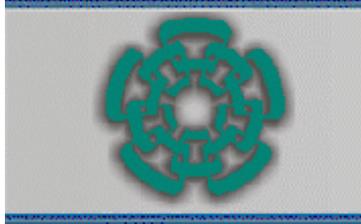


Centro de Investigación y Estudios Avanzados del I.P.N
Departamento de Ingeniería Eléctrica
Sección Computación



Reconocimiento de imágenes mediante redes
neuronales artificiales aplicado a la iridología

Tesis presentada por:
Ana Eugenia Romo González
para obtener el grado de Maestro en Ciencias en
Ingeniería Eléctrica
con opción en Computación
Febrero de 2005

Tesis dirigida por:
Dr. Feliú Davino Sagols Troncoso
CINVESTAV-IPN México, adscrito al Departamento de
Matemáticas
(antes al Departamento de Ingeniería Eléctrica,
Sección Computación)

Contenido

Capítulo 1. Introducción	3
1.1 Distribución general	4
1.2 Soluciones actuales	4
1.3 Metodología utilizada	5
Capítulo 2. La iridología	8
2.1 Definición de lesiones en el iris.	10
2.2 Lesiones relacionadas con los intestinos	19
2.3 Lesiones abiertas y cerradas	21
2.4 Estado de las lesiones.	22
2.4.1 La crisis curativa o la ley de curación de Hering	23
2.4.2 Agudo	24
2.4.3 Sub-agudo	24
2.4.4 Crónico	25
2.4.5 Degenerativo	25
Capítulo 3. Fundamentos del procesamiento digital de imágenes	26
3.1 Imágenes digitales[11]	26
3.2 Representación morfológica y residual	27
3.3 Definiciones fundamentales para transformaciones morfológicas de imágenes.	27
3.4 Transformaciones Morfológicas.	28
3.5 Muestreo espacial y niveles de gris	29
3.6 Relaciones entre píxeles: vecindad, conexidad	30
3.7 Distancia.	32
3.8 Métodos de adelgazamiento	34
3.8.1 Algoritmos de adelgazamiento secuenciales.	37
3.9 Extracción del perímetro	39
Capítulo 4. Filtrado de imágenes	40
4.1 Ruido en las imágenes digitales.	40
4.2 Algoritmos de Filtrado	41
4.2.1 Filtros lineales espaciales	44
4.2.2 Filtros no lineales	44
4.2.3 Filtros lineales de paso alto	44
4.2.4 Filtros pesados de paso alto	46
4.2.5 Filtros lineales de paso bajo	48
4.2.6 Filtros no lineales de paso bajo	49

Capítulo 5. Resumen de redes neuronales y arquitectura tentativa	51
5.1 Antecedentes y notación básica	51
5.1.1 Neurona de una sola entrada.	55
5.1.2 Índices de los valores de los pesos.	57
5.1.3 Arquitecturas de red.	58
5.1.4 Reglas de aprendizaje de las redes neuronales.	59
5.2 Ejemplos de redes neuronales.	60
5.2.1 Red de Hopfield.	60
5.2.2 Algoritmo de la red de Hopfield	60
5.3 Perceptron	62
5.3.1 Perceptron de una sola capa	62
5.3.2 Algoritmo de convergencia del Perceptron.	63
5.3.3 Perceptron multicapa.	64
5.4 Algoritmo de entrenamiento Marquard-Levenberg	65
5.4.1 Algoritmo de retropropagación	66
5.4.2 Modificación de Marquard-Levenberg.	69
Capítulo 6. Algoritmos de tratamiento de la imagen del iris	72
6.1 Algoritmos de manipulación de la imagen del iris previos al entrenamiento de la red.	73
6.1.1 Conversión de la imagen a escala de grises	73
6.1.2 Algoritmo de normalización de la imagen del iris	75
6.1.3 Algoritmos de filtrado de la imagen del iris	75
6.1.4 Algoritmo de intensificación de la imagen	77
6.2 Ejemplo del procesamiento de las imágenes obtenidas	77
6.3 Descomposición de la imagen	78
6.4 Adelgazamiento de las lesiones	81
6.5 Componentes conexas	82
6.6 Valores iniciales del algoritmo de entrenamiento de la red neuronal	84
6.7 Ubicación de las lesiones en la gráfica iridológica	87
6.8 Diagnóstico	91
6.9 Utilización de la herramienta de diagnóstico	92
Capítulo 7. Conclusiones	94

Resumen

La iridología es un método de diagnóstico clínico utilizado en varios países para determinar padecimientos o enfermedades de una persona. Se realiza mediante la identificación de lesiones en el iris empleando lo que en la práctica de este método se conoce como *gráfica iridológica*, con ella se determinan los órganos del cuerpo que afecta una *lesión*. Además de identificar las *lesiones* o los padecimientos de una persona, los iridólogos indican el grado de avance que presenta la *lesión* con base en la degradación de los tejidos del iris; a esto se le conoce como *estado* de la lesión. En este trabajo no pretendemos demostrar la validez del método, sino brindar una herramientas automáticas de diagnóstico.

