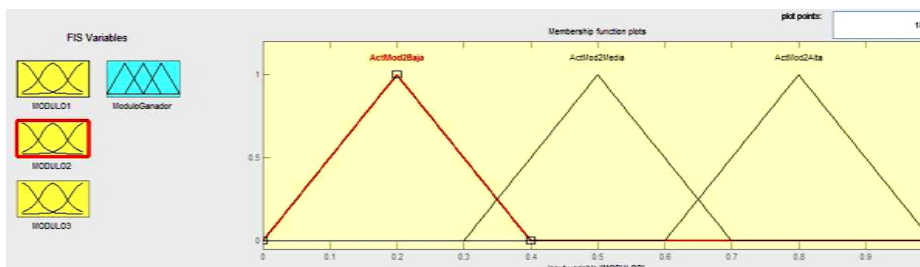


Figura 3.3 Funciones de Membresía Entrada Modulo2.

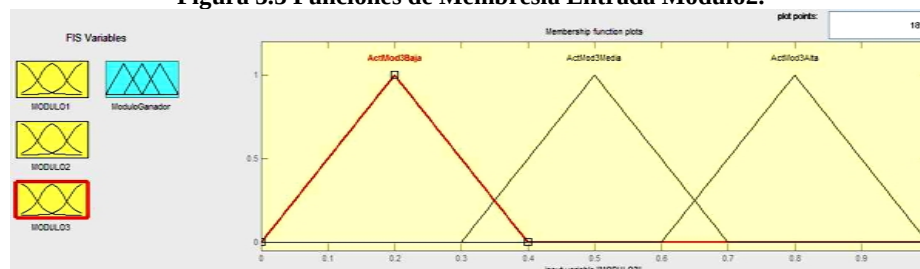


Figura 3.4 Funciones de Membresía Entrada Modulo3.

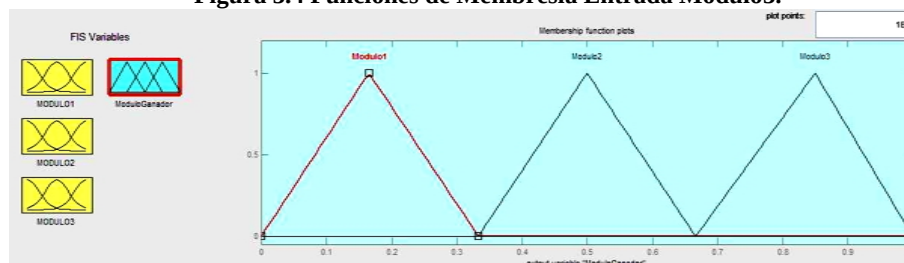


Figura 3.5 Funciones de Membresía de la Variable de Salida.

3.2.2 Integrador de Respuesta de lógica difusa tipo-1(Utilizando Algoritmos Genéticos para optimizar sus funciones de membresía)

Se utilizaron algoritmos genéticos para optimizar las funciones de membresía tipo triangular del sistema de lógica difusa tipo-1, los parámetros del GA son los siguientes:

razón de mutación del 10%, método de selección de la ruleta, los criterios utilizados para detener al Algoritmo son un máximo de 100 generaciones o si el error entre el sistema de lógica difusa base(mencionado anteriormente) y el sistema difuso de lógica difusa tipo-1 obtenido con el algoritmo genético, en la figura 3.6 se muestra la convergencia del algoritmo genético planteado.

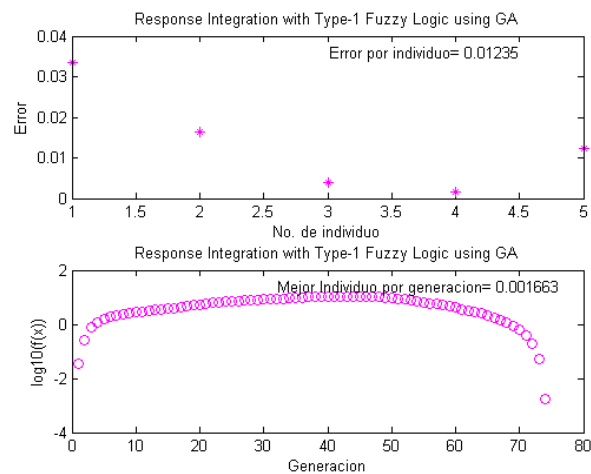


Figura 3.6. Resultados de la convergencia del Algoritmo Genético para el Integrador de Respuesta de lógica difusa tipo-1.

En las figuras 3.7 a 3.10 se muestran las funciones de membresía obtenidas con algoritmos genéticos usando el editor de lógica difusa de la toolbox de MATLAB [40].

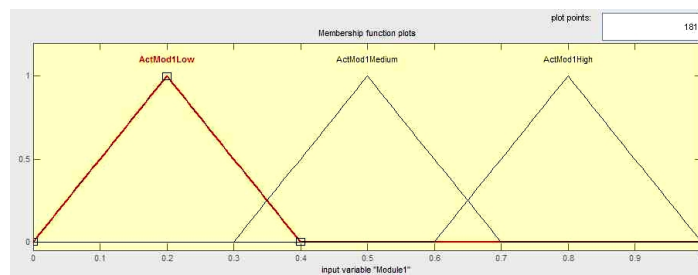


Figura 3.7 Función d Membresía de Entrada Modulo 1 usando el GA.

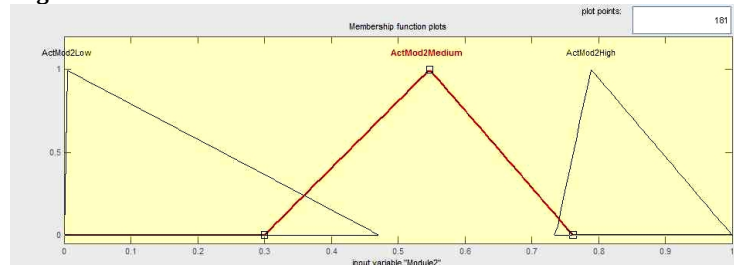


Figura 3.8 Funciones de Membresía de Entrada Modulo 2.

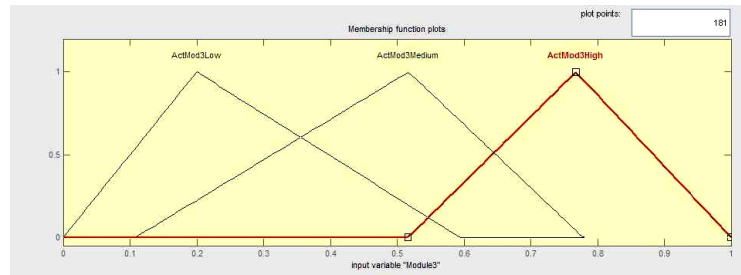


Figura 3.9 Funciones de Membresía de Entrada Modulo 3.

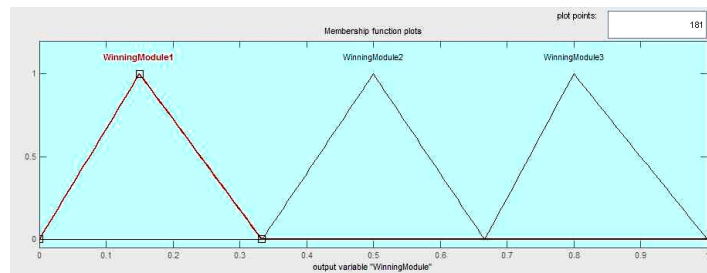


Figura 3.10 Funciones de Membresía de Salida.

3.2.3 Integrador de Respuesta con lógica difusa tipo-2

En la misma forma que se planteó el diseño del integrador de respuesta de lógica difusa tipo-1, para el integrador de respuesta de lógica difusa tipo-2, también se utilizaron tres variables de entrada y una de salida. Tres funciones de membresía por cada variable de entrada, Activación Baja, Activación Media, y Activación Alta, y una variable de salida Activación Ganadora con tres funciones de membresía. Se utilizaron funciones de membresía tipo Gaussianas porque fueron las que dieron una mejor respuesta, manejadas dentro de un rango de valores de 0 a 1. En las figuras 3.11 a la 3.14 se muestran las funciones de membresía usando el editor de lógica difusa del toolbox IT2DIFUSO, el cual fue desarrollado por nuestro grupo de investigación [41,42].

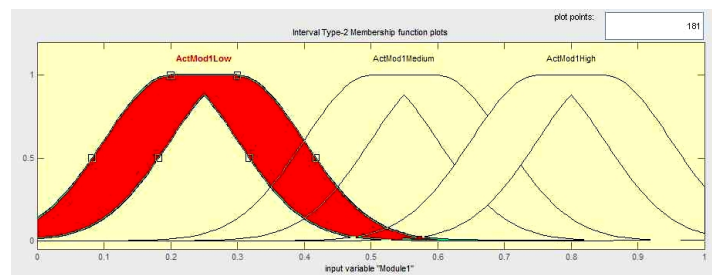


Figura 3.11 Funciones de Membresía tipo-2 Entrada Modulo1.

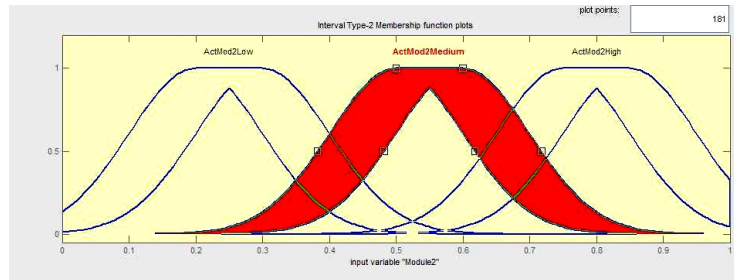


Figura 3.12 Funciones de Membresía tipo-2 Entrada Modulo2.

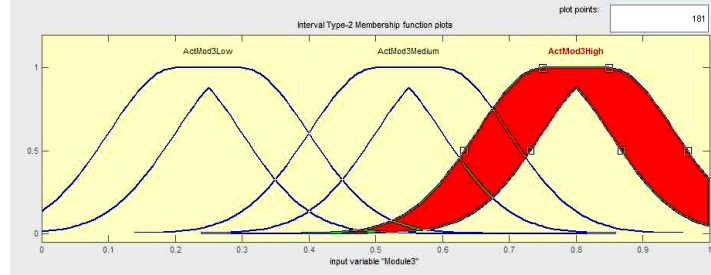


Figura 3.13 Funciones de Membresía tipo-2 Entrada Modulo3.

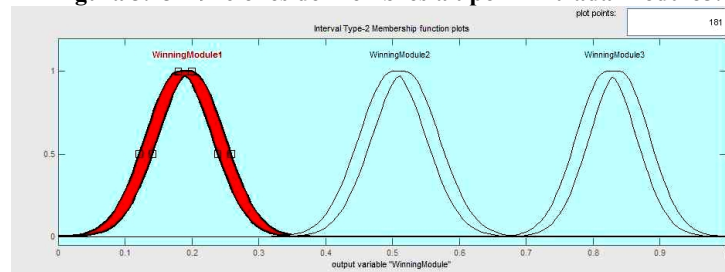


Figura 3.14 Funciones de Membresía tipo-2 de Salida.

3.2.4 Integrador de Respuesta de lógica difusa tipo-2 usando Algoritmos Utilizando Algoritmos Genéticos para optimizar sus funciones de membresía

Se utilizaron algoritmos genéticos para optimizar las funciones de membresía tipo triangular del sistema de lógica difusa tipo-1, los parámetros del GA son los siguientes: una población de 10 individuos razón de cruce de una razón de cruce del 15%, una razón de mutación del 10%, método de selección de la ruleta, los criterios utilizados para detener al Algoritmo son un máximo de 100 generaciones o si el error entre el sistema de lógica difusa base(mencionado anteriormente) y el sistema difuso de lógica difuso tipo-2 obtenido con el algoritmo genético, en la figura 3.15 se muestra la convergencia del algoritmo genético planteado.

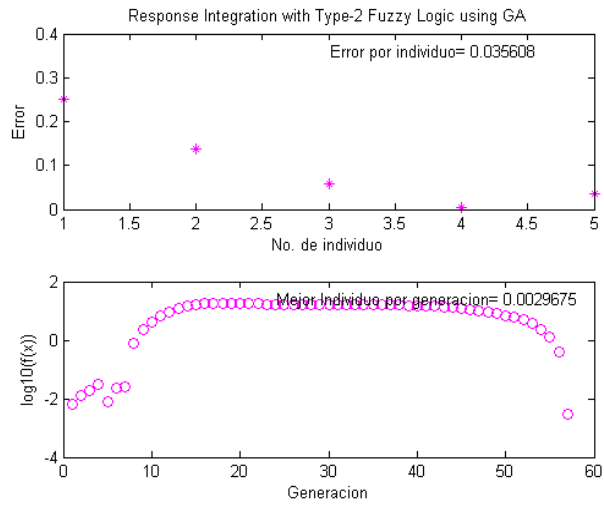


Figura 3.15 Resultados de la convergencia del Algoritmo Genético para el Integrador de Respuesta de lógica difusa tipo-2.

En las figuras 3.16 a 3.19 se muestran las funciones de membresía obtenidas con algoritmos genéticos usando el editor de lógica difusa de la toolbox de MATLAB [41,42].

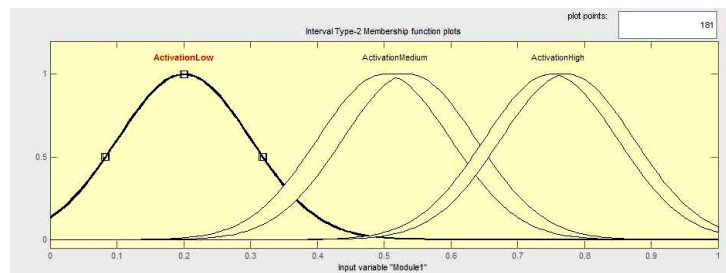


Figura 3.16 Funciones de Membresía tipo-2 Entrada Modulo1 usando GA.

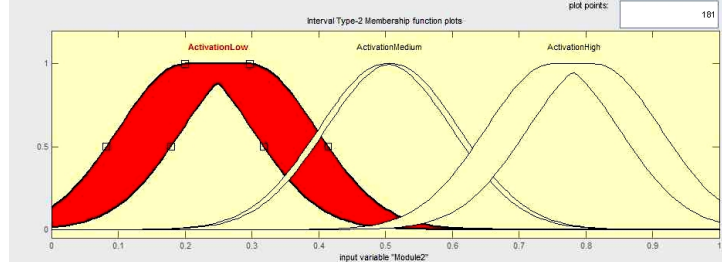
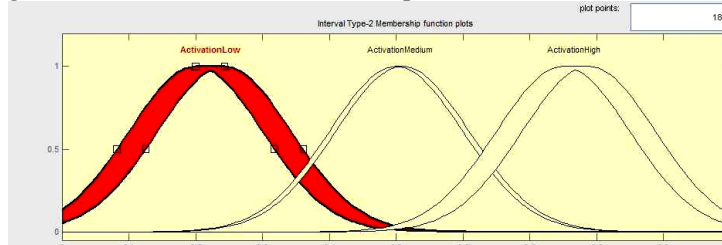


Figura 3.17 Funciones de Membresía tipo-2 Entrada Modulo2 usando GA.



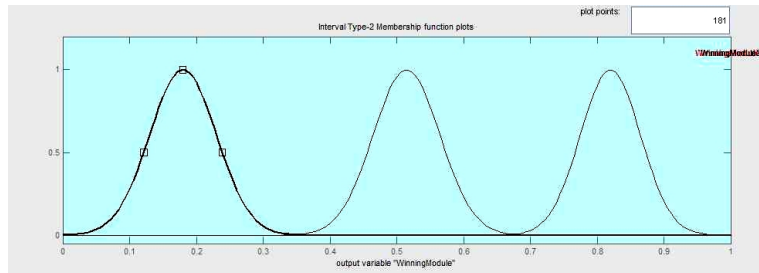


Figura 3.19 Funciones de Membresía tipo-2 Entrada Modulo1 usando GA.

3.2.5 Simulación de Resultados con Ruido borroso con movimiento lineal usando GAs (Optimizar Funciones de Membresía)

Una vez que la Red neuronal ensemble es entrenada, el sistema de inferencia de lógica difusa integra las salidas de los tres módulos, se utilizaron las 80 imágenes de la base de datos de huellas digitales aplicándoles diferentes niveles de ruido borroso con movimiento lineal, ambos sistemas de lógica difusa tipo tipo-1 y tipo-2 dan una respuesta para la siguiente etapa de decisión final, y muestra el resultado de la huella si esta fue identificada o no. A continuación en las figuras 3.20 y 3.21 se muestran los resultados experimentales obtenidos con el integrador de respuesta de lógica difusa tipo-1 usando el GA con un nivel de ruido borroso con movimiento lineal de 10 y 50 pixeles de desplazamiento respectivamente.

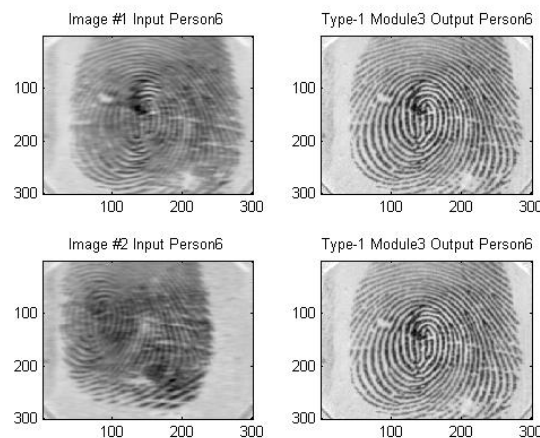


Figura 3.20 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-1 (con un nivel de ruido borroso con movimiento lineal de 10 pixeles de desplazamiento).

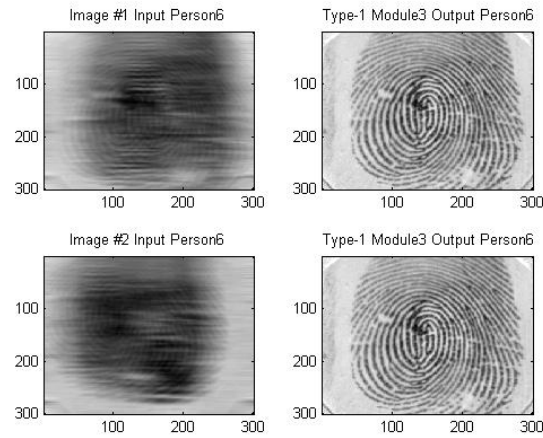


Figura 3.21 Resultados Experimentales de las huellas usando el integrador de respuesta de logica difusa tipo-1 (con un nivel de ruido borroso con movimiento lineal de 50 pixeles de desplazamiento).

A continuación en las figuras 3.22 y 3.23 se muestran los resultados experimentales obtenidos con el integrador de respuesta de lógica difusa tipo-2 usando el GA con un nivel de ruido borroso con movimiento lineal de 10 y 50 pixeles de desplazamiento respectivamente.

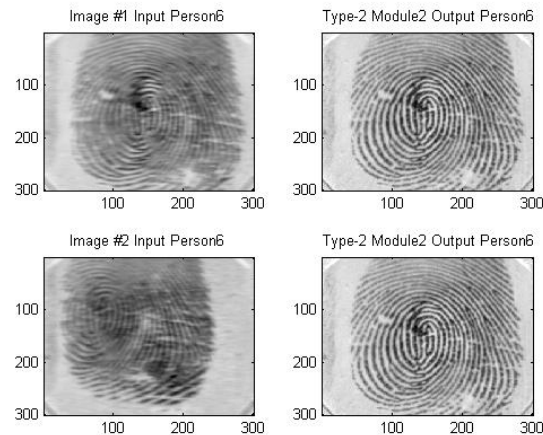


Figura 3.22 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-2 (con un nivel de ruido borroso con movimiento lineal de 10 pixeles de desplazamiento).

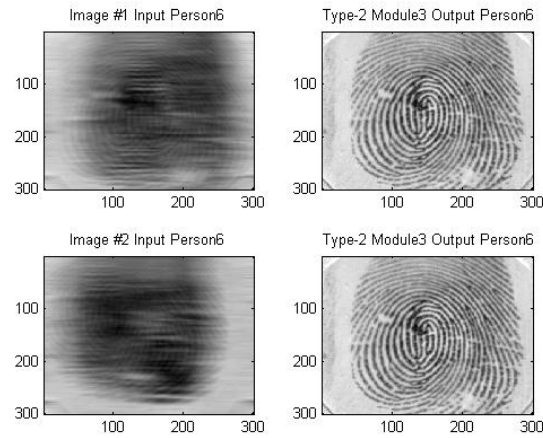


Figura 3.23 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-2 (con un nivel de ruido borroso con movimiento lineal de 50 pixeles de desplazamiento).

3.2.6 Simulación de resultados con ruido borroso con movimiento radial usando GAs (Optimizar Funciones de Membresía)

De la misma forma que se realizó la simulación con el ruido borroso con movimiento lineal, ahora se muestra la simulación con la aplicación de diferentes niveles de ruido borroso con movimiento radial, de igual manera ambos sistemas de lógica difusa tipo tipo-1 y tipo-2 dan una respuesta para la siguiente etapa de decisión final, y muestra el resultado de la huella si esta fue identificada o no. A continuación en las figuras 3.24 y 3.25 se muestran los resultados experimentales obtenidos con el integrador de respuesta de lógica difusa tipo-1 usando el GA con un nivel de ruido borroso con movimiento radial de 10 y 50 pixeles de desplazamiento respectivamente.