En el desarrollo de este trabajo hemos utilizado distintas técnicas para manipular imágenes digitales, que involucran varias áreas de las ciencias de la computación: topología digital, reconocimiento de patrones, procesamiento de imágenes, algoritmos y redes neuronales, así como las técnicas iridológicas.

En el primer capítulo se da una descripción general del trabajo. En el segundo mostramos la clasificaciones de lesiones posibles del iris humano en que se basa la iridología. En los capítulos tres y cuatro se desarrollan los elementos básicos sobre procesamiento de imágenes que hemos utilizado. El capítulo cinco presenta los aspectos básicos sobre redes neuronales haciendo énfasis en las que se usaron para reconocer lesiones en el iris. El capítulo seis es la síntesis de lo visto en los capítulos anteriores aplicada al diagnóstico iridológico, y desde nuestra perspectiva, en él se describen las aportaciones mas importantes del trabajo. Por último en el capítulo siete damos nuestras conclusiones.

Los resultados obtenidos han sido satisfactorios desde el punto de vista de que logramos identificar correctamente las lesiones presentes en imágenes de iris humanos, así como su estado y los órganos que afectan. Desde luego, esto no reemplaza el trabajo del iridologo porque existen características que sin ser lesiones en el iris en un momento dado podrían ser identificadas como tales.

Abstract

Iridology is a method used in several countries as a clinical tool to detect the health state in a person. It is based on the identification of lesions in the iris of the patient, by using what in iridology is known as *iridologic graph*, it helps to find the organs affected by a lesion. It is relevant in iridology to identify how advanced is an illness by analyzing the degradation of the iris tissue, it is known as *state* of the lesion. In this work we do not try to prove the effectiveness of iridology, our aim is just to offer some automatic diagnosis tools.

In our work we used different digital processing image techniques, they involve several branches of computer science: digital topology, pattern recognition, image processing, algorithms and neural networks; as well as iridologic techniques.

In the first chapter we give a general description of our work. In the second one we introduce the iridologic classification of possible iris lesions. In chapters three and four we develop the basic aspects on image processing involved in our work. Chapter five introduces neural networks and remarks those used to identify iris lesions. In chapter six we explain how we merged together the techniques of the previous chapters to build our iridologic diagnosis tools, which we consider our most relevant contribution. Finally, in chapter seven we give the conclusions of the work.

Our tools are able to identify correctly lesions from iris images as well as its state and affected organs. Of course, they cannot replace the iridologist work, because there exist some elements, as pigmentation and psoras, that our algorithms can wrongly identify as iris lesions.

1 Capítulo1. Introducción

Esta tesis pertenece al área de reconocimiento de patrones con redes neuronales artificiales [1]. Es un trabajo multidisciplinario aplicado a un método de diagnóstico clínico, denominado *iridología*, se basa en la identificación de lesiones en el iris causadas por enfermedades o padecimientos físicos de una persona[2].

El trabajo consiste en construir un sistema para la detección automática de este tipo de lesiones. En principio, se plantea un modelo de segmentación del iris (en forma anular y angular) que facilite la tarea de reconocimiento. Esta segmentación permite barrer la imagen para localizar lesiones que pueden aparecer en forma radial y angular en el iris (el capítulo 2 muestra una tipificación de lesiones), permite además obtener una representación adecuada para el cómputo de la información. Sobre esta segmentación se analizan las lesiones existentes comparando contra un conjunto representativo de lesiones definido por un experto iridólogo. Este conjunto constituye los patrones de entrenamiento de una red neuronal. En la fase de diagnóstico se parte de imágenes digitalizadas de los ojos del paciente, se aíslan primero las manchas en el iris mediante técnicas de filtrado de imágenes (suficientemente robustas como para que el proceso sea independiente de la coloración del ojo), a continuación se transforma cada mancha aislada a una representación matricial, que después se somete a una red neuronal para determinar el tipo de lesión correspondiente, por último, el área que ocupa cada lesión se compara contra la *gráfica iridológica* que es un mapa que relaciona las regiones del iris con el organismo humano para determinar los órganos afectados[2].

El mapa o carta del iris mencionado en el párrafo anterior ha sido utilizado y probado por sociedades de iridología en Estados Unidos, Europa y Gran Bretaña¹. Sin embargo, es importante aclarar que este trabajo no pretende demostrar ni refutar la efectividad de este método. Nuestro objetivo es simplemente reproducir de manera automática el trabajo de un iridólogo, y serán

¹en la página de Intenet: <http://www.geocities.com/albaruthenia/IA/science.html> se presentan direcciones de páginas de las academias y sociedades científicas que investigan estos temas y un artículo de la IEEE sobre los avances del área.

las opiniones de expertos en esta área las que determinen el nivel de éxito.

1.1 Distribución general

En este primer capítulo describimos las soluciones actuales computarizadas que se tienen para realizar diagnóstico iridológico. Mostramos además un resumen general de la metodología empleada en este trabajo.

En el capítulo 2 se presentan las bases de la iridología y se da una clasificación detallada de lesiones en el iris. Se concluye mostrando algunas fotografías que ejemplifican estas lesiones.

En el capítulo 3 analizamos las técnicas de representación de imágenes y definimos los conceptos fundamentales como: niveles de gris, vecindad, conexidad y métodos de adelgazamiento de imágenes.

El capítulo 4 explica las características de varios algoritmos de filtrado de imágenes y presenta algunos métodos concretos. También explica, a partir de sus características, cuáles filtros empleamos para el tratamiento de las imágenes del iris y los beneficios que aporta cada uno.

El capítulo 5 muestra un resumen sobre el estado del arte en el campo de las redes neuronales artificiales, y las formas de aprovechamiento de tal metodología en este proyecto.

El capítulo 6 describe como hemos empleado los conceptos tratados en los capítulos anteriores para construir nuestras herramientas de diagnóstico iridológico.

Finalmente en el capítulo 7 se presentan las conclusiones del trabajo.

1.2 Soluciones actuales

En [3], se presenta un análisis computarizado del iris basado en el filtrado de imágenes. Lo que hace es asociar los distintos órganos del cuerpo con regiones bien determinadas en el iris; se parte del llamado anillo de asimilación (que está asociado con la región que bordea la pupila) y la denominada corona del nervio autónomo que abarca los anillos antes mencionados así como el anillo de los intestinos. Se presenta además un programa que es utilizado básicamente para determinar si existe toxemia (presencia de sustancias tóxicas en la sangre que condicionan un proceso patológico) y su grado de avance (agudo, sub-agudo, crónico y degenerativo).

El trabajo antes mencionado demuestra que el filtrado de imágenes es a-propiado en la detección de algunas enfermedades, sin embargo, resulta insuficiente cuando se necesita identificar al conjunto total de lesiones en el iris. El problema proviene de los grandes volúmenes de información necesarios para detectarlas.

El tratamiento de la imagen y el uso de redes neuronales propuesto, nos ha permitido identificar distintos tipos de lesiones en el iris que se asocian a su vez con distintos tipos de enfermedades de manera tal que el análisis no se restrinja a un sólo tipo de padecimiento.

Existe una gran cantidad de algoritmos basados en redes neuronales artificiales que resultan efectivos en el reconocimiento de imágenes. Las redes neuronales tienen además la característica de resolver conflictos de ambigüedad sobre los datos que se les presentan. Esto aporta la ventaja adicional de reducir la cantidad de información que se almacena en el programa de diagnóstico.

Con los resultados obtenidos se ha pretendido automatizar el proceso de diagnóstico en iridología y proporcionar bases para extender la investigación en este campo.

1.3 Metodología utilizada

La construcción de las herramientas de diagnóstico se dividió en varias etapas:

- Se realizó una investigación sobre distintos tipos de redes neuronales aplicables en el reconocimiento de patrones [4][5].
- Se llevaron a cabo reuniones con un especialista en iridología para comprender la metodología de diagnóstico.
- Se investigaron y aplicaron distintos métodos de filtrado para aislar lesiones de manera que el proceso resulte independientemente de la coloración del iris.
- Se planteó un método de descomposición del iris que nos permitió aplicar técnicas de reconocimiento de patrones de manera que los resultados fueran relevantes para nuestros propósitos, el método se basó

en una segmentación del iris tanto anular como radial, que permitió, de una manera efectiva, identificar lesiones abiertas y cerradas (los tipos de lesiones se definen en el apartado 2.2). En adelante llamaremos a esto esquema de *segmentación del iris* o simplemente *segmentación del iris*.