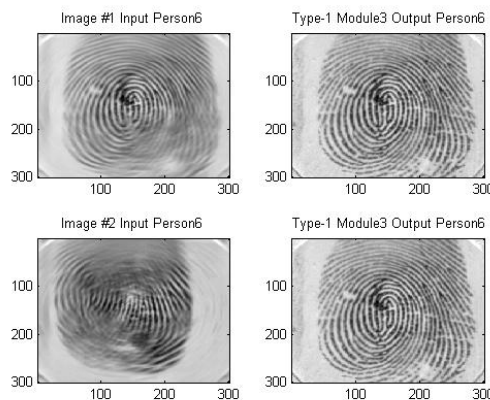


Figura 3.24 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-1 (con un nivel de ruido borroso con movimiento radial de 10 pixeles de desplazamiento).

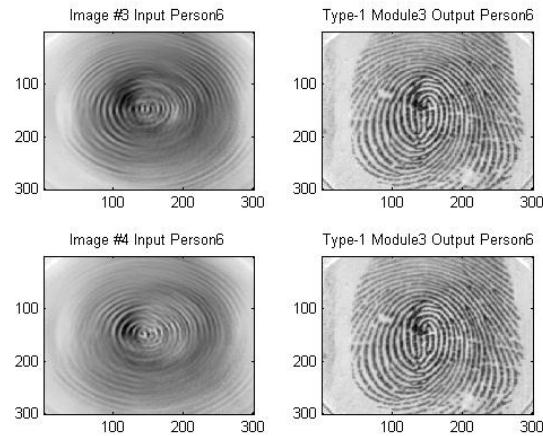


Figura 3.25 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-1 (con un nivel de ruido borroso con movimiento radial de 50 pixeles de desplazamiento).

A continuación en las figuras 3.26 y 3.27 se muestran los resultados experimentales obtenidos con el integrador de respuesta de lógica difusa tipo-2 usando el GA con un nivel de ruido borroso con movimiento radial de 10 y 50 pixeles de desplazamiento respectivamente.

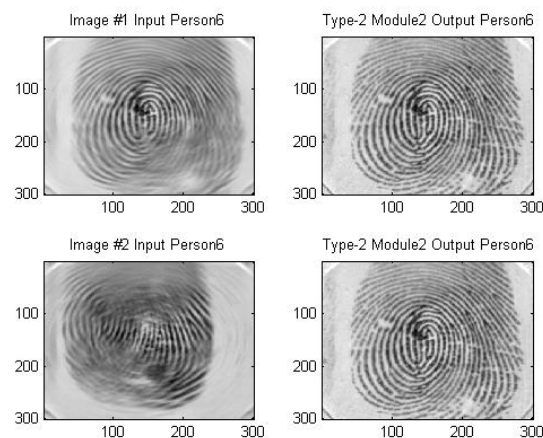


Figura 3.26 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-2 (con un nivel de ruido borroso con movimiento radial de 10 pixeles de desplazamiento).

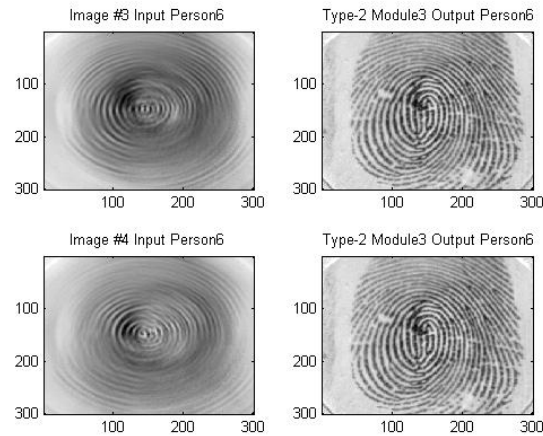


Figura 3.27 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-2 (con un nivel de ruido borroso con movimiento radial de 50 pixeles de desplazamiento).

3.2.7 Simulación de resultados con ruido borroso con movimiento gaussianousando GAs (Optimizar Funciones de Membresía)

De la misma forma que se realizó la simulación con el ruido borroso con movimiento lineal y el ruido borroso con movimiento radial, ahora se muestra la simulación con la aplicación de diferentes niveles de ruido borroso movimiento gaussiano, de igual manera ambos sistemas de lógica difusa tipo tipo-1 y tipo-2 dan una respuesta para la siguiente etapa de decisión final, y muestran el resultado de la huella si esta fue identificada o no. A continuación en las figuras 3.28 y 3.29 se muestran los resultados experimentales obtenidos con el integrador de respuesta de lógica difusa tipo-1 usando el GA con un nivel de ruido borroso con movimiento gaussiano 5 y 10 pixeles de desplazamiento respectivamente.

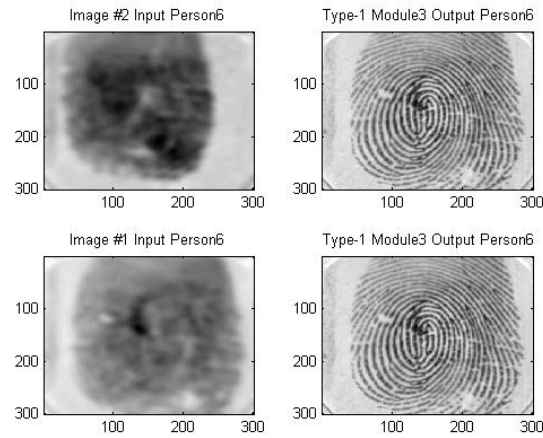


Figura 3.28 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-1 (con un nivel de ruido borroso con movimiento gaussiano 5 pixeles de desplazamiento).

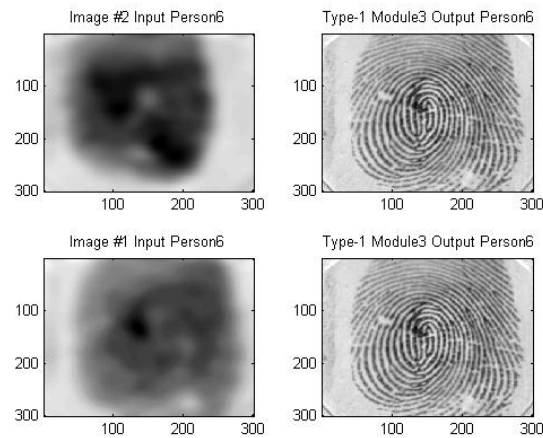


Figura 3.29 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-1 (con un nivel de ruido borroso con movimiento gaussiano 10 pixeles de desplazamiento).

A continuación en las figuras 3.30 y 3.31 se muestran los resultados experimentales obtenidos con el integrador de respuesta de lógica difusa tipo-2 usando el GA con un nivel de ruido borroso con movimiento gaussiano 5 y 10 pixeles de desplazamiento respectivamente

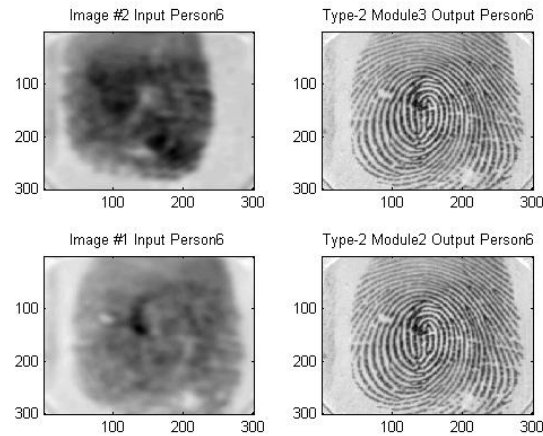


Figura 3.30 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-2 (con un nivel de ruido borroso con movimiento gaussiano 5 pixeles de desplazamiento).

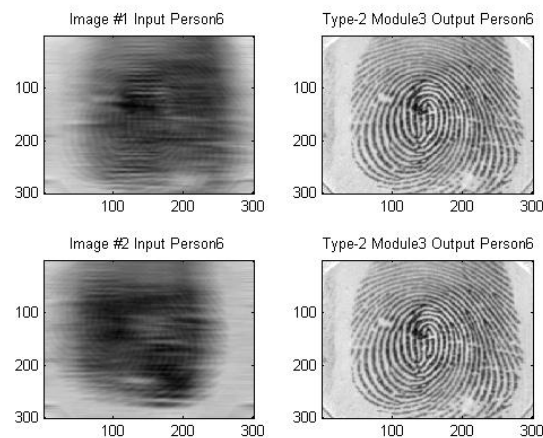


Figura 3.31 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-2 (con un nivel de ruido borroso con movimiento gaussiano 10 pixeles de desplazamiento).

3.3 EXPERIMENTOS DE ENN-FIS (Utilizando Algoritmos Genéticos para Optimizar las Reglas Difusas).

Se utilizaron GAs para optimizar las reglas difusa del integrador de respuesta de lógica difusa tipo-1, el algoritmo genético comienza con 27 reglas, los parámetros del GA utilizados son; población de 10 individuos, razón de cruce 70%, razón de mutación 10%, método de selección ruleta, las condiciones para detener el GA son 100 generaciones máximo o si el error cero entre el sistema difuso de lógica difusa tipo-1 base y el sistema de lógica difusa tipo-1 obtenido con el GA. El mejor resultado obtenido con el GA para

optimizar las reglas del sistema difuso de lógica difusa tipo-1 fueron 20 reglas las cuales se muestran a continuación.

1. If (MODULO1 is ActMod1Baja) and (MODULO2 is ActMod2Baja) and (MODULO3 is ActMod3Media) then (ModuloGanador is Modulo3)
2. If (MODULO1 is ActMod1Baja) and (MODULO2 is ActMod2Baja) and (MODULO3 is ActMod3Alta) then (ModuloGanador is Modulo3)
3. If (MODULO1 is ActMod1Baja) and (MODULO2 is ActMod2Media) and (MODULO3 is ActMod3Baja) then (ModuloGanador is Modulo2)
4. If (MODULO1 is ActMod1Baja) and (MODULO2 is ActMod2Media) and (MODULO3 is ActMod3Media) then (ModuloGanador is Modulo2)
5. If (MODULO1 is ActMod1Baja) and (MODULO2 is ActMod2Media) and (MODULO3 is ActMod3Alta) then (ModuloGanador is Modulo3)
6. If (MODULO1 is ActMod1Baja) and (MODULO2 is ActMod2Alta) and (MODULO3 is ActMod3Baja) then (ModuloGanador is Modulo2)
7. If (MODULO1 is ActMod1Baja) and (MODULO2 is ActMod2Alta) and (MODULO3 is ActMod3Media) then (ModuloGanador is Modulo2)
8. If (MODULO1 is ActMod1Baja) and (MODULO2 is ActMod2Alta) and (MODULO3 is ActMod3Alta) then (ModuloGanador is Modulo2)
9. If (MODULO1 is ActMod1Media) and (MODULO2 is ActMod2Baja) and (MODULO3 is ActMod3Alta) then (ModuloGanador is Modulo3)
10. If (MODULO1 is ActMod1Alta) and (MODULO2 is ActMod2Baja) and (MODULO3 is ActMod3Media) then (ModuloGanador is Modulo1)
11. If (MODULO1 is ActMod1Alta) and (MODULO2 is ActMod2Baja) and (MODULO3 is ActMod3Alta) then (ModuloGanador is Modulo3)
12. If (MODULO1 is ActMod1Alta) and (MODULO2 is ActMod2Media) and (MODULO3 is ActMod3Media) then (ModuloGanador is Modulo1)
13. If (MODULO1 is ActMod1Alta) and (MODULO2 is ActMod2Media) and (MODULO3 is ActMod3Alta) then (ModuloGanador is Modulo3)
14. If (MODULO1 is ActMod1Alta) and (MODULO2 is ActMod2Alta) and (MODULO3 is ActMod3Baja) then (ModuloGanador is Modulo2)
15. If (MODULO1 is ActMod1Alta) and (MODULO2 is ActMod2Alta) and (MODULO3 is ActMod3Media) then (ModuloGanador is Modulo2)
16. If (MODULO1 is ActMod1Alta) and (MODULO2 is ActMod2Baja) and (MODULO3 is ActMod3Alta) then (ModuloGanador is Modulo3)
17. If (MODULO1 is ActMod1Alta) and (MODULO2 is ActMod2Alta) and (MODULO3 is ActMod3Alta) then (ModuloGanador is Modulo3)
18. If (MODULO1 is ActMod1Media) and (MODULO2 is ActMod2Media) and (MODULO3 is ActMod3Alta) then (ModuloGanador is Modulo3)
19. If (MODULO1 is ActMod1Media) and (MODULO2 is ActMod2Alta) and (MODULO3 is ActMod3Media) then (ModuloGanador is Modulo2)
20. If (MODULO1 is ActMod1Media) and (MODULO2 is ActMod2Alta) and (MODULO3 is ActMod3Alta) then (ModuloGanador is Modulo3)

En la figura 3.32 se muestra la convergencia del GA para optimizar reglas difusas del sistema de lógica difusa tipo-1 usando el editor de MATLAB [42].

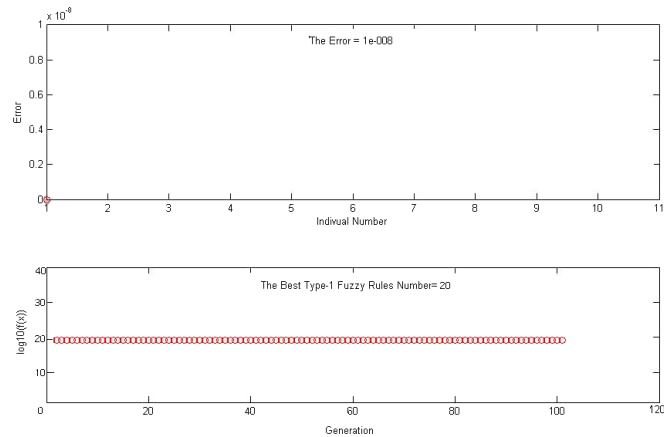


Figura 3.32 Resultados de la convergencia del Algoritmo Genético para optimizar las reglas del Integrador de Respuesta de lógica difusa tipo-1.

3.3.1 Simulación de Resultados con Ruido borroso con movimiento lineal usando GAs (Optimizar Reglas Difusas)

A continuación en las figuras 3.33 y 3.34 se muestran los resultados experimentales obtenidos con el integrador de respuesta de lógica difusa tipo-1 usando el GA con un nivel de ruido borroso con movimiento lineal de 10 y 50 pixeles de desplazamiento respectivamente.

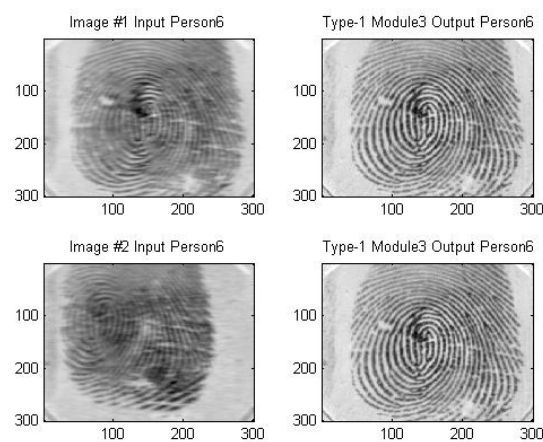


Figura 3.33 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-1 (con un nivel de ruido borroso con movimiento lineal de 10 pixeles de desplazamiento).

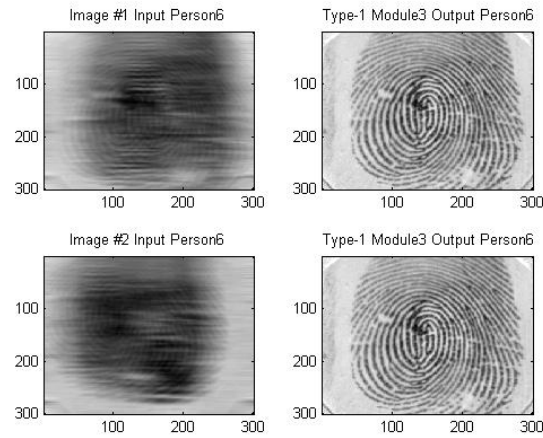


Figura 3.34 Resultados Experimentales de las huellas usando el integrador de respuesta de logica difusa tipo-1 (con un nivel de ruido borroso con movimiento lineal de 50 pixeles de desplazamiento).

A continuación en las figuras 3.35 y 3.36 se muestran los resultados experimentales obtenidos con el integrador de respuesta de lógica difusa tipo-2 usando el GA con un nivel de ruido borroso con movimiento lineal de 10 y 50 pixeles de desplazamiento respectivamente.

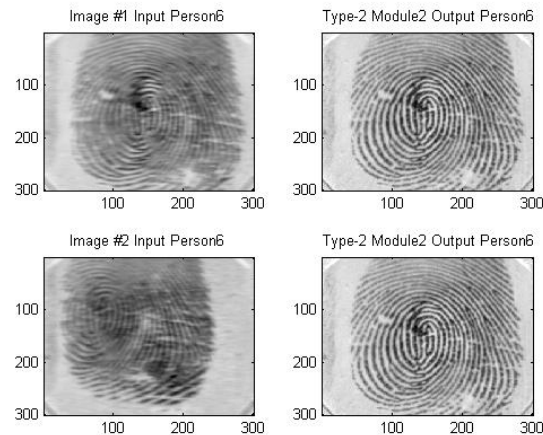


Figura 3.35 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-2 (con un nivel de ruido borroso con movimiento lineal de 10 pixeles de desplazamiento).

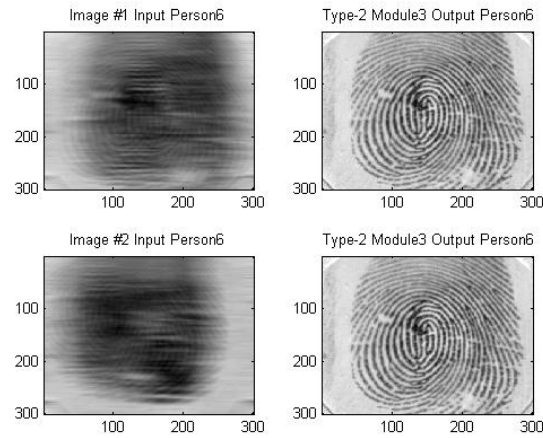


Figura 3.36 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-2 (con un nivel de ruido borroso con movimiento lineal de 50 pixeles de desplazamiento).

3.3.2 Simulación de resultados con ruido borroso con movimiento radial usando GAs (Optimizar Reglas Difusas)

De la misma forma que se realizó la simulación con el ruido borroso con movimiento lineal, ahora se muestra la simulación con la aplicación de diferentes niveles de ruido borroso con movimiento radial, de igual manera ambos sistemas de lógica difusa tipo tipo-1 y tipo-2 dan una respuesta para la siguiente etapa de decisión final, y muestra el resultado de la huella si esta fue identificada o no. A continuación en las figuras 3.37 y 3.38 se muestran los resultados experimentales obtenidos con el integrador de respuesta de lógica difusa tipo-1 usando el GA con un nivel de ruido borroso con movimiento radial de 10 y 50 pixeles de desplazamiento respectivamente.

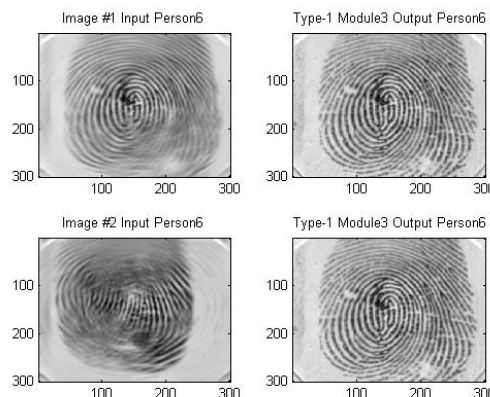


Figura 3.37 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-1 (con un nivel de ruido borroso con movimiento radial de 10 pixeles de desplazamiento).

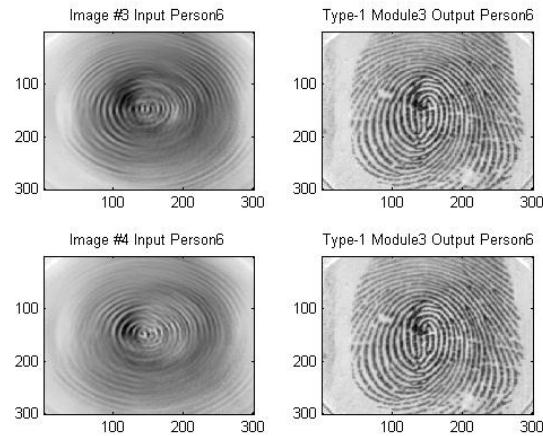


Figura 3.38 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-1 (con un nivel de ruido borroso con movimiento radial de 50 pixeles de desplazamiento).

A continuación en las figuras 3.39 y 3.40 se muestran los resultados experimentales obtenidos con el integrador de respuesta de lógica difusa tipo-2 usando el GA con un nivel de ruido borroso con movimiento radial de 10 y 50 pixeles de desplazamiento respectivamente.