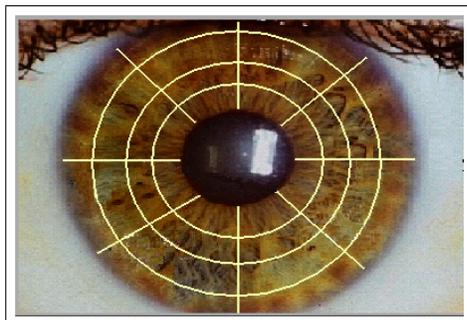


Figura 1: Esquema de segmentación del iris

- Cada uno de los elementos en la *segmentación del iris* se transformó a una *representación matricial*. Estos elementos son la base de imágenes que utilizamos para entrenar nuestras redes neuronales.
- Se construyeron varios tipos de redes neuronales y se analizó su efectividad para reconocer lesiones en el iris. Al final se eligieron las redes que mostraron un mejor desempeño, concretamente estas redes fueron: Perceptron [4], Hopfield [6] y Marquardt [7].
- Una vez identificadas las lesiones éstas se ubicaron sobre el *mapa iridológico* para determinar los órganos que se encuentran afectados, esta relación se muestra en la Figura 2.



Figura 2: Mapa de asociaciones de órganos. Ojo Izquierdo y derecho

- Se construyó el sisema de diagnóstico.
- Se evaluó la efectividad de los métodos presentados, esto se hizo comparando los resultados del programa de diagnóstico con los resultados reportados en la literatura.

Agradecimientos

Queremos agradecer de manera muy especial a **Ariel Guzik** por su ayuda en la clasificación de lesiones y su apoyo para comprender el diagnóstico de iridología. El **Dr. Guillermo Morales Luna** propuso correcciones valiosas y **Nicandro Silva** me apoyo de manera incondicional durante todo el trabajo.

2 Capítulo 2. La iridología

El iris recibe su nombre de los múltiples colores que en él se encuentran. Es una estructura de 4 capas muy delgadas suspendidas entre la córnea y el cristalino, perforada ligeramente en el lado nasal, cerca de la pupila. La *iridología* (estudio del iris) es un método de diagnóstico alternativo holístico, en la práctica convencional de ésta, el cuerpo se concibe como un sistema integral y su representación (según los iridólogos) está dada por un *mapa* (o carta) *iridológica*[2] similar al mostrado en la Figura 3.

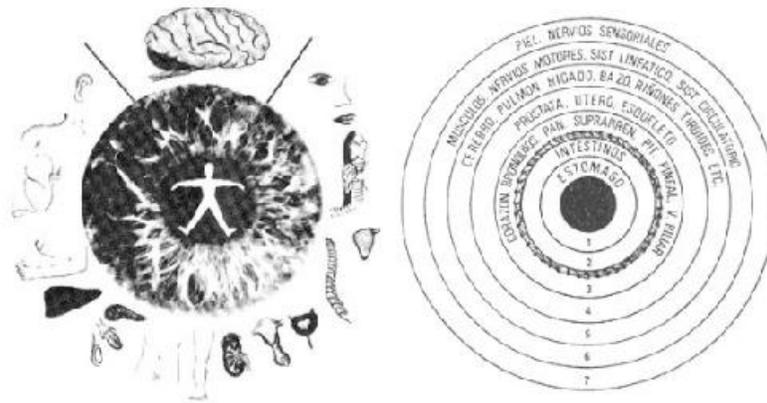


Figura 3: Representación del cuerpo en el iris

En la Figura 4, se muestra la distribución de los órganos representados en el iris. Una línea gruesa, oscura, representa el cerebro, indica dónde comienza y termina el área cerebral, además indica dónde comienza el sistema digestivo, dónde el colon y dónde están los límites del intestino delgado. En ella el área del intestino delgado está separada, con el fin de que se pueda reconocer con sólo un vistazo. La gráfica también está dividida en 7 zonas circulares, esto ha sido fundamental en nuestro sistema para determinar las zonas afectadas por una lesión. Estas zonas son: (1) estómago (2) intestinos (3) corazón, bronquios, etc. (4) próstata, útero, etc. (5) cerebro, pulmón, hígado (6) músculos, sistema linfático, sistema circulatorio (7) piel .