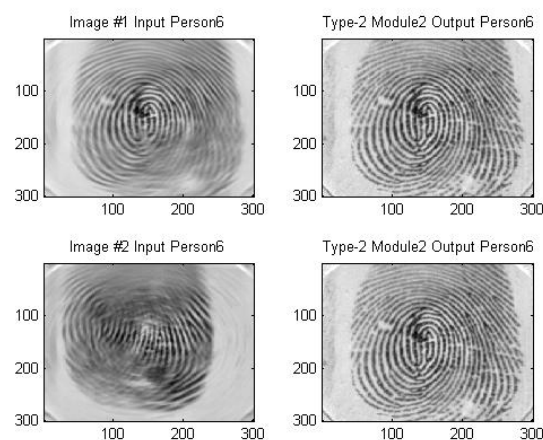


Figura 3.39 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-2 (con un nivel de ruido borroso con movimiento radial de 10 pixeles de desplazamiento).

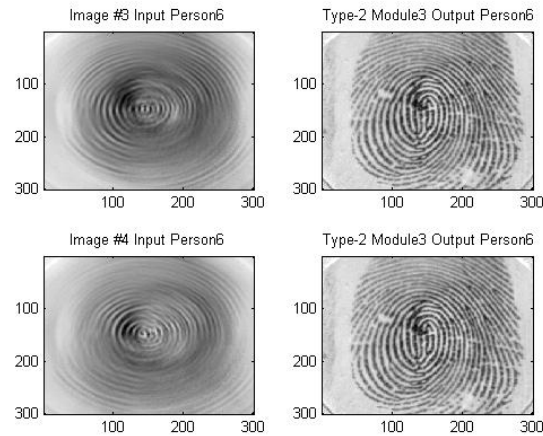


Figura 3.40 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-2 (con un nivel de ruido borroso con movimiento radial de 50 pixeles de desplazamiento).

3.3.3 Simulación de resultados con ruido borroso con movimiento gaussianousando GAs (Optimizar Reglas Difusas)

De la misma forma que se realizó la simulación con el ruido borroso con movimiento lineal y el ruido borroso con movimiento radial, ahora se muestra la simulación con la aplicación de diferentes niveles de ruido borroso movimiento gaussiano, de igual manera ambos sistemas de lógica difusa tipo tipo-1 y tipo-2 dan una respuesta para la siguiente etapa de decisión final, y muestran el resultado de la huella si esta fue identificada o no. A continuación en las figuras 3.41 y 3.42 se muestran los resultados experimentales obtenidos con el integrador de respuesta de lógica difusa tipo-1 usando el GA con un nivel de ruido borroso con movimiento gaussiano 5 y 10 pixeles de desplazamiento respectivamente.

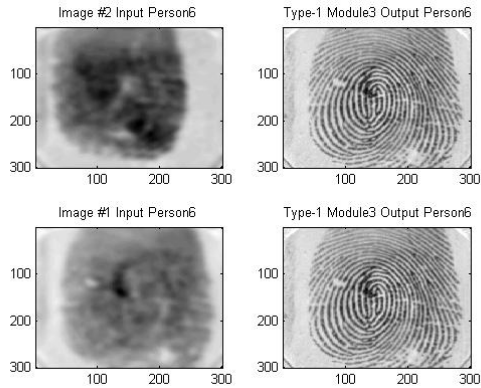


Figura 3.41 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-1 (con un nivel de ruido borroso con movimiento gaussiano 5 pixeles de desplazamiento).

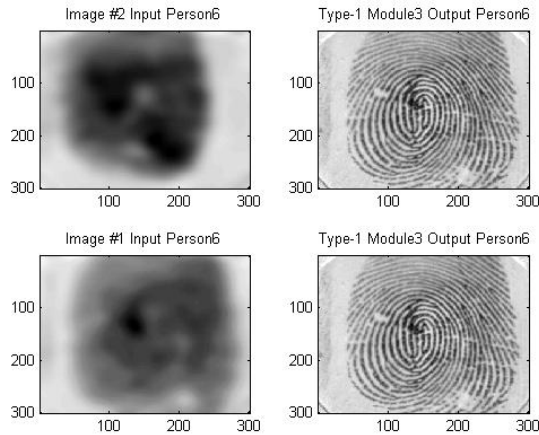


Figura 3.42 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-1 (con un nivel de ruido borroso con movimiento gaussiano de 10 pixeles de desplazamiento).

A continuación en las figuras 3.43 y 3.44 se muestran los resultados experimentales obtenidos con el integrador de respuesta de lógica difusa tipo-2 usando el GA con un nivel de ruido borroso movimiento gaussiano 5 y 10 pixeles de desplazamiento respectivamente.

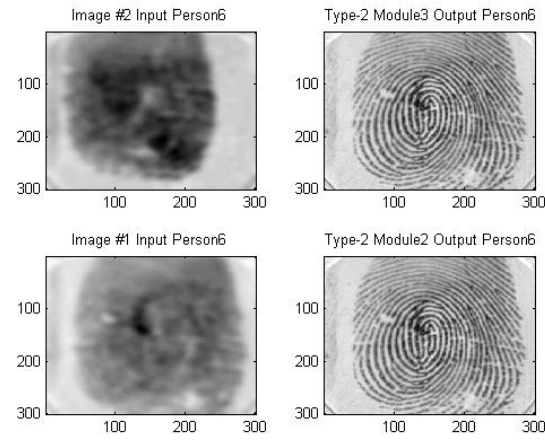


Figura 3.43 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-2 (con un nivel de ruido borroso con movimiento gaussiano 5 pixeles de desplazamiento).

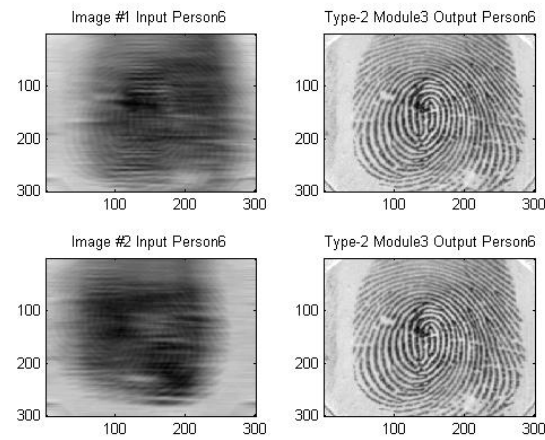


Figura 3.44 Resultados Experimentales de las huellas usando el integrador de respuesta de lógica difusa tipo-2 (con un nivel de ruido borroso con movimiento gaussiano 10 pixeles de desplazamiento).

CAPÍTULO 4. ANALISIS Y COMPARACION DE RESULTADOS

4.1 ANALISIS DE RESULTADOS OBTENIDOS CON FIS-ENN (Utilizando Algoritmos Genéticos para Optimizar las funciones de Membresía).

Después de realizar las pruebas de simulación con la base de datos de huella digitales FCV2000 para el integrador de respuesta de lógica difusa tipo-1 usando GAs para optimizar sus funciones de membresía con tres tipos de ruido, ruido borroso con movimiento lineal, ruido borroso con movimiento radial y ruido borroso con movimiento gaussiano, incrementando el nivel de ruido. La simulación se ejecuto 30 veces obteniendo un promedio del 100% de reconocimiento sin agregar ningún tipo de ruido. Para el caso en que se agrego el ruido borroso con movimiento lineal se obtuvo un promedio del 98.75% de identificación para un nivel de ruido de 10 pixeles de desplazamiento para ambos métodos de integración usando GAs (optimizar funciones de membresía) y sin optimizar, cuando se incremento el nivel a 50 pixeles de desplazamiento se obtuvo un promedio de 80 % de identificación usando GAs y 75% sin optimizar, en la tabla 4.1 se muestra un tabla comparativa de resultados entre los integradores de respuesta de lógica difusa tipo-1 y tipo-2.

Tabla 4.1. Comparación de resultados entre el Integrador de Respuestas de Lógica Difusa Tipo-1 usando GAs y sin optimizar sus funciones de membresía con niveles de ruido borroso con movimiento lineal.

Integración De Respuesta	% Rec.	% Identificación				
		Pixeles de distancia				
		10	20	30	40	50
Tipo-1 (no optimizado)	80/80	79/80	77/80	72/80	69/80	66/80
	100%	98.75%				82.5%
Tipo-1usando GAs	80/80	79/80	78/80	73/80	70/80	68/80
	100%	98.75%				85%

Posteriormente se agrego ruido borroso con movimiento radial con un nivel de 10 pixeles de desplazamiento obteniéndose un promedio del 92.5% de identificación usando GAs (optimizar las funciones de membresía) y 91.75% sin optimizar, cuando se

incremento el nivel a 50 pixeles de desplazamiento se obtuvo un promedio de 75 % de identificación usando GAs y 73.75% sin optimizar, en la tabla 4.2 se muestra un tabla comparativa de resultados.

Tabla 4.2 Comparación de resultados entre el Integrador de Respuestas de Lógica Difusa Tipo-1 usando GAs y sin optimizar sus funciones de membresía con niveles de ruido borroso con movimiento radial

<i>Integrador De Respuesta</i>	<i>% Rec</i>	<i>% Identificación</i>				
		<i>Pixeles de distancia</i>				
		<i>10</i>	<i>20</i>	<i>30</i>	<i>40</i>	<i>50</i>
Tipo-1	80/80	73/80	73/80	71/80	68/80	59/80
(no optimizado)	100%	91.75%				73.75%
Tipo-1 usando GAs	80/80	74/80	74/80	73/80	72/80	68/80
	100%	92.5%				75%

Por último se agrego ruido borroso con movimiento gaussiano con un nivel de 5 pixeles de radio obteniéndose un promedio del 92.5% de identificación usando GAs (optimizar las funciones de membresía) y 91.25% sin optimizar, cuando se incremento el nivel a 10 pixeles de radio se obtuvo un promedio de 83.75 % de identificación usando GAs y 82.5% sin optimizar, en la tabla 4.3 se muestra un tabla comparativa de resultados.

Tabla 4.3 Comparación de resultados entre el Integrador de Respuestas de Lógica Difusa Tipo-1 usando GAs y sin optimizar sus funciones de membresía con niveles de ruido borroso con movimiento gaussiano

<i>Integración De Respuesta</i>	<i>% Rec</i>	<i>% Identificación</i>	
		<i>Pixeles de distancia</i>	
		<i>5</i>	<i>10</i>
Tipo-1	80/80	73/80	66/80
(no optimizado)	100%	91.75%	82.5%
Tipo-1 usando GAs	80/80	74/80	67/80
	100%	92.5%	83.75%

Se puede observar de los resultados obtenidos en las tablas anteriores, que para los tres tipos de ruido, ruido borroso con movimiento lineal, ruido borroso con movimiento radial, y ruido borroso con movimiento gaussiano, se obtiene una mejor identificación con el integrador de respuesta de lógica difusa tipo-1 cuando se utilizan algoritmos genéticos para optimizar sus funciones de membresía.

4.2 ANALISIS DE RESULTADOS OBTENIDOS CON ENN-IT2FIS (Utilizando Algoritmos Genéticos para Optimizar las funciones de Membresía).

Las mismas condiciones que se utilizaron para obtener los resultados del integrador de respuesta de lógica difusa tipo-1 para la red neuronal ensemble, también se usaron para el integrador de respuesta de lógica difusa tipo-2 (ruido borroso con movimiento lineal, ruido borroso con movimiento radial y ruido borroso en movimiento gaussiano). Se usaron GAs para optimizar sus funciones de membresía y se incremento el nivel de ruido de 10 a 50 pixeles de distancia. La simulación se ejecuto 30 veces obteniendo un promedio del 100% de reconocimiento sin agregar ningún tipo de ruido. Para el caso en que se agrego el ruido borroso con movimiento lineal se obtuvo un promedio del 97.5% de identificación para un nivel de ruido de 10 pixeles de desplazamiento para ambos métodos de integración usando GAs (para optimizar sus funciones de membresía) y sin optimizar un promedio de 96.25%, cuando se incremento el nivel a 50 pixeles de desplazamiento se obtuvo un promedio de 76.25 % de identificación usando GAs y 73.75% sin optimizar, en la tabla 4.4 se muestra un tabla comparativa de resultados entre los integradores de respuesta de lógica difusa tipo-1 y tipo-2 usando GAs.

Tabla 4.4. Comparación de Resultados entre Tipo-1 y Tipo-2 usando GAs con ruido ruido borroso con movimiento lineal

Integración de Respuesta	% Rec.	% Identificación				
		Pixeles de Distancia				
		10	20	30	40	50
Tipo-1 (no optimizado)	80/80	79/80	77/80	72/80	69/80	66/80
	100%	98.75%				82.5%
Tipo-1 usando GAs	80/80	79/80	78/80	73/80	70/80	68/80
	100%	98.75%				85%
Tipo-2 (no optimizado)	80/80	77/80	73/80	71/80	69/80	59/80
	100%	96.25%				73.75%
Tipo-2 usando GAs	80/80	78/80	74/80	72/80	71/80	61/80
	100%	97.5%				76.25%

A continuación en la figura 4.1 se muestra el comportamiento de los integradores de respuesta de lógica difusa tipo-1 y tipo-2 basándose en la tabla 4.4.

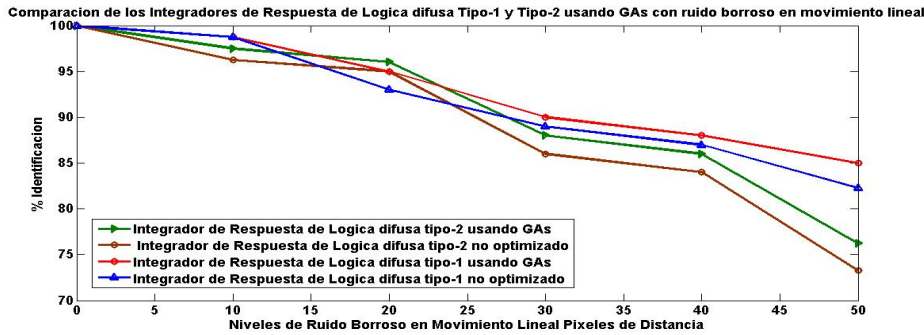


Figura 4.1. Comparación de Resultados entre Tipo-1 y Tipo-2 usando GAs con ruido borroso con movimiento lineal.

De acuerdo con los resultados obtenidos se puede observar en la figura 4.1 que el comportamiento del integrador de respuesta de lógica difusa tipo-1 usando GAs tiene una mejor identificación durante la mayor parte del rango nivel de ruido borroso con movimiento lineal con respecto al integrador de respuesta de lógica difusa tipo-2 que también usa GA's.

Posteriormente se agrego ruido borroso con movimiento radial al integrador de respuesta de lógica difusa tipo-2 con un nivel de 10 pixeles de desplazamiento obteniéndose un promedio del 92.5% de identificación usando GAs (para optimizar sus funciones de membresía) y 91.75% sin optimizar, cuando se incremento el nivel a 50 pixeles de desplazamiento se obtuvo un promedio de 75 % de identificación usando GAs y 73.75% sin optimizar, en la tabla 4.5 se muestra un tabla comparativa de resultados entre los integradores de respuesta tipo-1 y tipo-2.

Tabla 4.5. Comparación de Resultados entre Tipo-1 y Tipo-2 usando GAs con ruido borroso con movimiento radial

Integración de Respuesta	% Rec.	% Identificación Pixeles de Distancia				
		10	20	30	40	50
Tipo-1 (no optimizado)	80/80	73/80	73/80	71/80	68/80	59/80
	100%	91.75%				73.75%
Tipo-1 usando GAs	80/80	74/80	74/80	73/80	72/80	60/80
	100%	92.5%				75%
Tipo-2 (no optimizado)	80/80	78/80	73/80	70/80	63/80	63/80
	100%	97.5%				73.75%

A continuación en la figura 4.2 se muestra el comportamiento de los integradores de respuesta de lógica difusa tipo-1 y tipo-2 basándose en la tabla 4.5

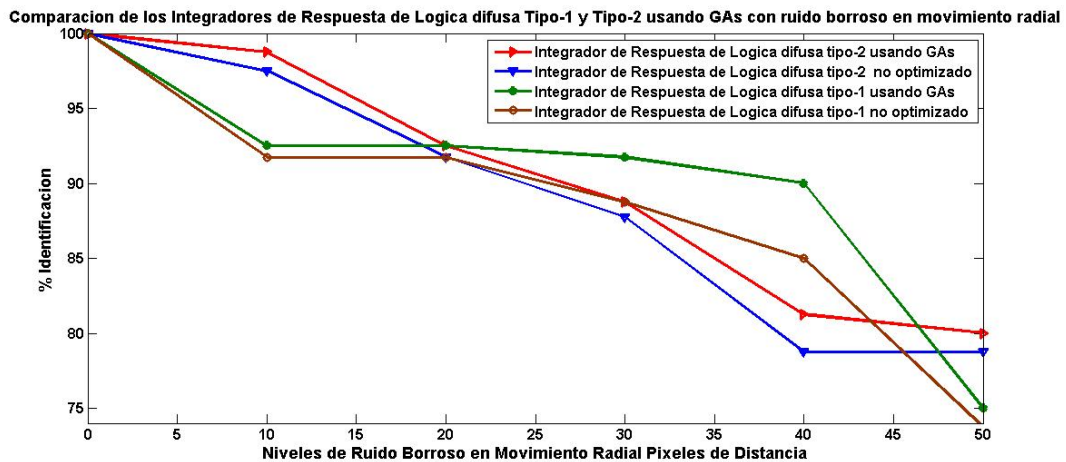


Figura 4.2. Comparación de Resultados entre Tipo-1 y Tipo-2 usando GAs con ruido borroso con movimiento radial

De acuerdo con los resultados obtenidos se puede observar en la figura 4.2 que el comportamiento del integrador de respuesta de lógica difusa tipo-2 usando GAs tiene una mejor identificación durante una parte del rango de nivel de ruido borroso en movimiento radial con respecto al integrador de respuesta de lógica difusa tipo-1 que también usa GA's.

Basándose en las tablas de resultados 4.4 y 4,5, para los tipos de ruido, ruido borroso en movimiento lineal, y ruido borroso en movimiento radial, se obtiene una mejor identificación con el integrador de respuesta de lógica difusa tipo-2 con respecto al integrador de lógica difusa tipo-1 en una parte del rango de nivel de los dos tipos de ruido antes mencionados cuando se utilizan los algoritmos genéticos para optimizar sus funciones de membresía.

4.3 ANALISIS DE RESULTADOS OBTENIDOS CON ENN-FIS-GA

(Utilizando Algoritmos Genéticos para Optimizar las Reglas Difusas).

Después de haber analizado los resultados obtenidos para el sistema híbrido inteligente con los integradores de respuesta de lógica difusa tipo-1 y tipo-2 utilizando GAs para optimizar sus funciones de membresía. Se propuso optimizar con algoritmos genéticos el número de reglas difusas de los integradores de respuesta de lógica difusa tipo-1 y tipo-2.

Comenzaremos por mencionar los parámetros del algoritmo genético (GA's) que se utilizaron para optimizar las reglas difusas, se utilice una población de 40 individuos, una razón de cruce del 70%, razón de mutación del 10%, el método de selección de la ruleta, un máximo de 100 generaciones para el sistema de lógica difusa tipo-1 y 150 generaciones para el tipo-2. Los criterios para detener al algoritmo genético son que alcance el número máximo de generaciones o si el error es cero entre el sistema difuso base propuesto y el sistema difuso obtenido con el algoritmo genético para los sistemas difusos tipo-1 y tipo-2.

A continuación en la tabla 4.6 se muestran los resultados obtenidos para los integradores de lógica difusa tipo-1 y tipo-2 usando GAs para optimizar sus reglas con diferentes niveles de ruido de ruido borroso con movimiento lineal de 10 a 50 píxeles de distancia.

Tabla 4.6 Comparación entre Tipo-1 y Tipo-2 usando GAs con ruido borroso con movimiento lineal

Metodo de Integración	% Rec.	% Identificación				
		Píxeles de Distancia				
		10	20	30	40	50
Tipo-1 usando GAs	80/80	79/80	74/80	69/80	64/80	60/80
	100%	98.75%				75%
Tipo-2 usando GAs	80/80	78/80	74/80	72/80	71/80	61/80
	100%	97.5%				76.2%

En la figura 4.3 se muestra una comparación de resultados con los integradores de respuesta de lógica difusa tipo-1 y tipo 2 en redes neuronales ensemble con niveles de ruido borroso con movimiento lineal.

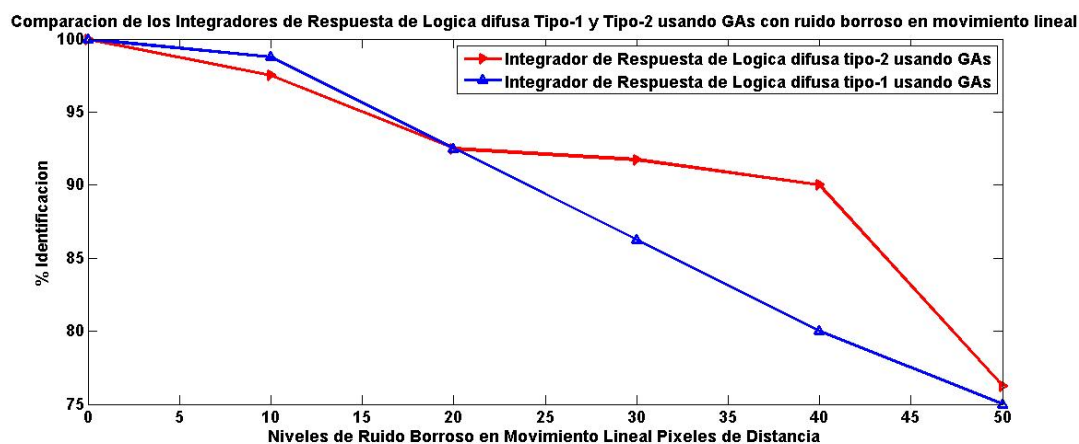


Figura 4.3 Comparación de resultados entre lógica difusa tipo-1 y tipo-2 usando GAs con diferentes niveles de ruido borroso en movimiento lineal.

A continuación en la tabla 4.7 se muestran los resultados obtenidos para los integradores de respuesta de lógica difusa tipo-1 y tipo-2 usando GAs para optimizar sus reglas usando niveles de ruido de ruido borroso con movimiento radial de 10 a 50 pixeles de distancia.

Tabla 4.7 Comparación entre Tipo-1 y Tipo-2 usando GAs con ruido borroso con movimiento radial.

Integrador de respuesta	% Rec.	% Identificacion				
		Pixeles de Distancias				
		10	20	30	40	50
Tipo-1 usando GAs	80/80	74/80	74/80	73/80	72/80	60/80
	100%	92.5%				75%
Tipo-2 usando GAs	80/80	79/80	74/80	71/80	65/80	64/80
	100%	98.75%				80%

En la figura 4.4 se muestra una comparación de resultados con los integradores de respuesta de lógica difusa tipo-1 y tipo 2 en redes neuronales ensemble con niveles de ruido borroso en movimiento radial.

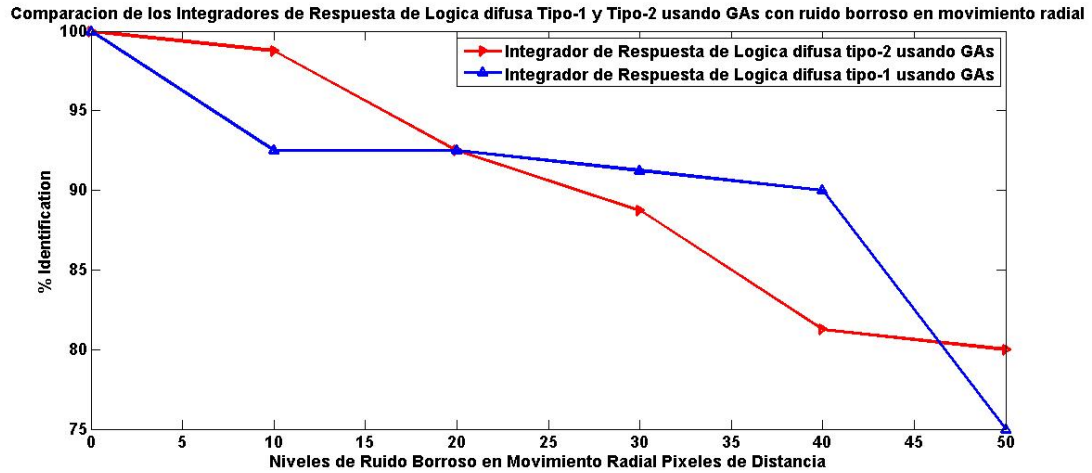


Figura 4.4 Comparación de resultados entre lógica difusa tipo-1 y tipo-2 usando GAs con diferentes niveles de ruido borroso en movimiento radial.

4.4 ANALISIS DE RESULTADOS OBTENIDOS CON ENN-IT2FIS (Utilizando Algoritmos Genéticos para Optimizar las Reglas Difusas).

Todos los resultados que se han obtenido, utilizaron los mismos parámetros para cada uno de los tres módulos de la red neuronal ensemble, es decir cada modulo es una red neuronal monolítica con las mismas características.

Se propuso usar diferentes parámetros para cada modulo de la red neuronal ensemble, para el modulo 1 la razón de aprendizaje es .001 y el error meta .001, para el modulo 2 la razón de aprendizaje es .0001 y el error meta .0001, y para el modulo 3 la razón de aprendizaje es .0001 y el error meta .00001, de igual manera como se han obtenido los resultados de las pruebas anteriores se incremento el nivel de ruido borroso en movimiento lineal, y ruido borroso en movimiento radial de 10 a 90 pixeles de distancia. Los resultados obtenidos se muestran las tablas 4.8 y 4.9.

Tabla 4.8. Comparación entre Tipo-1 y Tipo-2 con los parámetros modificados para cada modulo de la red neuronal ensemble usando ruido borroso con movimiento lineal.

Integracion Respuesta	% Rec.	% Identificación Pixeles de Distancia					
		10	20	30	40	50	90
Tipo-2 usando GAs par. modif. para cada modulo	80/80						
	100%	100%	98.75%	97.5%	91.25%	88.75%	68.75%
Tipo-2 usando GAs	80/80						
	100%	97.5%	90%	87.5%	83.75%	78.75%	55%
Tipo-1 usando GAs par. modif. para cada modulo	80/80						
	100%	100%	98.75%	96.25%	90%	88.75%	68.75%
Tipo-1 usando GAs	80/80						
	100%	97.5%	93.75%	90%	85%	77.5%	53.75%

Tabla 4.9. Comparación entre Tipo-1 y Tipo-2 con los parámetros modificados para cada modulo de la red neuronal ensemble usando ruido borroso con movimiento radial.

Integracion Respuesta	% Rec.	% Identificación Pixeles de Distancia					
		10	20	30	40	50	90
Tipo-2 usando GAs par. modif. para cada modulo	80/80						
	100%	100%	96.25%	93.75%	88.75%	85%	67.5%
Tipo-2 usando GAs	80/80						
	100%	98.75%	93.75%	90%	81.25%	71.25%	50%
Tipo-1 usando GAs par. modif. para cada modulo	80/80						
	100%	100%	96.25%	95%	91.25%	82.5%	66.25%
Tipo-1 usando GAs	80/80						
	100%	98.75%	93.75%	90%	82.5%	71.25%	48.75%

En la figura 4.5 se muestran los resultados obtenidos para los integradores de respuesta de lógica difusa tipo-1 y tipo-2, usando diferentes parámetros para cada modulo de la red neuronal ensemble, usando GAs para optimizar sus reglas, y con diferentes niveles de ruido borroso con movimiento lineal basándose en las tablas 4.8.

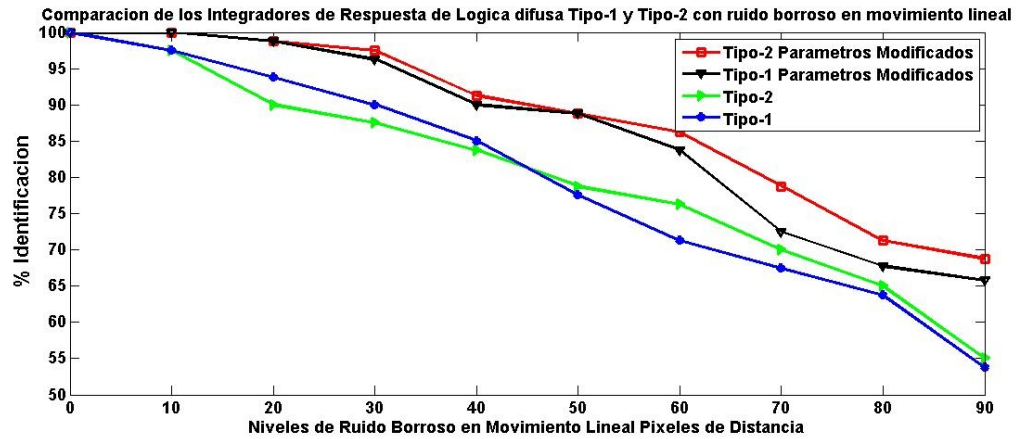


Figura 4.5 Comparación de resultados entre lógica difusa tipo-1 y tipo-2 usando GAs con diferentes niveles de ruido borroso en movimiento lineal.

En la figura 4.6 se muestran los resultados obtenidos basándose en la tabla 4.9 para los integradores de respuesta de lógica difusa tipo-1 y tipo-2, usando diferentes parámetros para cada modulo de la red neuronal ensemble, y con diferentes niveles de ruido borroso con movimiento radial.

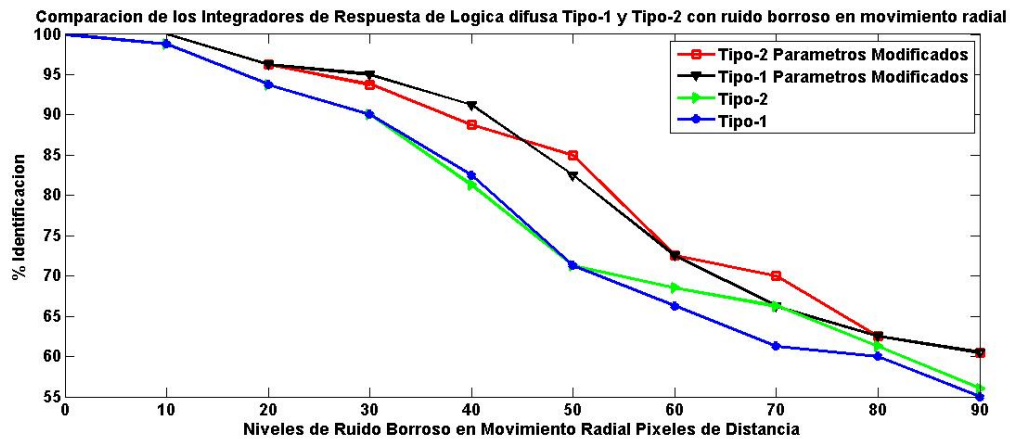


Figura 4.6 Comparación de resultados entre lógica difusa tipo-1 y tipo-2 usando GAs con diferentes niveles de ruido borroso en movimiento radial.

Analizando los resultados obtenidos al utilizar diferentes parámetros para cada uno de los módulos de la red neuronal ensemble se observa una mejora significativa en el porcentaje de identificación de los integradores de respuesta de lógica difusa tipo-1 y tipo-2 usando algoritmos genéticos para optimizar sus reglas con diferentes niveles de ruido borroso en movimiento lineal, el porcentaje de identificación aumento un

promedio de 8.75% para el integrador de lógica difusa tipo-2 y un 7.5% para el integrador de respuesta lógica difusa tipo-1.

También se observa un aumento en el porcentaje de identificación de los integradores de respuesta de lógica difusa tipo-1 y tipo-2 usando algoritmos genéticos para optimizar sus reglas con diferentes niveles de ruido borroso con movimiento radial, el porcentaje de identificación aumento un promedio de 7.70% para ambos integradores de lógica difusa tipo-1 y tipo-2.

Respecto a los resultados entre los integradores de lógica difusa tipo-2 y tipo-1 en promedio no hay un porcentaje de diferencia en promedio entre los dos integradores de respuesta tipo-2 y tipo-1.

4.5 Evaluación de resultados usando SNR (Relación señal a ruido)

Para evaluar los resultados obtenidos para los integradores de respuesta con diferentes niveles de ruido borroso en movimiento lineal y ruido borroso en movimiento radial, se propuso utilizar la relación señal a ruido SNR para imágenes [45,46]. Únicamente se generaron resultados para el ruido borroso en movimiento lineal. A continuación se muestra la ecuación 4.1 que se utilizo para generar la tabla de valores de la relación señal a ruido en decibels para los diferentes niveles de ruido borroso en movimiento lineal aplicados a cada una de las huellas digitales.

$$\sigma_I^2 = \frac{1}{2} \text{var}(I_1 - I_2), \text{ where } I_1 = I + \eta_1, I_2 = I + \eta_2 \quad (4.1)$$

σ_I = Desviación estándar de la imagen de la huella digital original.

var = Varianza entre la imagen de la huella digital original y la imagen de la huella digital aplicándole ruido borroso con movimiento lineal.

$\eta_1 = 0$, significa que no se aplica ruido a la imagen de la huella digital original.

η_2 = nivel de ruido blur mtain en un rango de 10 a 90 pixeles de distancia.

I_2 , es la imagen de la huella digital aplicándole ruido borroso con movimiento lineal.

La relación de señal a ruido (SNR) de una imagen I se calcula como se muestra en la ecuación 4.2 que se muestra a continuación.

$$SNR = 10\log \left(\frac{var I}{\sigma^2 I} \right) \quad (4.2)$$

Basándose en las ecuaciones 4.1 y 4.2 se calcularon los valores de SNR para cada una de las huellas digitales con un rango de nivel de ruido borroso con movimiento lineal de 10 a 90 pixeles de distancia como se muestra a continuación en la tabla 4.10.

Tabla 4.10 Relacion señal a ruido promedio para ASNR para cada una de las imágenes de las huellas digitales de las personas con un nivel de ruido borroso con movimiento lineal de 10 a 90 pixeles de distancia para los integradores de respuesta de lógica difusa tipo-1 y tipo-2.

# Persona	Ruido borroso con movimiento lineal								
Huella Digital	10	20	30	40	50	60	70	80	90
Promedio SNR P1 db	10.97	7.75	8.08	7.65	7.43	7.28	7.11	6.99	6.89
Difuso Tipo-2 %	100	87.5	87.5	87.5	87.5	62.5	62.5	62.5	62.5
Difuso Tipo-1 %	87.5	87.5	87.5	87.5	75	62.5	62.5	62.5	62.5
Promedio SNR P2 db	8.33	8.02	7.52	7.38	7.16	7.01	6.89	6.78	6.68
Difuso Tipo-2 %	100	100	87.5	100	87.5	75	62.5	50	37.5
Difuso Tipo-1 %	100	100	87.5	100	87.5	75	75	50	50
Promedio SNR P3 db	8.01	7.02	6.76	6.56	6.44	6.34	6.21	6.10	5.98
Difuso Tipo-2 %	100	100	87.5	87.5	62.5	62.5	37.5	25	12.5
Difuso Tipo-1 %	100	100	87.5	75	75	50	0	0	0
Promedio SNR P4 db	7.67	5.75	5.57	5.46	5.26	5.17	5.075	5.01	4.94
Difuso Tipo-2 %	100	100	87.5	87.5	75	75	62.5	62.5	25
Difuso Tipo-1 %	87.5	87.5	87.5	87.5	75	75	62.5	62.5	37.5
Promedio SNR P5 db	11.65	10.62	10.08	9.77	9.47	9.28	9.08	8.89	8.71
Difuso Tipo-2 %	100	100	100	100	100	87.5	87.5	75	50
Difuso Tipo-1 %	100	100	100	100	100	87.5	75	75	50
Promedio SNR P6 db	7.29	7.28	6.88	6.75	6.58	6.46	6.33	6.21	6.1
Difuso Tipo-2 %	100	100	100	100	100	100	100	87.5	75
Difuso Tipo-1 %	100	100	100	100	100	100	100	87.5	75
Promedio SNR P7 db	8.21	6.11	6.26	6.18	5.93	5.85	5.79	5.67	5.57
Difuso Tipo-2 %	100	100	100	87.5	87.5	87.5	62.5	50	37.5
Difuso Tipo-1 %	100	100	100	75	75	87.5	62.5	50	37.5
Promedio SNR P8 db	10.64	10.07	9.40	9.09	8.8	8.56	8.47	8.15	7.97
Difuso Tipo-2 %	100	100	100	87.5	87.5	87.5	62.5	50	37.5
Difuso Tipo-1 %	100	100	100	75	75	87.5	62.5	50	37.5
Promedio SNR P9 db	12.76	12	11.65	11.35	11.7	10.9	10.81	10.6	10.42
Difuso Tipo-2 %	100	100	100	100	100	100	100	100	100
Difuso Tipo-1 %	100	100	100	100	100	100	100	100	100
PromedioSNR P10 db	8.94	6.43	6	5.79	5.52	5.33	5.24	5.14	5.03
Difuso Tipo-2 %	100	100	100	100	100	100	100	100	100
Difuso Tipo-1 %	100	100	100	100	100	100	100	100	100

Analizando los resultados de la tabla 4.10 se puede observar que las imágenes de la las huellas digitales que pertenecen a las personas 4, 7 y 10 son las que presentan un nivel de

ruido mayor (una menor relación señal a ruido promedio ASNR) durante el rango de los niveles de ruido borroso con movimiento lineal, y las imágenes de las personas 5 y 9 son las que presentan un nivel de ruido menor (una mayor relación señal a ruido promedio ASNR) durante el rango de los niveles de ruido borroso con movimiento lineal, base de datos completa de las huellas digitales clasificadas por personas se muestra continuación en la figura 4.7.

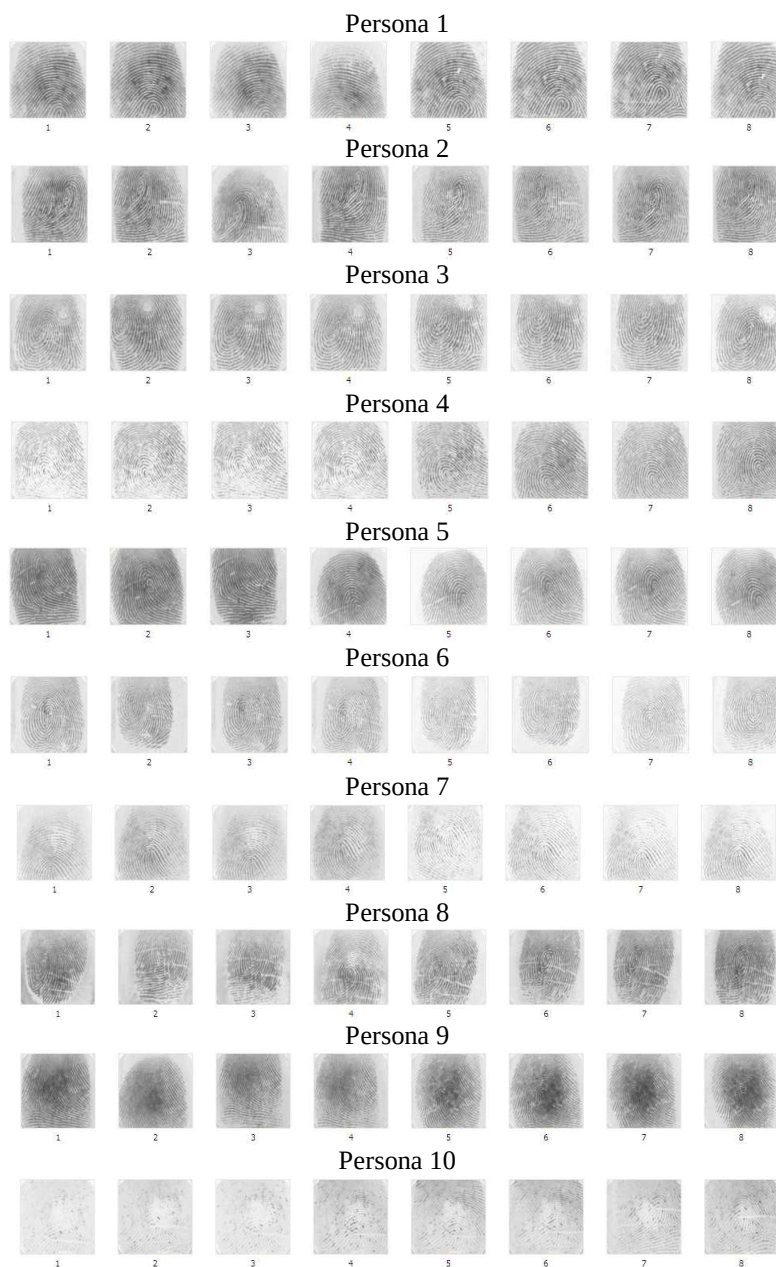


Figura 4.7 Base de datos Huellas digitales originales clasificadas por personas.

A continuación en las figura 4.8 a 4.17 se muestra por persona en la parte superior el comparativo de la respuesta de los integradores de lógica difusa tipo-1 y tipo-2, y en la parte inferior el promedio de la relación señal a ruido ASNR cuando se usa ruido borroso con movimiento lineal para el rango de niveles de 0 a 90 pixeles de distancia.

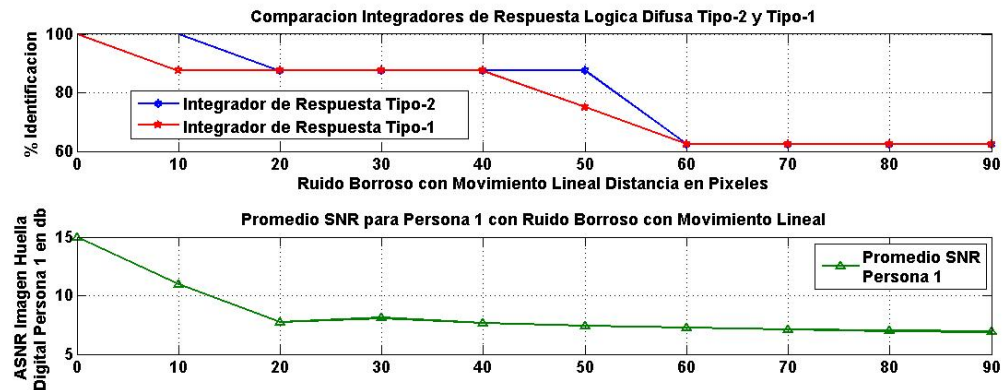


Figura 4.8 Comparativo Integradores de Respuesta de Lógica difusa tipo-1 y tipo-2 con respecto a la relación señal a ruido promedio ASNR para las imágenes de la persona 1.

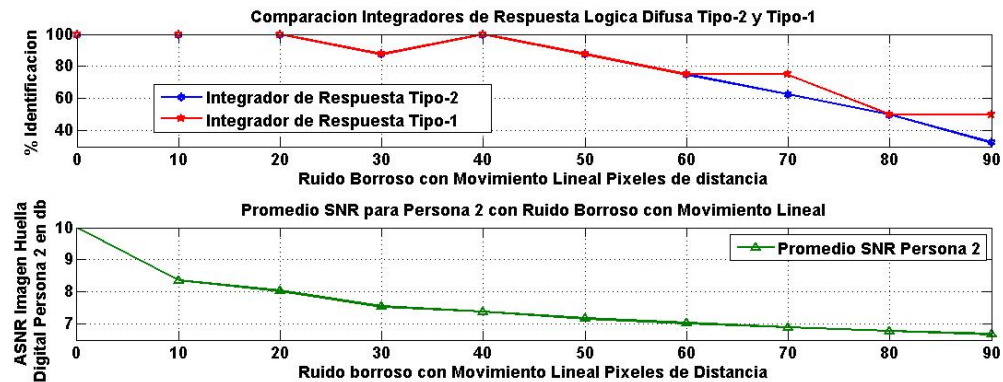


Figura 4.9 Comparativo Integradores de Respuesta de Logica difusa tipo-1 y tipo-2 con respecto a la relación señal a ruido promedio ASNR para las imágenes de la persona 2.

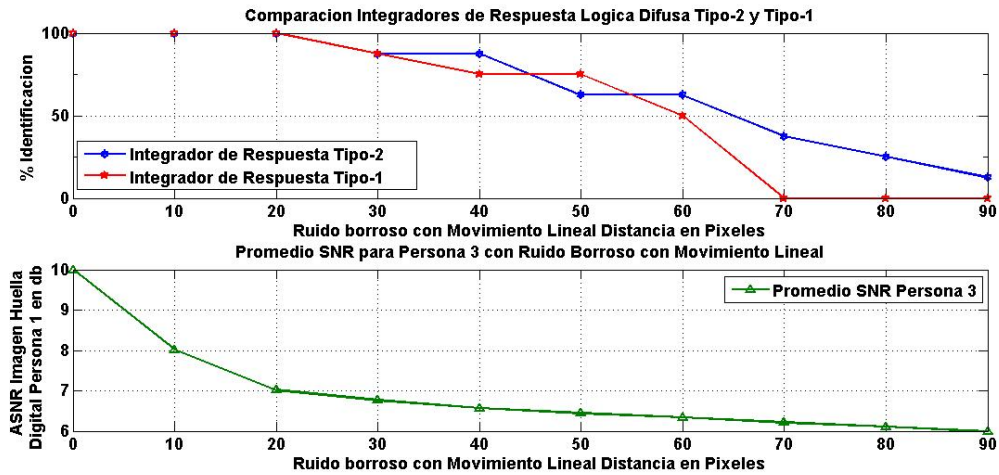


Figura 4.10 Comparativo Integradores de Respuesta de Lógica difusa tipo-1 y tipo-2 con respecto a la relación señal a ruido promedio ASNR para las imágenes de la persona 3.

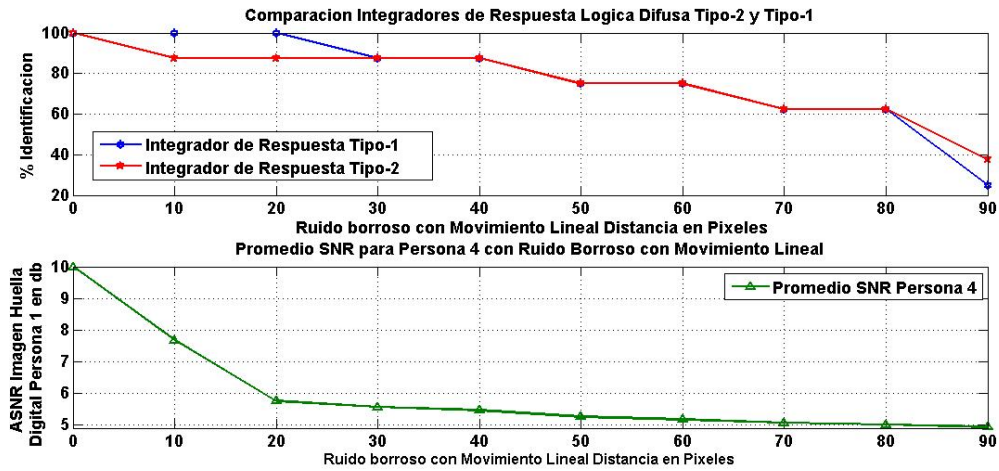


Figura 4.11 Comparativo Integradores de Respuesta de Lógica difusa tipo-1 y tipo-2 con respecto a la relación señal a ruido promedio ASNR para las imágenes de la persona 4.

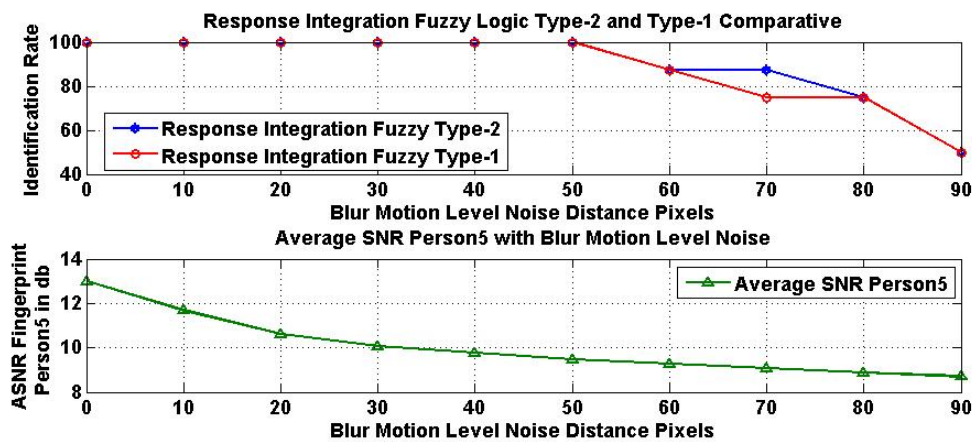


Figura 4.12 Comparativo Integradores de Respuesta de Lógica difusa tipo-1 y tipo-2 con respecto a la relación señal a ruido promedio ASNR para las imágenes de la persona 5.

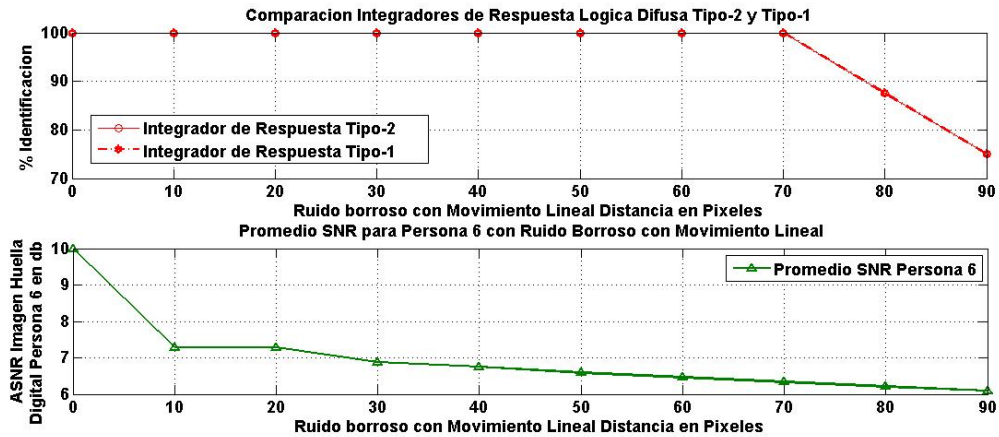


Figura 4.13 Comparativo Integradores de Respuesta de Lógica difusa tipo-1 y tipo-2 con respecto a la relación señal a ruido promedio ASNR para las imágenes de la persona 6.

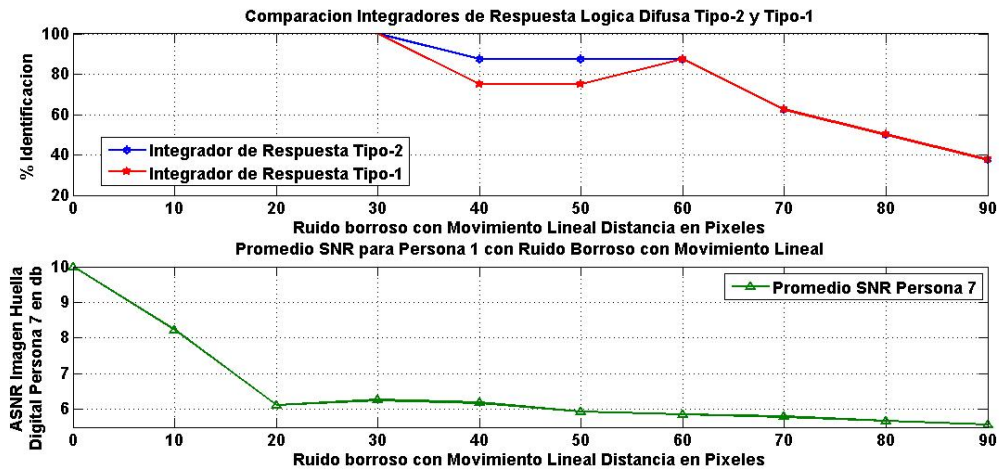


Figura 4.14 Comparativo Integradores de Respuesta de Lógica difusa tipo-1 y tipo-2 con respecto a la relación señal a ruido promedio ASNR para las imágenes de la persona 7.

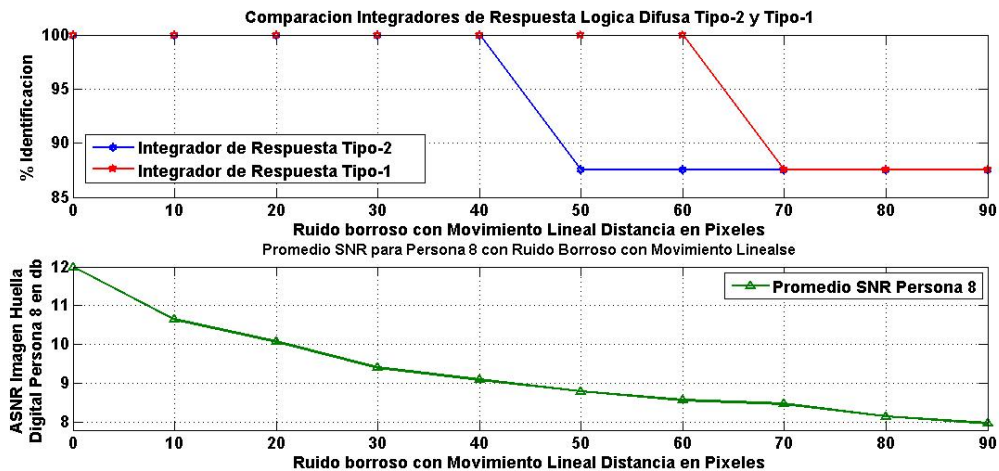


Figura 4.15 Comparativo Integradores de Respuesta de Lógica difusa tipo-1 y tipo-2 con respecto a la relación señal a ruido promedio ASNR para las imágenes de la persona 8.

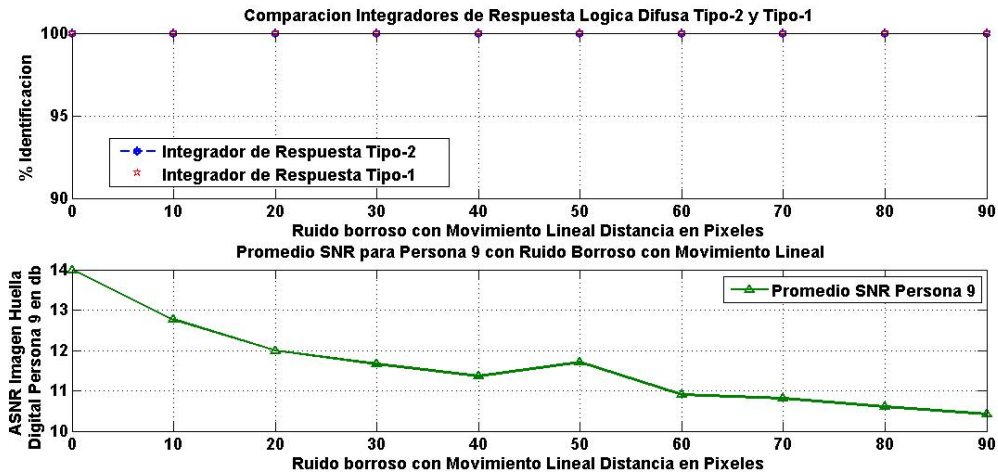


Figura 4.16 Comparativo Integradores de Respuesta de Lógica difusa tipo-1 y tipo-2 con respecto a la relación señal a ruido promedio ASNR para las imágenes de la persona 9.

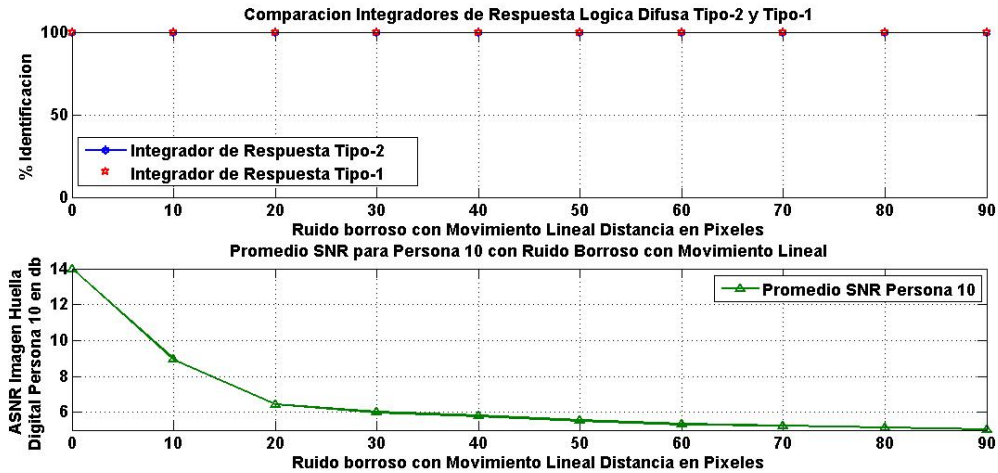


Figura 4.17 Comparativo Integradores de Respuesta de Lógica difusa tipo-1 y tipo-2 con respecto a la relación señal a ruido promedio ASNR para las imágenes de la persona 10.

A continuación en la figura 4.18 se muestra el resumen del comportamiento de las huellas digitales clasificadas por personas con respecto al promedio de la relación señal a ruido en decibeles ASNR cuando se incrementa el nivel de ruido borroso en movimiento lineal de 0 a 90 pixeles de distancia, se observa que la huella digital de la persona que tiene en promedio menos ruido (una relación señal a ruido alta) es la persona 9 (P9), y la huella de la persona que tiene en promedio más ruido (una relación señal a ruido baja) es la persona 4.

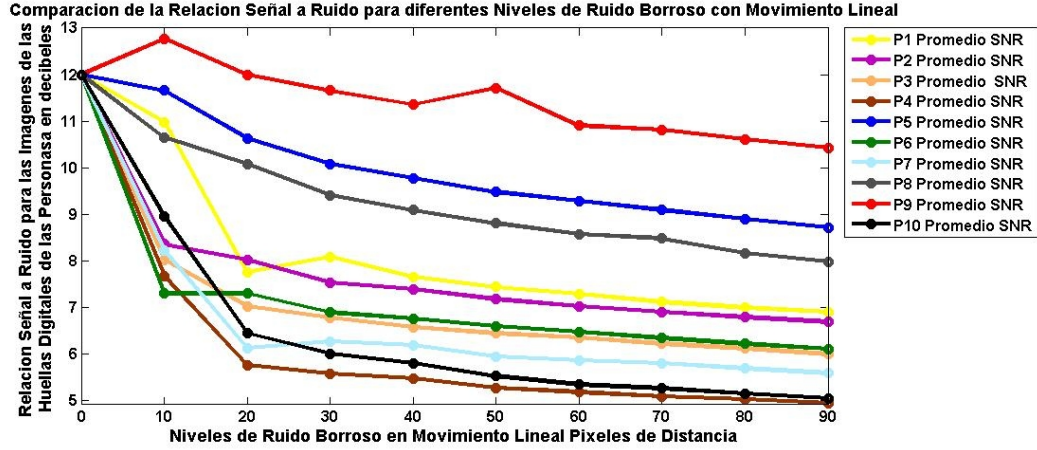


Figura 4.18 Comparación de la relación señal a ruido ASNR para cada una de las huellas digitales por persona (P1 a P10) para un rango de nivel de ruido borroso en movimiento lineal de 0 a 90 píxeles de distancia.

4.6 Fiabilidad del Sistema de Reconocimiento de Patrones

A continuación se hace un análisis para la evaluación de los resultados obtenidos para los integradores de respuesta de lógica difusa tipo-1 y tipo-2 utilizando el cálculo de la fiabilidad para imágenes [47]. La fiabilidad de un sistema de Reconocimiento de Patrones se calcula por medio de su porcentaje de reconocimiento R_c , su porcentaje de rechazo R_r y su porcentaje de error R_e de manera general, como se define a continuación en la ecuación 4.3.

$$Fiabilidad = \frac{R_c}{1 - R_r} = \frac{R_c}{R_c + R_r} \quad (4.3)$$

Se realizaron los cálculos de la fiabilidad para los métodos de integración de respuesta de lógica difusa tipo-2 y lógica difusa tipo-1 con los parámetros modificados para cada una de los módulos usando algoritmos genéticos para optimizar sus reglas, y con los integradores de respuesta de lógica difusa tipo-2 y tipo-1 módulos usando algoritmos genéticos para optimizar sus reglas, los resultados que se muestran a continuación en la tabla 4.11 únicamente se realizaron utilizando diferentes niveles de ruido para ruido borroso con movimiento lineal.

Tabla 4.11. Comparación de la fiabilidad entre los integradores de respuesta de lógica difusa Tipo-1 y Tipo-2 con los parámetros modificados para cada modulo de la red neuronal ensemble usando ruido borroso con movimiento radial.

Integracion Respuesta	% Fiabilidad	% Identificación Pixeles de Distancia					
		10	20	30	40	50	90
Tipe-2 usando GAs par. modif. para cada modulo	80/80						
	100%	100%	98.75%	96%	93%	88.75%	71.42%
Tipo-2 usando GAs	80/80						
	100%	97.5%	93.75%	90%	85%	77.5%	55%
Tipo-1 usando GAs par. modif. para cada modulo	80/80						
	100%	100%	97.5%	95%	92.5%	88.75%	70.14%
Tipo-1 usando GAs	80/80						
	100%	97.5%	93.75%	90%	85%	77.25%	53.75%

Basándose en la tabla 4.11 se muestra en la figura 4.18 una comparación de la fiabilidad de los integradores de respuesta de lógica difusa tipo-2 y tipo-1, con los parámetros modificados para cada uno de los módulos de la red neuronal ensemble usando algoritmos genéticos para optimizar sus reglas, y los integradores de respuesta de lógica difusa tipo-2 y tipo-1 usando algoritmos genéticos para optimizar sus reglas.

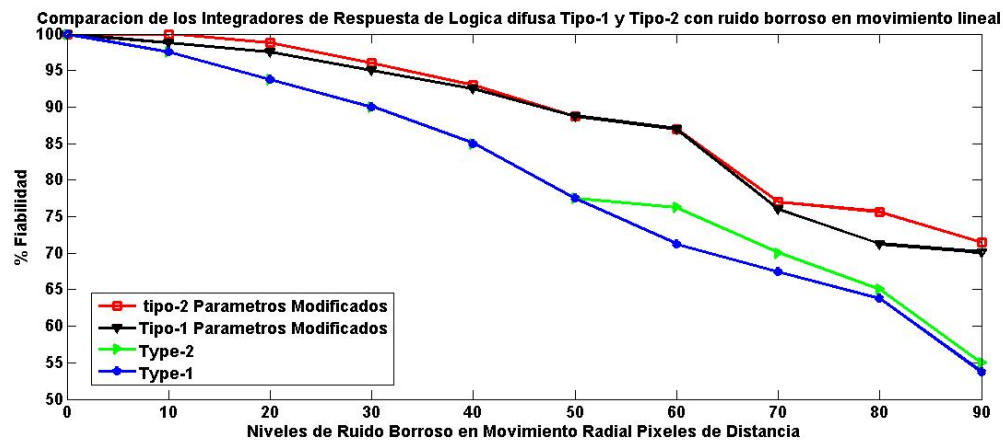


Figura 4.18 Comparación de la fiabilidad de los integradores de respuesta de lógica difusa tipo-2 y tipo-1.

4.7 Prueba Estadística t Student para comparación de los resultados de los integradores de respuesta lógica difusa tipo-2 y tipo-1

Para comparar los resultados se utilizó la prueba t Student, la cual es un método de análisis estadístico. Es una prueba paramétrica, basado en la comparación de variables numéricas de distribución normal. En caso de tener que analizar variables numéricas de distribución no normal, se debe utilizar otro tipo de pruebas no paramétricas.

La ecuación para la prueba estadística t-student es una relación (una división). El numerador es la diferencia entre las dos medias de las variables que se quieran analizar, para nuestro caso analizaremos los resultados entre los integradores de respuesta de lógica difusa tipo-2 y tipo-1. El denominador es una medida de la variabilidad o dispersión de los resultados. Esta ecuación es otro ejemplo de la relación señal a ruido, la diferencia entre las medias es la señal y la medida de la variabilidad es el ruido. A continuación en la ecuación 4.1 se muestra la ecuación para t-student.

La prueba de hipótesis para diferencia en las medias μ_1 y μ_2 de dos poblaciones:

$$H_0 : \mu_1 = \mu_2 \quad (4.1)$$

$$H_1 : \mu_1 \neq \mu_2 \quad (4.2)$$

Se basa en la estadística t de student

$$t = \frac{\bar{\mu}_1 - \bar{\mu}_2}{\sqrt{\frac{\sigma_{\bar{\mu}_1}^2}{n} + \frac{\sigma_{\bar{\mu}_2}^2}{n}}} \quad (4.3)$$

y si t se encuentra entre $-t_{\alpha/2} < t < t_{\alpha/2}$ se acepta H_0 , en caso contrario se acepta H_1 .

$$\bar{\mu}_1 = \text{Media del Integrador de Respuesta de Logica Difusa Tipo - 2} \quad (4.2)$$

$$\bar{\mu}_2 = \text{Media del Integrador de Respuesta de Logica Difusa Tipo - 1} \quad (4.3)$$

$$\sigma_{\bar{\mu}_1}^2 = \text{Varianza del Integrador de Respuesta de Logica Difusa Tipo - 2} \quad (4.4)$$

$$\sigma_{\bar{\mu}_2}^2 = \text{Varianza del Integrador de Respuesta de Logica Difusa Tipo - 2} \quad (4.5)$$

$$n = 14 = \text{Numero de resultados de cada Integrador de Respuesta} \quad (4.6)$$

El valor de t será positivo si la primera media es más grande que la segunda y negativo si es más pequeña. Una vez que se calcula el valor de t se tiene que observar el significado de la prueba, si la relación es lo suficientemente grande se dice que la diferencia entre los grupos es evidente, para nuestro caso sería mostrar si hay una evidencia lo suficientemente grande para considerar si los resultados obtenidos entre los integradores de respuesta de lógica difusa tipo-2 y los de tipo-1. Si el resultado de t es mayor a ± 2 entonces existe suficiente evidencia estadística, y si está entre ± 1.95 entonces se considera que falta evidencias estadística para demostrar que un método de integración es mejor que otro.

Considerando las ecuaciones anteriores se procedió a realizar el cálculo de la t entre los integradores de respuesta de lógica difusa tipo-2 y tipo-1, los resultados se muestran a continuación en la tabla 4.12.

Tabla 4.12. Comparación del valor t student, media, desviación estándar y varianza entre los integradores de respuesta de lógica difusa tipo-2 y tipo-1.

Integradores de Respuesta	t student	Media	Desviación Estándar	Varianza
		$\bar{\mu}_1$ $\bar{\mu}_2$	$\sigma_{\bar{\mu}_1}$ $\sigma_{\bar{\mu}_2}$	$\sigma_{\bar{\mu}_1}^2$ $\sigma_{\bar{\mu}_2}^2$
Tipo-2 y Tipo-1, usando GAs, con par.Mod. ENN	.3541	89.85 88.45	10.123 11.537	102.48 133.1
Tipo-2 usando GAs con par. Mod. y Tipo-2 no optimizado	2.23	89.85 81.01	10.123 11.486	102.48 131.92
Tipo-1 usando GAs con par. Mod. y Tipo-1 no optimizado	2.0012	88.45 79.43	11.537 13.093	133.1 171.42
Tipo-2 usando GAs con par. Mod. y Tipo-1 no optimizado	2.4384	89.85 79.43	10.123 13.093	102.48 171.42
Tipo-2 usando GAs con par. Mod. y Red Neuronal Monolítica	2.7181	89.85 77.46	10.123 13.818	102.48 190.93
Tipo-1 usando GAs con par. Mod. Y Red Neuronal Monolítica	2.2866	88.45 77.46	11.537 13.818	133.1 190.93

Observándose los resultados obtenidos en la tabla 4.12 se puede concluir que los métodos que superan el valor 2 del estadístico t-student, son el integrador de respuesta de lógica difusa tipo-2 usando algoritmos genéticos con respecto al integrador de lógica difusa tipo-1 no optimizado, integrador de respuesta de lógica difusa tipo-2 usando algoritmos genéticos con respecto al integrador de lógica difusa tipo-2 no optimizado, y también el integrador de lógica difusa tipo-1 usando algoritmos genéticos con respecto al integrador de respuesta de lógica difusa tipo-1 no optimizado, demostrando que hay suficiente evidencia para decir que son mejores los sistemas difusos que fueron optimizados. En otras palabras, existe suficiente evidencia estadística para decir que la optimización por medio de GAs resulto en una mejora significativa en los resultados de reconocimiento.

CAPÍTULO 5. CONCLUSIONES

En el trabajo de investigación de esta tesis se han abordado los sistemas híbridos inteligentes para reconocimiento de patrones neuro-difusos tipo-2 y neuro –difusos tipo-1, utilizando algoritmos genéticos para optimizar las funciones de membresía y las reglas de los sistemas de lógica difusa tipo-1 y tipo-2. Se propuso una arquitectura para una red neuronal ensemble, para los siguientes modelos ENN-FIS (Utilizando Algoritmos Genéticos para Optimizar las funciones de Membresía), ENN-FIS2 (Utilizando Algoritmos Genéticos para Optimizar las funciones de Membresía), ENN-FIS-GA (Utilizando Algoritmos Genéticos para Optimizar las Reglas Difusas) y ENN-FIS2-GA (Utilizando Algoritmos Genéticos para Optimizar las Reglas Difusas).

Basándonos en los resultados obtenidos se puede concluir que cuando se utiliza la combinación de dos o más métodos de inteligencia artificial para abordar un problema de gran complejidad para reconocimiento de patrones se obtienen mejores resultados, en el presente trabajo se realizaron todas las pruebas para una base de datos de huellas digitales.

Se agrego niveles de 0 a 90 pixeles de distancia de ruido borroso con movimiento lineal, y ruido borroso con movimiento radial, y de 0 a 10 pixeles de distancia de ruido borroso movimiento gaussiano, para evaluar la respuesta del sistema de reconocimiento de patrones híbrido, el porcentaje de identificación de las huellas digitales aumento de manera significativa cuando se utilizaron algoritmos genéticos para optimizar las funciones de membresía y las reglas de los sistemas difusos de lógica difusa tipo-2 y tipo-1.

También es importante mencionar que cuando se utilizaron diferentes parámetros para cada uno de los tres módulos de la red neuronal ensemble, para que cada uno de los módulos sea experto, el porcentaje de identificación del sistema de reconocimiento híbrido aumento.

Se observó que la identificación de los integradores de respuesta de lógica difusa tipo-2, y tipo-1 en algunos rangos de los niveles de ruido borroso con movimiento lineal, y ruido borroso con movimiento radial era mejor alguno de los dos, es decir para todo el rango de niveles de ruido totalmente un solo integrador no tenía siempre el más alto porcentaje de identificación, por lo que se podría trabajar en un tercer integrador difuso que evalúe la respuesta de los dos y obtenga la respuesta del que tenga el más alto porcentaje de identificación.

El análisis del sistema híbrido de Reconocimiento de Patrones utilizando la relación señal a ruido SNR para imágenes en decibeles, demostró el comportamiento de los integradores de respuesta de lógica difusa tipo-2 y tipo-1, que al aumentar la relación SNR (el ruido disminuye) y al disminuir la relación señal a ruido (el ruido aumenta), para la mayoría de las muestras de las huellas digitales cuando se incrementa el SNR aumenta su porcentaje de identificación y cuando decrece el SNR disminuye el porcentaje de identificación, solo en uno de los casos de las muestras de las huellas muy particular no afectó la disminución del SNR, aumentar el nivel de tipo de ruido, agregar otros tipos de ruido, la identificación siempre se obtuvo un 100% de identificación, lo que se puede concluir con esas muestras es que tienen mucha iluminación, por lo que se podría calcular su nivel de iluminación y usar el mismo nivel de iluminación para todas las muestras que entran a la red neuronal ensemble usándolo como un preprocesamiento de imágenes para la entrada de las redes neuronales ensemble.

El análisis de la fiabilidad del Sistema de Reconocimiento de Patrones demostró que el sistema híbrido tiene un alto porcentaje de fiabilidad para el tratamiento de huellas digitales.

La prueba estadística de t student para el caso de los integradores de respuesta de lógica difusa tipo-2 usando algoritmos genéticos tuvo suficiente evidencias para demostrar que es un mejor método de elección con respecto a los integradores de respuesta de lógica difusa tipo-2 y tipo-1 no optimizados (que no usan algoritmos genéticos).

Se comprobó que al utilizar un sistema híbrido inteligente, es decir combinar dos o más métodos de inteligencia artificial se logra obtener para el caso de reconocimiento de patrones un porcentaje mayor de identificación.

TRABAJOS FUTUROS

Los resultados obtenidos en la presente tesis se pueden seguir mejorando, por lo que se pretende seguir en la búsqueda de aplicar algoritmos híbridos inteligentes utilizando las técnicas de lógica difusa tipo-2, algoritmos genéticos y redes neuronales artificiales para reconocimiento de patrones.

Desarrollar un algoritmo genético que tenga como objetivo optimizar las reglas difusas y las funciones de membresía al mismo tiempo de los integradores de respuesta de lógica difusa tipo-2 y tipo-1.

Otro línea que se pretende investigar es utilizar algoritmos híbridos inteligentes para preprocesamiento de imágenes, para el caso de huellas digitales trabajar con un algoritmo genético y lógica difusa tipo-2 para adaptar el rango de iluminación de las imágenes que procese el sistema de reconocimiento de patrones.

REFERENCIAS

- [1]. Arun D. Kulkarni , Computer Vision and Difuso-Neural Systems, Prentice Hall, ISBN0-13-570599-1,2001.
- [2]. T. Huang, S. D. Blostein and E. A. Margerum , “Least-squares estimation of motion parameters from 3D point correspondences”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition., pp 198 -200, 1986.
- [3]. Sepulveda, R., Castillo, O., Melin, P., Rodriguez-Diaz, A., Montiel, O., “Experimental study of intelligent controllers under uncertainty using Tipo-1 and Tipo-2 difuso logic”. INFORMATION SCIENCES 177(10):2023-2048, 2007.
- [4]. Singh, M., Srivastava, S., Gupta, J.R.P., Hanmandlu, M., “A new algorithm-based Tipo-2 difuso controller for diabetic patient”. INTERNATIONAL JOURNAL OF BIOMEDICAL ENGINEERING AND TECHNOLOGY 1(1):18-40, 2007.
- [5]. Astudillo, L., Castillo, O., Melin, P., Alanis, A., Soria, J., Aguilar, L., “Intelligent Control of an Autonomous Mobile Robot using Tipo-2 Difuso Logic”. Journal of Engineering Letters 13(2): 93-97, September 2006.
- [6]. Baguley, P., Page, T., Koliza, V., Maropoulos, P., “Time to market prediction using Tipo-2 difuso sets”. JOURNAL OF MANUFACTURING TECHNOLOGY MANAGEMENT 17(4): 513-520, 2006.
- [7]. N. N. Karnik and J. M. Mendel, “Applications of Tipo-2 difuso logic systems to forecasting of time-series,” Inform. Sci., vol. 120, pp. 89-111, 1999.
- [8]. Mendez, G. M., Hernandez, M. A., “Modelling and Prediction of the MXNUSD Exchange Rate Using Hybrid RLS_BP Interval Tipo-1 Non-Singleton Tipo-2 Difuso Logic Systems”. Journal of Research in Computer Science 32:204-213, November 2007.
- [9]. Mendoza, O., Melin, P., Licea, G., “Difuso Inference Systems Tipo-1 and Tipo-2 for Digital Images Edge Detection”. Journal of Engineering Letters 15(1):45-52, August 2007.
- [10]. Melin, P., Urias, J., Solano, D., Soto, M., Lopez, M., Castillo, O., “Voice Recognition with Neural Networks, Tipo-2 Difuso Logic and Genetic Algorithms”. Journal of Engineering Letters 13(2):108-116, September 2006.
- [11]. D. Wu and J. M. Mendel, "A Vector Similarity Measure for Interval Tipo-2 Difuso Sets and Tipo-1 Difuso Sets," Information Sciences, 178, pp. 381-402, 2008.
- [12]. Hybrid learning algorithm for interval Tipo-2 difuso neural networks. Castro, J.R., Castillo, O., Melin, P., Rodriguez-Diaz, A.; In Proceedings of Granular Computing 2007, Silicon Valley, Ca, USA,. ISBN: 0-7695-3032-X ,pp 157-162, November 2007.
- [13]. Lee, C.-H, Lin, Y.-C, “Tipo-2 Difuso Neuro System Via Input-to-State-Stability Approach”. LECTURE NOTES IN COMPUTER SCIENCE (4492):317-327, 2007.

- [14]. A Hybrid Learning Algorithm for Interval Tipo-2 Difuso Neural Networks: The Case of Time Series Prediction. Castro, J. R., Castillo, O., Melin, P., Rodriguez-Diaz, A.; In Soft Computing for Hybrid Intelligent Systems. Book Chapter, Volume 154/2008: 363-386,. Springer Berlin / Heidelberg. ISBN: 978-3-540-70811-7, ISSN: 1860-949X (Print) 1860-9503 (Online), September 2008.
- [15].Mendez, G. M., "Interval Tipo-2 ANFIS". Advances in Soft Computing – Innovations in Hybrid Intelligent Systems; pp 64-71, 2007.
- [16].R. I. John and C. Czarnecki, "A type 2 adaptive difuso inferencing system," in Proc. IEEE Systems, Man and Cybernetics, pp. 2068-2073, 1998.
- [17].Lee, C.-H, Lin, Y.-C, "Tipo-2 Difuso Neuro System via Input-to-State-Stability Approach". LECTURE NOTES IN COMPUTER SCIENCE, pp. 317-327, 2007.
- [18].Yu-Ching Lin and Ching-Hung Lee, "System Identification and Adaptive Filter Using a Novel Difuso Neuro System", INTERNATIONAL JOURNAL OF COMPUTATIONAL COGNITION, VOL. 5, NO. 1, MARCH 2007.
- [19]. Castillo, O., Melin, P., "Comparison of Hybrid Intelligent Systems, Neural Networks, and Interval Tipo-2 Difuso logic for Time Series Prediction". In Jennie Si, editor, Proc. IJCNN-IEEE 2007, Orlando, USA, August 2007.
- [20]. Mendoza, O., Melin, P., Castillo, O., Licea, G., "Tipo-2 Difuso Systems as a Method for Improving Training Data and Decision Making in Modular Neural Networks for Pattern Recognition". International Journal of Information Technology and Intelligent Computing Vol. 1, Num 4, :pp 1-10, December 2006.
- [21]. Castillo, O., Martinez, A. I., Martinez, A. C., "Evolutionary Computing for Topology Optimization of Tipo-2 Difuso Systems". In Patricia Melin et al, editor, Analysis and Design of Intelligent Systems using Soft Computing Techniques, Number 1 in Studies in Fuzziness and Soft Computing, , Springer-Verlag, Germany, Vol 1, Num. 7, pp. 63-75,June 2007.
- [22]. Melin, P., Urias, J., Solano, D., Soto, M., Lopez, M., Castillo, O., "Voice Recognition with Neural Networks, Tipo-2 Difuso Logic and Genetic Algorithms". Journal of Engineering Letters, Vol. 13, Num 2, pp. 108-116, September 2006.
- [23]. J.S.R. Jang, C.T. Sun, and E. Mizutani, Neuro-difuso and Soft Computing. New York: Prentice-Hall, 1997.
- [24]. Castillo, O., Melin, P., "Comparison of Hybrid Intelligent Systems, Neural Networks, and Interval Tipo-2 Difuso logic for Time Series Prediction". In Jennie Si, editor, Proc. IJCNN-IEEE 2007, Orlando, USA, August 2007.
- [25]. L. A. Zadeh, "Difuso Logic, Neural Neural Networks and Soft Computing"
- [26]. Funobashi M., Moeda A., Morooka Y., Mori K. Difuso and Neural Hybrid Expert Systems: Sinergetic AI. - AI in Japan, IEEE, pp. 33-40, August, 1995.

- [27]. M. Bishop, Neural networks for pattern recognition. United Kingdom: Clarendon Press, chapter 5-7, 1995.
- [28]. Syed Muhammad Aqil Burney, Tahseen Ahmed Jilani, Cemal Ardil, "Approximate Bounded Knowledge Extraction Using Type-I Difuso Logic", Proceedings of world academy of science, Engineering and Technology Vol. 7, ISSN 1307-6884, August 2005.
- [29]. Aqil Burney S.M., Jilani A. Tahseen, "Time Series forecasting using artificial neural network methods for Karachi Stock Exchange", A Project in the Dept. of Computer Science, University of Karachi. 2002.
- [30]. F. Scarselli and A. C. Tosi, "Universal approximation using feed forward neural networks: A survey of some existing methods, and some new results," Neural Networks, Vol. 11, no. 1, pp. 15-37, 1998.
- [31]. Puha, P. K. H. Daohua Ming, "Parallel nonlinear optimization techniques for training neural networks.", IEEE Trans. on Neural Networks, Vol. 14, no. 6, pp 1460-1468, 2003.
- [32]. Randy L. Haupt and Sue Ellen Haupt, Practical Genetic Algorithms, Second Edition, John Wiley & Sons, Inc, .ISBN 0-471-45565-2 , 2004.
- [33]. Sh. Chang, S. Greenberg, Difuso Measures and Integrals: Theory and Applications, Physica-Verlag, NY, pp.415-34, 2003.
- [34]. P. Cunningham, Overfitting and Diversity in Classification Ensembles based on Feature Selection, TCD Computer Science Technical Report, TCD-CS-2000-07.
- [35]. J. Urias, D. Solano, .M. Soto, M. Lopez, P. Melín, "Tipo-2 Difuso Logic as a Method of Response Integration in Modular Neural Networks". IC-AI 2006, pp 584-590, 2006.
- [36]. F. Hoffmann, Evolutionary Algorithms for Difuso Control System Design, Proceedings of the IEEE, special issue on Industrial Applications of Soft Computing, Oct. 2001.
- [37]. D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman and A. K. Jain, "FVC2004: Third Fingerprint Verification Competition", Proc. International Conference on Biometric Authentication (ICBA), Hong Kong, pp. 1-7, July 2004.
- [38]. D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, The full FVC2000 and FVC2002 databases are available in the DVD included in: Handbook of Fingerprint Recognition, Springer, New York, 2003.
- [39]. Zadeh, L.A., "Difuso Logic", Computer, Vol. 1, No. 4, pp. 83-93, 1998.
- [40]. MATLAB Trade Marks, ©1994-2007 by the MathWorks, Inc.
- [41]. J. R. Castro, O. Castillo, P. Melin, L.G. Martinez, S. Escobar, I. Camacho, Building Difuso Inference Systems with Interval Tipo-2 Difuso Logic Toolbox Number 1 in Studies in Fuzziness and Soft Computing, Vol. 6, Num. 1, pp. 53-62, Springer-Verlag, Germany, June 2007.

- [42]. J. R. Castro, O. Castillo, P. Melin, An Interval Tipo-2 Difuso Logic Toolbox for Control Applications, In Proc. FUZZ-IEEE 2007, 2007.
- [43]. Felix Krahmer_, Youzuo Lin, Bonnie McAdoo, Katharine Ott, Jiakou Wang, David Widemannk, Brendt Wohlberg, Blind Image Deconvolution: Motion Blur Estimation, August 2006.
- [44]. Tuan Q. Pham, Lucas J. van Vliet, Klammer Schutte, Influence of signal-to-noise ratio and point spread function on limits of super resolution, Image Processing, Algorithms and Systems IV, Vol. 5672, pp. 169-180, 2005.
- [45]. Yue Lu and Chew Lim Tan, Combination of multiple classifiers using probabilistic dictionary and its application to postcode recognition, Pattern Recognition, Vol. 35, pp. 2823-2832, 2002.
- [46]. Mohamed Mostafa Abd Allah, "Artificial Neural Networks Fingerprints Authentication with Clusters Algorithm", Informatica, Vol. 29, pp. 303-307, 2005.
- [47]. H. Nemmour, Y. Chibani, "Neural Network Combination by Difuso Integral for Robust Change Detection in Remotely Sensed Imagery ",EURASIP Journal on Applied Signal Processing, Vol.14, pp. 2187-2195,2005.
- [48]. Sh. Chang, S. Greenberg, Difuso Measures and Integrals: Theory and Applications, pp.415-34 Physica-Verlag, NY.
- [49]. M. Grabisch, T. Murofushi, and M. Sugeno Difuso Measures and Integrals: Theory and Applications, pp.348-73. Physica-Verlag, NY.
- [50]. J. Keller, P. Gader, and A. Hocaoglu Difuso integrals in image processing and recognition. In M. Grabisch, et al., eds., Difuso Measures and Integrals: Theory and Applications, pp.435-66, Physica-Verlag, NY, 2000.
- [51]. M. Grabisch , A new algorithm for identifying difuso measures and its application to pattern recognition, In Proc. of 4th IEEE Int. Conf. on Difuso Systems, pp.145-50, Yokohama, Japan, 1995 .
- [52]. Puha, P. K. H. Daohua Ming, "Parallel nonlinear optimization techniques for training neural networks.", IEEE Trans. on Neural Networks, vol. 14, no. 6, , pp 1460-1468, 2003.
- [53]. Moller, M. F., "A scaled conjugate gradient algorithm for fast supervised learning," Neural Networks, vol. 6, pp. 525-533, 1993.
- [54]. D. W. Opitz,, Feature Selection for Ensembles, Sixteenth National Conference on Artificial/ Intelligence AAAI, pp. 379-384, Orlando, FL, 1999.
- [55]. D. W. Opitz, and R. Maclin, Popular Ensemble Methods: An Empirical Study, Journal of Artificial Intelligence Research, Vol. 11, pp. 169-19, 1999.
- [56]. D. W. Opitz and J. W. Shavlik Generating accurate and diverse members of a neural network ensemble, ,in: Advances in Neural Information Processing Systems 8, D. S. Touretzky, M. Mozer,and M. Hasselmo, eds, MIT Press, Cambridge, MA, , pp.535-541, 1996.

- [57]. Tuan Q. Pham, Lucas J. van Vliet, Klammer Schutte, Influence of signal-to-noise ratio and point spread function on limits of super resolution, Image Processing, Algorithms and Systems IV, Vol. 5672, pp. 169-180, 2005.
- [58]. M. Lopez, P. Melin: Response integration in Ensemble Neural Networks using interval Tipo-2 Difuso logic. IJCNN, pp. 1503-1508, 2008.

APÉNDICE

Código de la función d entrenamiento para los tres módulos de la red neuronal ensemble.

```
function [Net,Imagen,DimensionImagen,P,T,tiempo,Definicion] =  
entrena(folderimagenes,x_inicial,x_final,y_inicial,y_final)  
  
% warning off all;  
warning on all;  
% clear all;  
clc;  
tiempo.Inicio = now;tiempo.InicioT=tiempo.Inicio;disp(datestr(tiempo.InicioT));  
% %Definicion de la Red  
    Definicion.metodo='trainscg';  
    Definicion.meta=0.001;  
    Definicion.mostrar=50;  
    Definicion.epocas=500;  
    Definicion.tazaaprendizaje=0.001;  
    Definicion.sigma=0.00005;  
    Definicion.E=100;  
    Definicion.FuncionDesemp='mse';  
  
Imagen={};  
numimg=0;  
    for j = y_inicial : y_final  
        for i = x_inicial : x_final  
            numimg = numimg + 1;  
%    Rostro ORL Original  
            imagenl = imread([folderimagenes '\BaseData\s' int2str(i) '\' int2str(j) '.tif']);  
            [Imagen{1}(:,numimg) Imagen{2}(:,numimg) Imagen{3}(:,numimg)  
DimensionImagen]=lee(imagenl(:,1));  
  
                %mod1            %mod2            %mod3  
  
        end;  
  
    end;  
  
tiempo.Final = now;  
  
fprintf('-->> Duracion Leer Imagen %s\n\n',datediff(tiempo.Inicio,tiempo.Final));  
  
% funcion "mapstd", reduce el tiempo de entrenamiento
```

```

[P.pn_mod1,P.mod1]=mapstd(double(Imagen{1}));
[P.pn_mod2, P.ps_mod2 ]=mapstd(double(Imagen{2}));
[P.pn_mod3, P.ps_mod3 ]=mapstd(double(Imagen{3}));
tiempo.Inicio = now;
k = i; % se usaron las fotos de i sujetos
m = j; % se usaron las primeras j imagenes de cada sujeto
T.t_mod1=[];
Salida=k;
for n=1:m;
    T.t_mod1=cat(2,T.t_mod1,eye(k));
end;
T.t_mod2=T.t_mod1;
T.t_mod3=T.t_mod1;
%
net{1}=newff(minmax(pn_frente),[2*(k+m),k+m,k],{'tansig','tansig','logsig','purelin'},metodo);%trainrp

```

```

Net{1}=newff(minmax(P.pn_frente),[2*(k+m),(k+m),Salida],{'tansig','tansig','logsig'},Definicion.metodo);

```

```

% Net{1}.LW{3,1}= Net{1}.LW{3,1}*0.05;
% Net{1}.b{3}= Net{1}.b{3}*0.05;
% Net{1}.performFcn= Definicion.FuncionDesemp;
Net{1}.trainParam.show= Definicion.mostrar;
Net{1}.trainParam.goal= Definicion.meta;
Net{1}.trainParam.epochs=Definicion.epocas;
Net{1}.trainParam.lr= Definicion.tazaaprendizaje;
% Net{1}.trainParam.sigma= Definicion.sigma;
% pf=[];
[Net{1},y,e,pf] = adapt(Net{1},P.pn_mod1,T.t_mod1,[]);% mse(e)
[Net{1},T.tr_frente]=train(Net{1},P.pn_mod1,T.t_mod1); %% Entrenamiento
tiempo.Final = now;
fprintf('-->> (Mod. 1) Duracion %s\n\n',datediff(tiempo.Inicio,tiempo.Final));

tiempo.Inicio = now;

```



```
Net{2}=newff(minmax(P.pn_ojos),[2*(k+m),k+m,Salida],{'tansig','tansig','logsig'},Definicion.metodo);
```

```
% % Net{2}.LW{3,1}= Net{2}.LW{3,1}*0.001;
% % Net{2}.b{3}= Net{2}.b{3}*0.001;
Net{2}.LW{3,1}=Net{1}.LW{3,1};
Net{2}.b{3}=Net{1}.b{3};
% Net{2}.performFcn= Definicion.FuncionDesemp;
Net{2}.trainParam=Net{1}.trainParam;
```

```
[Net{2},y,e,pf] = adapt(Net{2},P.pn_mod2,T.t_mod2,[]);% mse(e)
[Net{2},T.tr_mod2]=train(Net{2},P.pn_mod2,T.t_mod2); %% Entrenamiento
```

```
% for i=1:m*k
```

```
% a_mod2(:,i)=sim(net{2},pn_mod2(:,i));
```

```
% end
```

```
tiempo.Final = now;
```

```
fprintf('-->> (Mod. 2)Duracion %s\n\n',datediff(tiempo.Inicio,tiempo.Final));
```

```
tiempo.Inicio = now;
```

```
Net{3}=newff(minmax(P.pn_boca),[2*(k+m),k+m,Salida],{'tansig','tansig','logsig'},Definicion.metodo);
```

```
% % Net{3}.LW{3,1}= Net{3}.LW{3,1}*0.001;
```

```
% % Net{3}.b{3}= Net{3}.b{3}*0.001;
```

```
Net{3}.LW{3,1}=Net{1}.LW{3,1};
```

```
Net{3}.b{3}=Net{1}.b{3};
```

```
% Net{3}.performFcn= Definicion.FuncionDesemp;
```

```
Net{3}.trainParam=Net{1}.trainParam;
```

```
[Net{3},y,e,pf] = adapt(Net{3},P.pn_mod3,T.t_mod3,[]); %mse(e)
```

```
[Net{3},T.tr_mod3]=train(Net{3},P.pn_mod3,T.t_mod3); %%Entrenamiento
```

```
tiempo.Final = now;
```

```
tiempo.FinalT=tiempo.Final;
```

```
fprintf('-->> (Mod. 3)Duracion %s\n',datediff(tiempo.Inicio,tiempo.Final));
```

```
disp(strcat(datestr(tiempo.InicioT),13,datestr(tiempo.FinalT)));
```

```
fprintf('>>>> Duracion Total %s <<<<\n',datediff(tiempo.InicioT,tiempo.FinalT));
```

```
clear ('pf','i','j','e','y','n','imagenl');
```

```
save(['Entrenamientos\' num2str(numimg) ' imag - Suj (' int2str(k) ') imXsuj (' int2str(m) ')  
' Definicion.metodo ' (' datestr(tiempo.InicioT,'dd-mm-yy HH-MM-SS PM') ') ' '['  
datediff(tiempo.InicioT,tiempo.FinalT) ' ]' ] )
```

Programa para el Integrador de Respuesta de lógica difusa tipo-1.

```
function [answer] = reconocefis(Archivo,Imagen,Net,DimensionImagen)
warning off;
global ContadorErrores;
Figura(1);

% P.PS1=DimensionImagen.PS1;
% P.PS2=DimensionImagen.PS2;
% P.PS3=DimensionImagen.PS3;
global arriba_abajo;
global persona;
if (arriba_abajo~=1)
    arriba_abajo=1;
    pos=0;
else
    arriba_abajo=0;
    pos=2;
end;
[pathstr, name, ext, versn] = fileparts(Archivo);
x=strfind(Archivo,'\');
titulo=Archivo(x(1)+1:x(2)-1);
tic;
Inicio=now; %disp(datestr(Inicio));

rimagen=Imagen;
img = imread(Archivo);
img=img(:,:,1); % Elijo solo una dimension de las tres que tiene la imagen por omision

%Procedimiento para Redimensionar la imagen al tamaño utilizado para Entrenar
[alto ancho]=size(img);
% global DimensionImagen;
% DimensionImagen.ancho
if (ancho~=DimensionImagen.ancho)
    ancho=DimensionImagen.ancho;
    img=imresize(img,[alto,ancho],'bilinear');
end;
[alto ancho]=size(img);
if (alto~=DimensionImagen.lheight*3)
    alto=DimensionImagen.lheight*3;
    img=imresize(img,[alto,ancho],'bilinear');
end;
% [alto ancho]=size(img) %este solo lo utilizo para saber si esta bien la redimensionada
%Procedimiento para Redimensionar la imagen al tamaño utilizado para Entrenar

[rimagen{1}(:,1) rimagen{2}(:,1) rimagen{3}(:,1) DimensionImagen]=lee(img);
%frente          %ojos          %boca
```

```

subplot(2,2,pos+1);
imagesc(img);
title(strcat('Image #',name,32,'Input Person',titulo));

%pause;
%% Proceso para Desplegar
longitud=3;
[rimagen{1}(:,1) rimagen{2}(:,1) rimagen{3}(:,1)]=lee(img);

[pn_mod1,ps_mod1]=mapstd(double(rimagen{1}));
[pn_mod2,ps_mod2]=mapstd(double(rimagen{2}));
[pn_mod3,ps_mod3]=mapstd(double(rimagen{3}));

matriz(:,1)=sim(Net{1}, pn_mod1(:,1));
matriz(:,2)=sim(Net{2}, pn_mod2(:,1));
matriz(:,3)=sim(Net{3}, pn_mod3(:,1));

[W1,Answer1]=max(matriz(:,1)); % Se elige la activacion mas alta y se guarda en Answer,
y W para saber el peso de la misma
[W2,Answer2]=max(matriz(:,2));
[W3,Answer3]=max(matriz(:,3));

a=readfis('IRSDIFUSOMOD_20.fis');
resultado_difuso=evalfis([W1 W2 W3], a);

disp('Respuesta Sistema Difuso Typo-1:');disp(resultado_difuso);

if ((resultado_difuso>=0)&&(resultado_difuso<=0.333))

im=[Imagen{1}(:,Answer1) Imagen{2}(:,Answer2) Imagen{3}(:,Answer3)];
final=VectorAImagen(im,DimensionImagen);
subplot(2,2,pos+2);
imagesc(final); colormap(gray);

fprintf('Imagen Número: >>>%d\n',Answer1);
fprintf('Indice de: >>>>>>%f\n',W1);
disp(Answer1)
disp(titulo)
disp(resultado_difuso)
disp(Answer3)
disp(W3)
disp(Answer2)
disp(W2)
disp(Answer1)
disp(W1)
t='Tipo-1 Module1';
title(strcat(t,32,'Output Person',int2str(Answer1)));

```

```

elseif ((resultado_difuso>0.333)&&(resultado_difuso<=0.666))
im=[Imagen{1}{:},Answer1) Imagen{2}{:},Answer2) Imagen{3}{:},Answer3)];
final=VectorAImagen(im,DimensionImagen);
subplot(2,2,pos+2);
imagesc(final); colormap(gray);

fprintf('Imagen Número: >>>%d\n',Answer2);
fprintf('Indice de: >>>>>>%f\n',W2);
disp(Answer2)
disp(titulo)
disp(resultado_difuso)
disp(Answer3)
disp(W3)
disp(Answer2)
disp(W2)
disp(Answer1)
disp(W1)
t=' Tipo-1 Module2';
title(strcat(t,32,'Output Person',int2str(Answer2)));

elseif ((resultado_difuso>0.666)&&(resultado_difuso<=1))
im=[Imagen{1}{:},Answer1) Imagen{2}{:},Answer2) Imagen{3}{:},Answer3)];
final=VectorAImagen(im,DimensionImagen);
subplot(2,2,pos+2);
imagesc(final); colormap(gray);

fprintf('Imagen Número: >>>%d\n',Answer3);
fprintf('Indice de: >>>>>>%f\n',W3);
disp(Answer3)
disp(titulo)
disp(resultado_difuso)
disp(Answer3)
disp(W3)
disp(Answer2)
disp(W2)
disp(Answer1)
disp(W1)
t=' Tipo-1 Module3';
title(strcat(t,32,'Output Person',int2str(Answer3)));
elseif ContadorErrores==ContadorErrores+1;
end

```

Programa para el Integrador de Respuesta de lógica difusa tipo-2.

```
a=readfis('DifusoGaType2_20.fis');

% resultado_difuso=evalifistype2([Answer1 Answer2 Answer3], a);
resultado_difuso=evalifistype2([W1 W2 W3], a);

disp('Respuesta Sistema Difuso Tipo 2:');disp(resultado_difuso);

if ((resultado_difuso>=0)&&(resultado_difuso<=0.333))
im=[Imagen{1}(:,Answer1) Imagen{2}(:,Answer2) Imagen{3}(:,Answer3)];
final=VectorAImagen(im,DimensionImagen);
subplot(2,2,pos+2);
imagesc(final); colormap(gray);

    fprintf('Image Number: >>>%d\n',Answer1);
    fprintf('Indice de: >>>>>>%f\n',W1);

disp(titulo)
disp(resultado_difuso)
disp(Answer3)
disp(W3)
disp(Answer2)
disp(W2)
disp(Answer1)
disp(W1)

t=' Tipo-2 Module1';
title(strcat(t,32,'Output Person',int2str(Answer1)));

end

if ((resultado_difuso>0.333)&&(resultado_difuso<=0.666))
im=[Imagen{1}(:,Answer1) Imagen{2}(:,Answer2) Imagen{3}(:,Answer3)];
final=VectorAImagen(im,DimensionImagen);
subplot(2,2,pos+2);
imagesc(final); colormap(gray);

    fprintf('Image Number : >>>%d\n',Answer2);
    fprintf('Indice de: >>>>>>%f\n',W2);

disp(titulo)
disp(resultado_difuso)
disp(Answer3)
disp(W3)
```

```

disp(Answer2)
disp(W2)
disp(Answer1)
disp(W1)
t=' Tipo-2 Module2';
title(strcat(t,32,'Output Person',int2str(Answer2)));

end

if ((resultado_difuso>0.666)&&(resultado_difuso<=1))
im=[Imagen{1}(:,Answer1) Imagen{2}(:,Answer2) Imagen{3}(:,Answer3)];
final=VectorAImagen(im,DimensionImagen);
subplot(2,2,pos+2);
imagesc(final); colormap(gray);

fprintf('Image Number: >>>%d\n',Answer3);
fprintf('Indice de: >>>>>>%f\n',W3);
disp(titulo)
disp(resultado_difuso)
disp(Answer3)
disp(W3)
disp(Answer2)
disp(W2)
disp(Answer1)
disp(W1)
t=' Tipo-2 Module3';
title(strcat(t,32,'Output Person',int2str(Answer3)));

end

```

Programa principal de Reconocimiento donde el usuario selecciona el método de integración de respuesta que desea utilizar.

% Reconoce Todos

function [x] = reconocetodos(Imagen,Net,DimensionImagen,x_inicial,x_final,y_inicial,y_final)

global arriba;

global persona;

arriba=0;

cadena=[];

clf;

cont=0;

for x=x_inicial:x_final

persona=x;

for y=y_inicial:y_final

cont=cont+1;

%Rostro ORL Original

cadena=['BaseDatosGaussian10\' num2str(x) \' \' num2str(y) '.tif'];

% cadena=['BaseDatos\'s' num2str(x) \'small\' num2str(y) '.bmp'];

% Huellas

% cadena=['BaseDatosCompleta\'s' num2str(x) \'small\' num2str(y) '.png'];

% [a]=reconoce(cadena,Imagen,Net,DimensionImagen);

% reconocesugeno(cadena,Imagen,Net,DimensionImagen);

% reconocefis(cadena,Imagen,Net,DimensionImagen);

% reconocefis2(cadena,Imagen,Net,DimensionImagen);

if (mod(cont,2)==0)

if not(isempty(cadena))

pause;

end;

clf;

end;

end;

end;

disp('Fin del ciclo de Reconocer');

Programa del Algoritmo genético para optimizar el numero de reglas del integrador de respuesta de lógica difusa tipo-1

warning off;

NIND = 80; % Número de individuos por subpoblaciones
MAXGEN = 30; % Maximo Numero de generaciones
GGAP = 0.5; % "Generation gap", Que porcentaje del total de la poblacion de individuos nuevos son creados
PRECI = 351; % Precisión de representaciones binarias

%leer sistema difuso
sdifuso = readfis('IRSDIFUSO.fis');

showrule(sdifuso,[1:27],'indexed')
%disp('evaluacion del sistema difuso [0,0]');evalfis([-1,0],sdifuso)
%disp('evaluacion del sistema difuso [0.5,0]');evalfis([0,-1],sdifuso)
%disp('evaluacion del sistema difuso [0,0.5]');evalfis([-1,0],sdifuso)

disp('evaluacion del sistema difuso [0,0,0]');evalfis([0,0,0],sdifuso)
disp('evaluacion del sistema difuso [0,0,1]');evalfis([0 0.1; 0 0.1; 0 0.1],sdifuso)
disp('evaluacion del sistema difuso [0.11,0.2]');evalfis([0.11 0.2; .11 0.2; .11 0.2],sdifuso)
disp('evaluacion del sistema difuso [.21,0.3]');evalfis([0.21 0.3; 0.21 0.3; 0.21 0.3],sdifuso)
disp('evaluacion del sistema difuso [0.31,0.35]');evalfis([0.31 0.35; 0.31 0.35; 0.31 0.35],sdifuso)
disp('evaluacion del sistema difuso [0.36,0.4]');evalfis([0.36 0.4; 0.36 0.4; 0.36 0.4],sdifuso)
disp('evaluacion del sistema difuso [0.4,0.5]');evalfis([0.4 0.5; 0.4 0.5; 0.4 0.5],sdifuso)
disp('evaluacion del sistema difuso [0.5,0.6]');evalfis([0.5 0.6; 0.5 0.6; 0.5 0.6],sdifuso)
disp('evaluacion del sistema difuso [0.6,0.65]');evalfis([0.6 0.65; 0.6 0.65; 0.6 0.65],sdifuso)
disp('evaluacion del sistema difuso [0.65,0.75]');evalfis([0.65 0.75; 0.65 0.75; 0.65 0.75],sdifuso)
disp('evaluacion del sistema difuso [0.75,0.8]');evalfis([0.75 0.8; 0.75 0.8; 0.75 0.8],sdifuso)
disp('evaluacion del sistema difuso [.8,0.9]');evalfis([0.75 0.8; 0.75 0.8; 0.75 0.8],sdifuso)
disp('evaluacion del sistema difuso [0.9,1]');evalfis([0.75 0.8; 0.75 0.8; 0.75 0.8],sdifuso)

pause; %disp('presiona una tecla para continuar');

% Inicializa población

Chrom = crtbp(NIND, PRECI);
% Chrom

disp('presiona una tecla para continuar');
pause
% Borrar contadores(reinicializar a 0)
Best = NaN*ones(MAXGEN,1); % EL mejor de la población actual
gen = 0; % contador generacional

% Evaluar población inicial
ObjV = FuncionObjDifuso_1(Chrom, sdifuso);

```

% ObjV = objfun1(Chrom, sdifuso); % ya hice este archivo
showrule(sdifuso,[1:27],'indexed')

%disp(' ');

% Tomar el mejor individuo y desplegar convergencia
Best(gen+1) = min(ObjV);
% pause % PRESIONA una tecla para desplegar el resultado...
clf %limpia la gráfica.

subplot(212);plot(log10(Best),'ro');xlabel('generacion'); ylabel('log10(f(x))');
text(0.5,0.95,['El mejor = ', num2str(Best(gen+1))],'Units','normalized');
drawnow;

% Ciclo Generacional
while gen < MAXGEN,

    % Asignación de valor aptitud para la población entera
    ObjV = ObjV'
    FitnV = ranking(ObjV);
    ObjV = ObjV'

    % Seleccionar individuos para separarlos utilizando método de ruleta (rulette wheel)
    SelCh = select('rws', Chrom, FitnV, GGAP);

    % Recombinar individuos seleccionados (crossover multi point)
    SelCh = recomb('xovmp',SelCh,0.5);

    % Realizar mutación en offspring
    SelCh = mut(SelCh);
    %SelCh = inversionMutation (SelCh);
    % Evaluar offspring, llamar función objetivo
    ObjVSel = FuncionObjDifuso_1(SelCh, sdifuso);
%    ObjV = objfun1(Chrom, sdifuso); % ya hice este archivo

    % Reinsertar offspring a la población actual
    ObjVSel =ObjVSel'
    ObjV = ObjV'
    [Chrom ObjV]=reins(Chrom,SelCh,1,1,ObjV,ObjVSel);
    ObjVSel =ObjVSel'
    ObjV = ObjV'

    % Incrementar contador generacional
    gen = gen+1;

    % Actualizar Despliegue y registrar el mejor individuo actual
    Best(gen+1) = min(ObjV);

```

```
subplot(2,1,2);plot(log10(Best),'ro'); xlabel('Generación'); ylabel('log10(f(x))');  
text(0.5,0.95,['El mejor = ', num2str(Best(gen+1))], 'Units','normalized');  
drawnow;  
end  
pause, '% PRESIONA una tecla para continuar...';
```

Chrom

% Fin de GA

Programa Cromosoma del Algoritmo Genetico para el integrador de respuesta de lógica difusa tipo-1

%programa que evalua la función objetivo

```
function [ObjV] = FuncionObjDifuso351_27(Chrom, sdifuso)
```

```
    echo off
    [Nind t]= size(Chrom);

    tempdifuso = sdifuso
    for i=1:1:Nind
        % disp('declarando reglas');pause;
        regla1(i) =
            bin2real(Chrom(i,28),Chrom(i,29),Chrom(i,30))+bin2real(Chrom(i,31),Chrom(i,32),Chrom(i,33))+
            bin2real(Chrom(i,34),Chrom(i,35),Chrom(i,36))+bin2real(Chrom(i,37),Chrom(i,38),Chrom(i,39));
        regla2(i) = bin2real(Chrom(i,40),Chrom(i,41),Chrom(i,42))+
            bin2real(Chrom(i,43),Chrom(i,44),Chrom(i,45))+bin2real(Chrom(i,46),Chrom(i,47),Chrom(i,48))+
            bin2real(Chrom(i,49),Chrom(i,50),Chrom(i,51));
        regla3(i) =
            bin2real(Chrom(i,52),Chrom(i,53),Chrom(i,54)),bin2real(Chrom(i,55),Chrom(i,56),Chrom(i,57)),bi
            n2real(Chrom(i,58),Chrom(i,59),Chrom(i,60))+bin2real(Chrom(i,61),Chrom(i,62),Chrom(i,63));
        regla4(i) =
            bin2real(Chrom(i,64),Chrom(i,65),Chrom(i,66),bin2real(Chrom(i,67),Chrom(i,68),Chrom(i,69)),bi
            n2real(Chrom(i,70),Chrom(i,71),Chrom(i,72))+bin2real(Chrom(i,73),Chrom(i,74),Chrom(i,75));
        regla5(i) =
            bin2real(Chrom(i,76),Chrom(i,77),Chrom(i,78)),bin2real(Chrom(i,79),Chrom(i,80),Chrom(i,81)),bi
            n2real(Chrom(i,82),Chrom(i,83),Chrom(i,84))+bin2real(Chrom(i,85),Chrom(i,86),Chrom(i,87));
        regla6(i) =
            bin2real(Chrom(i,88),Chrom(i,89),Chrom(i,90)),bin2real(Chrom(i,91),Chrom(i,92),Chrom(i,93)),bi
            n2real(Chrom(i,94),Chrom(i,95),Chrom(i,96))+bin2real(Chrom(i,97),Chrom(i,98),Chrom(i,99));
        regla7(i) =
            bin2real(Chrom(i,100),Chrom(i,101),Chrom(i,102)),bin2real(Chrom(i,103),Chrom(i,104),Chrom(i,
            105)),bin2real(Chrom(i,106),Chrom(i,107),Chrom(i,108))+bin2real(Chrom(i,109),Chrom(i,110),Ch
            rom(i,111));
        regla8(i) =
            bin2real(Chrom(i,112),Chrom(i,113),Chrom(i,114)),bin2real(Chrom(i,115),Chrom(i,116),Chrom(i,
            117)),bin2real(Chrom(i,118),Chrom(i,119),Chrom(i,120))+bin2real(Chrom(i,121),Chrom(i,122),Ch
            rom(i,123));
        regla9(i) =
            bin2real(Chrom(i,124),Chrom(i,125),Chrom(i,126)),bin2real(Chrom(i,127),Chrom(i,128),Chrom(i,
            129)),bin2real(Chrom(i,130),Chrom(i,131),Chrom(i,132))+bin2real(Chrom(i,133),Chrom(i,134),Ch
            rom(i,135));
        regla10(i) =
            bin2real(Chrom(i,136),Chrom(i,137),Chrom(i,138)),bin2real(Chrom(i,139),Chrom(i,140),Chrom(i,
            141)),bin2real(Chrom(i,142),Chrom(i,143),Chrom(i,144))+bin2real(Chrom(i,145),Chrom(i,146),Ch
            rom(i,147));
        regla11(i) =
            bin2real(Chrom(i,148),Chrom(i,149),Chrom(i,150)),bin2real(Chrom(i,151),Chrom(i,152),Chrom(i,
```

```

153)),bin2real(Chrom(i,154),Chrom(i,155),Chrom(i,156))+bin2real(Chrom(i,157),Chrom(i,158),Chrom(i,159));
    regla12(i) =
bin2real(Chrom(i,160),Chrom(i,161),Chrom(i,162)),bin2real(Chrom(i,163),Chrom(i,164),Chrom(i,165)),bin2real(Chrom(i,166),Chrom(i,167),Chrom(i,168))+bin2real(Chrom(i,169),Chrom(i,170),Chrom(i,171));
    regla13(i) =
bin2real(Chrom(i,172),Chrom(i,173),Chrom(i,174))+bin2real(Chrom(i,175),Chrom(i,176),Chrom(i,177))+bin2real(Chrom(i,178),Chrom(i,179),Chrom(i,180))+bin2real(Chrom(i,181),Chrom(i,182),Chrom(i,183));
    regla14(i) = bin2real(Chrom(i,184),Chrom(i,185),Chrom(i,186))+
bin2real(Chrom(i,187),Chrom(i,188),Chrom(i,189))+bin2real(Chrom(i,190),Chrom(i,191),Chrom(i,192))+bin2real(Chrom(i,193),Chrom(i,194),Chrom(i,195));
    regla15(i) =
bin2real(Chrom(i,196),Chrom(i,197),Chrom(i,198)),bin2real(Chrom(i,199),Chrom(i,200),Chrom(i,201)),bin2real(Chrom(i,202),Chrom(i,203),Chrom(i,204))+bin2real(Chrom(i,205),Chrom(i,206),Chrom(i,207));
    regla16(i) =
bin2real(Chrom(i,208),Chrom(i,209),Chrom(i,210)),bin2real(Chrom(i,211),Chrom(i,212),Chrom(i,213)),bin2real(Chrom(i,214),Chrom(i,215),Chrom(i,216))+bin2real(Chrom(i,217),Chrom(i,218),Chrom(i,219));
    regla17(i) =
bin2real(Chrom(i,220),Chrom(i,221),Chrom(i,222)),bin2real(Chrom(i,223),Chrom(i,224),Chrom(i,225)),bin2real(Chrom(i,226),Chrom(i,227),Chrom(i,228))+bin2real(Chrom(i,229),Chrom(i,230),Chrom(i,231));
    regla18(i) =
bin2real(Chrom(i,232),Chrom(i,233),Chrom(i,234)),bin2real(Chrom(i,235),Chrom(i,236),Chrom(i,237)),bin2real(Chrom(i,238),Chrom(i,239),Chrom(i,240))+bin2real(Chrom(i,241),Chrom(i,242),Chrom(i,243));
    regla19(i) =
bin2real(Chrom(i,244),Chrom(i,245),Chrom(i,246)),bin2real(Chrom(i,247),Chrom(i,248),Chrom(i,249)),bin2real(Chrom(i,250),Chrom(i,251),Chrom(i,252))+bin2real(Chrom(i,253),Chrom(i,254),Chrom(i,255));
    regla20(i) =
bin2real(Chrom(i,256),Chrom(i,257),Chrom(i,258)),bin2real(Chrom(i,259),Chrom(i,260),Chrom(i,261)),bin2real(Chrom(i,262),Chrom(i,263),Chrom(i,264))+bin2real(Chrom(i,265),Chrom(i,266),Chrom(i,267));
    regla21(i) =
bin2real(Chrom(i,268),Chrom(i,269),Chrom(i,270)),bin2real(Chrom(i,271),Chrom(i,272),Chrom(i,273)),bin2real(Chrom(i,274),Chrom(i,275),Chrom(i,276))+bin2real(Chrom(i,277),Chrom(i,278),Chrom(i,279));
    regla22(i) =
bin2real(Chrom(i,280),Chrom(i,281),Chrom(i,282)),bin2real(Chrom(i,283),Chrom(i,284),Chrom(i,285)),bin2real(Chrom(i,286),Chrom(i,287),Chrom(i,288))+bin2real(Chrom(i,289),Chrom(i,290),Chrom(i,291));
    regla23(i) =
bin2real(Chrom(i,292),Chrom(i,293),Chrom(i,294)),bin2real(Chrom(i,295),Chrom(i,296),Chrom(i,297)),bin2real(Chrom(i,298),Chrom(i,299),Chrom(i,300))+bin2real(Chrom(i,301),Chrom(i,302),Chrom(i,303));
    regla24(i) =
bin2real(Chrom(i,304),Chrom(i,305),Chrom(i,306)),bin2real(Chrom(i,307),Chrom(i,308),Chrom(i,

```

```

309)),bin2real(Chrom(i,310),Chrom(i,311),Chrom(i,312))+bin2real(Chrom(i,313),Chrom(i,314),Chrom(i,315));
    regla25(i) =
bin2real(Chrom(i,316),Chrom(i,317),Chrom(i,318)),bin2real(Chrom(i,319),Chrom(i,320),Chrom(i,321)),bin2real(Chrom(i,322),Chrom(i,323),Chrom(i,324))+bin2real(Chrom(i,325),Chrom(i,326),Chrom(i,327));
    regla26(i) =
bin2real(Chrom(i,328),Chrom(i,329),Chrom(i,330)),bin2real(Chrom(i,331),Chrom(i,332),Chrom(i,333)),bin2real(Chrom(i,334),Chrom(i,335),Chrom(i,336))+bin2real(Chrom(i,337),Chrom(i,338),Chrom(i,339));
    regla27(i) =
bin2real(Chrom(i,340),Chrom(i,341),Chrom(i,342)),bin2real(Chrom(i,343),Chrom(i,344),Chrom(i,345)),bin2real(Chrom(i,346),Chrom(i,347),Chrom(i,348))+bin2real(Chrom(i,349),Chrom(i,350),Chrom(i,351));

```

```

% disp('termina declarado reglas');pause;

```

```

end

```

Programa Cromosoma del Algoritmo Genetico para el integrador de respuesta de lógica difusa tipo-2

%programa que evalua la función objetivo

```
function [ObjV] = FuncionObjDifuso351_27(Chrom, sdifuso)
```

```
    echo off
    [Nind t]= size(Chrom);

    tempdifuso = sdifuso
    for i=1:1:Nind
        % disp('declarando reglas');pause;
        regla1(i) =
        bin2real(Chrom(i,28),Chrom(i,29),Chrom(i,30))+bin2real(Chrom(i,31),Chrom(i,32),Chrom(i,33))+
        bin2real(Chrom(i,34),Chrom(i,35),Chrom(i,36))+bin2real(Chrom(i,37),Chrom(i,38),Chrom(i,39));
        regla2(i) = bin2real(Chrom(i,40),Chrom(i,41),Chrom(i,42))+
        bin2real(Chrom(i,43),Chrom(i,44),Chrom(i,45))+bin2real(Chrom(i,46),Chrom(i,47),Chrom(i,48))+
        bin2real(Chrom(i,49),Chrom(i,50),Chrom(i,51));
        regla3(i) =
        bin2real(Chrom(i,52),Chrom(i,53),Chrom(i,54)),bin2real(Chrom(i,55),Chrom(i,56),Chrom(i,57)),bi
        n2real(Chrom(i,58),Chrom(i,59),Chrom(i,60))+bin2real(Chrom(i,61),Chrom(i,62),Chrom(i,63));
        regla4(i) =
        bin2real(Chrom(i,64),Chrom(i,65),Chrom(i,66),bin2real(Chrom(i,67),Chrom(i,68),Chrom(i,69)),bi
        n2real(Chrom(i,70),Chrom(i,71),Chrom(i,72))+bin2real(Chrom(i,73),Chrom(i,74),Chrom(i,75));
        regla5(i) =
        bin2real(Chrom(i,76),Chrom(i,77),Chrom(i,78)),bin2real(Chrom(i,79),Chrom(i,80),Chrom(i,81)),bi
        n2real(Chrom(i,82),Chrom(i,83),Chrom(i,84))+bin2real(Chrom(i,85),Chrom(i,86),Chrom(i,87));
        regla6(i) =
        bin2real(Chrom(i,88),Chrom(i,89),Chrom(i,90)),bin2real(Chrom(i,91),Chrom(i,92),Chrom(i,93)),bi
        n2real(Chrom(i,94),Chrom(i,95),Chrom(i,96))+bin2real(Chrom(i,97),Chrom(i,98),Chrom(i,99));
        regla7(i) =
        bin2real(Chrom(i,100),Chrom(i,101),Chrom(i,102)),bin2real(Chrom(i,103),Chrom(i,104),Chrom(i,
        105)),bin2real(Chrom(i,106),Chrom(i,107),Chrom(i,108))+bin2real(Chrom(i,109),Chrom(i,110),Ch
        rom(i,111));
        regla8(i) =
        bin2real(Chrom(i,112),Chrom(i,113),Chrom(i,114)),bin2real(Chrom(i,115),Chrom(i,116),Chrom(i,
        117)),bin2real(Chrom(i,118),Chrom(i,119),Chrom(i,120))+bin2real(Chrom(i,121),Chrom(i,122),Ch
        rom(i,123));
        regla9(i) =
        bin2real(Chrom(i,124),Chrom(i,125),Chrom(i,126)),bin2real(Chrom(i,127),Chrom(i,128),Chrom(i,
        129)),bin2real(Chrom(i,130),Chrom(i,131),Chrom(i,132))+bin2real(Chrom(i,133),Chrom(i,134),Ch
        rom(i,135));
        regla10(i) =
        bin2real(Chrom(i,136),Chrom(i,137),Chrom(i,138)),bin2real(Chrom(i,139),Chrom(i,140),Chrom(i,
        141)),bin2real(Chrom(i,142),Chrom(i,143),Chrom(i,144))+bin2real(Chrom(i,145),Chrom(i,146),Ch
        rom(i,147));
        regla11(i) =
        bin2real(Chrom(i,148),Chrom(i,149),Chrom(i,150)),bin2real(Chrom(i,151),Chrom(i,152),Chrom(i,
```

```

153)),bin2real(Chrom(i,154),Chrom(i,155),Chrom(i,156))+bin2real(Chrom(i,157),Chrom(i,158),Chrom(i,159));
    regla12(i) =
bin2real(Chrom(i,160),Chrom(i,161),Chrom(i,162)),bin2real(Chrom(i,163),Chrom(i,164),Chrom(i,165)),bin2real(Chrom(i,166),Chrom(i,167),Chrom(i,168))+bin2real(Chrom(i,169),Chrom(i,170),Chrom(i,171));
    regla13(i) =
bin2real(Chrom(i,172),Chrom(i,173),Chrom(i,174))+bin2real(Chrom(i,175),Chrom(i,176),Chrom(i,177))+bin2real(Chrom(i,178),Chrom(i,179),Chrom(i,180))+bin2real(Chrom(i,181),Chrom(i,182),Chrom(i,183));
    regla14(i) = bin2real(Chrom(i,184),Chrom(i,185),Chrom(i,186))+
bin2real(Chrom(i,187),Chrom(i,188),Chrom(i,189))+bin2real(Chrom(i,190),Chrom(i,191),Chrom(i,192))+bin2real(Chrom(i,193),Chrom(i,194),Chrom(i,195));
    regla15(i) =
bin2real(Chrom(i,196),Chrom(i,197),Chrom(i,198)),bin2real(Chrom(i,199),Chrom(i,200),Chrom(i,201)),bin2real(Chrom(i,202),Chrom(i,203),Chrom(i,204))+bin2real(Chrom(i,205),Chrom(i,206),Chrom(i,207));
    regla16(i) =
bin2real(Chrom(i,208),Chrom(i,209),Chrom(i,210)),bin2real(Chrom(i,211),Chrom(i,212),Chrom(i,213)),bin2real(Chrom(i,214),Chrom(i,215),Chrom(i,216))+bin2real(Chrom(i,217),Chrom(i,218),Chrom(i,219));
    regla17(i) =
bin2real(Chrom(i,220),Chrom(i,221),Chrom(i,222)),bin2real(Chrom(i,223),Chrom(i,224),Chrom(i,225)),bin2real(Chrom(i,226),Chrom(i,227),Chrom(i,228))+bin2real(Chrom(i,229),Chrom(i,230),Chrom(i,231));
    regla18(i) =
bin2real(Chrom(i,232),Chrom(i,233),Chrom(i,234)),bin2real(Chrom(i,235),Chrom(i,236),Chrom(i,237)),bin2real(Chrom(i,238),Chrom(i,239),Chrom(i,240))+bin2real(Chrom(i,241),Chrom(i,242),Chrom(i,243));
    regla19(i) =
bin2real(Chrom(i,244),Chrom(i,245),Chrom(i,246)),bin2real(Chrom(i,247),Chrom(i,248),Chrom(i,249)),bin2real(Chrom(i,250),Chrom(i,251),Chrom(i,252))+bin2real(Chrom(i,253),Chrom(i,254),Chrom(i,255));
    regla20(i) =
bin2real(Chrom(i,256),Chrom(i,257),Chrom(i,258)),bin2real(Chrom(i,259),Chrom(i,260),Chrom(i,261)),bin2real(Chrom(i,262),Chrom(i,263),Chrom(i,264))+bin2real(Chrom(i,265),Chrom(i,266),Chrom(i,267));
    regla21(i) =
bin2real(Chrom(i,268),Chrom(i,269),Chrom(i,270)),bin2real(Chrom(i,271),Chrom(i,272),Chrom(i,273)),bin2real(Chrom(i,274),Chrom(i,275),Chrom(i,276))+bin2real(Chrom(i,277),Chrom(i,278),Chrom(i,279));
    regla22(i) =
bin2real(Chrom(i,280),Chrom(i,281),Chrom(i,282)),bin2real(Chrom(i,283),Chrom(i,284),Chrom(i,285)),bin2real(Chrom(i,286),Chrom(i,287),Chrom(i,288))+bin2real(Chrom(i,289),Chrom(i,290),Chrom(i,291));
    regla23(i) =
bin2real(Chrom(i,292),Chrom(i,293),Chrom(i,294)),bin2real(Chrom(i,295),Chrom(i,296),Chrom(i,297)),bin2real(Chrom(i,298),Chrom(i,299),Chrom(i,300))+bin2real(Chrom(i,301),Chrom(i,302),Chrom(i,303));
    regla24(i) =
bin2real(Chrom(i,304),Chrom(i,305),Chrom(i,306)),bin2real(Chrom(i,307),Chrom(i,308),Chrom(i,

```



```

309)),bin2real(Chrom(i,310),Chrom(i,311),Chrom(i,312))+bin2real(Chrom(i,313),Chrom(i,314),Chrom(i,315));
    regla25(i) =
bin2real(Chrom(i,316),Chrom(i,317),Chrom(i,318)),bin2real(Chrom(i,319),Chrom(i,320),Chrom(i,321)),bin2real(Chrom(i,322),Chrom(i,323),Chrom(i,324))+bin2real(Chrom(i,325),Chrom(i,326),Chrom(i,327));
    regla26(i) =
bin2real(Chrom(i,328),Chrom(i,329),Chrom(i,330)),bin2real(Chrom(i,331),Chrom(i,332),Chrom(i,333)),bin2real(Chrom(i,334),Chrom(i,335),Chrom(i,336))+bin2real(Chrom(i,337),Chrom(i,338),Chrom(i,339));
    regla27(i) =
bin2real(Chrom(i,340),Chrom(i,341),Chrom(i,342)),bin2real(Chrom(i,343),Chrom(i,344),Chrom(i,345)),bin2real(Chrom(i,346),Chrom(i,347),Chrom(i,348))+bin2real(Chrom(i,349),Chrom(i,350),Chrom(i,351));

```

```

fprintf('terminando las reglas elseif');
showrule(tempdifuso,[1:numoreglas],'indexed')

```

```

%presiona una tecla para continuar
% echo off

```

```

%s prueba el sistema difuso
for j=-1:0.5:1
    for t=-1:0.5:1
        evalua = evalfis([j,t],tempdifuso);
        error(i+1)= ((0.001)-(evalua*-1))^2;
    end
end

```

```

subplot(1,2,1); plot(evalua,'ro');xlabel('Numero de individuo'); ylabel('Error');
text(0.5,0.95,['El Error = ', num2str(error(i+1))],'Units','normalized');
set(gcf, 'name', 'evalua', 'numbertitle', 'off');
drawnow;

```

```

end
end
% pause;
%le mido el error acumulado
% error = ((evalua)-(0.00761))^2;
error(i+1)= ((0.0001)-(evalua*-1))^2;

```

```

subplot(2,1,1); plot(error,'ro');xlabel('Numero de individuo'); ylabel('Error');
text(0.5,0.95,['El Error = ', num2str(error(i+1))],'Units','normalized');
set(gcf, 'name', 'Funcion Objetivo', 'numbertitle', 'off');
drawnow;

```

```

%x = -pi:pi/10:pi;
%y = tan(sin(x)) - sin(tan(x));
%plot(x,y,'--rs','LineWidth',2,...
%      'MarkerEdgeColor','k',...
%      'MarkerFaceColor','g',...
%      'MarkerSize',10)

```

```
%ObjV(i) = sumar (error*100) y el numero de reglas
ObjV(i) = numeroreglas + error(i+1);

fprintf(' evaluado tempdifuso');
showrule(tempdifuso,[1:numeroreglas],'indexed')
disp(error)

%ruleview, gensurf
% showrule(tempdifuso,[1:numeroreglas],'verbose')
% numeroreglas
end
```