GRÁFICA IRIDOLÓGICA

DR. BERNARD JENSEN

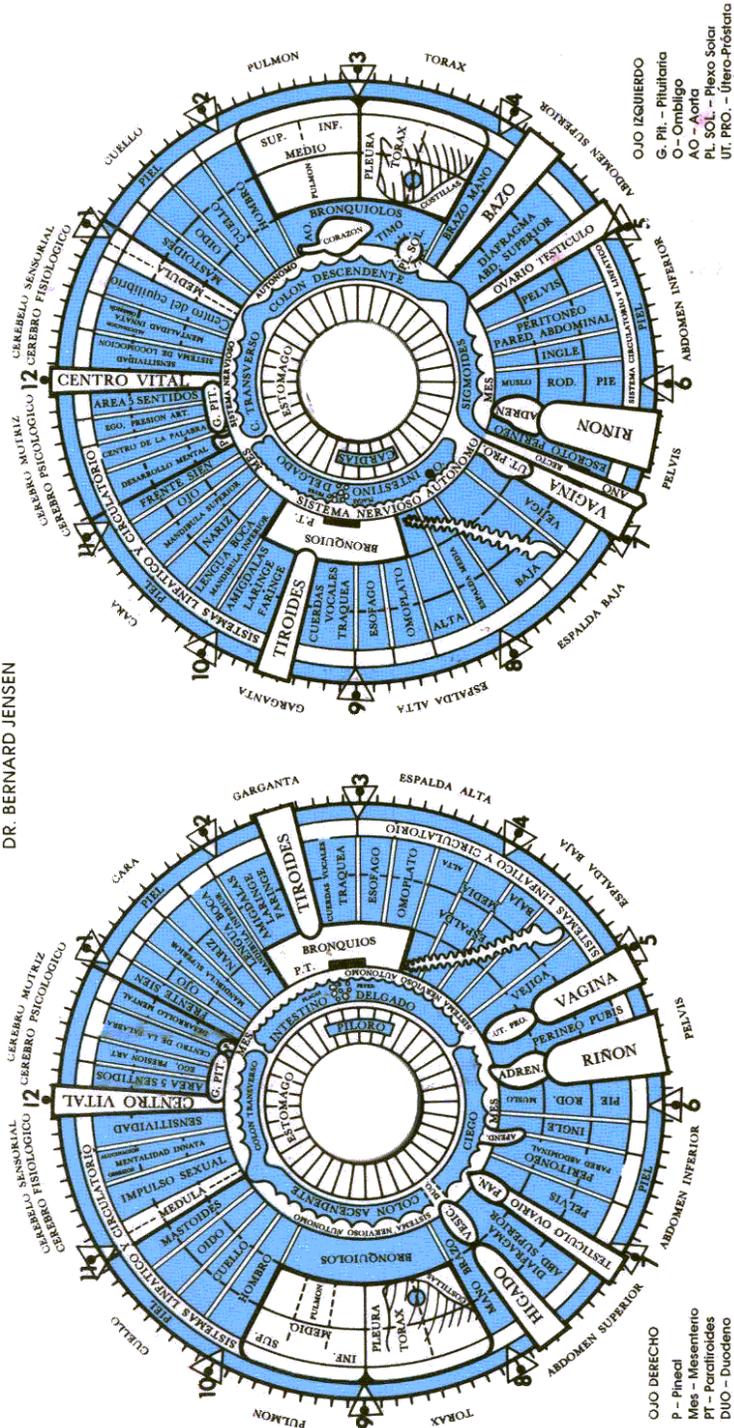


Figura 4: Gráfica Iridológica

D. R. © EDITORA Y DISTRIBUIDORA YUG, S.A., Hamburgo 290, Col. Juárez, C.F. 06600, México, D.F. D. R. © 1981, BERNARD JENSEN

2.1 Definición de lesiones en el iris

Una *lesión* es una alteración de las características anatomohistológicas de un órgano o tejido, con déficit funcional del mismo, producida por la acción de una causa patógena, un golpe o una herida [8].

Existen múltiples lesiones que pueden identificarse en el iris, estas lesiones tienen distintas formas y colores. A continuación se presentan varias lesiones que se pretenden identificar con el sistema. Daremos sus características principales.

Rosario linfático

Esta lesión se caracteriza por una serie de pequeñas manchas o nubes en forma semicircular, un poco indefinidas y “gelatinosas”. Su color es claro y forma un anillo (rosario) total o parcialmente definido en la periferia del iris. Abarca de un 30% a un 70% del área del iris. Indica congestión, inflamación y aumento del tejido linfático.

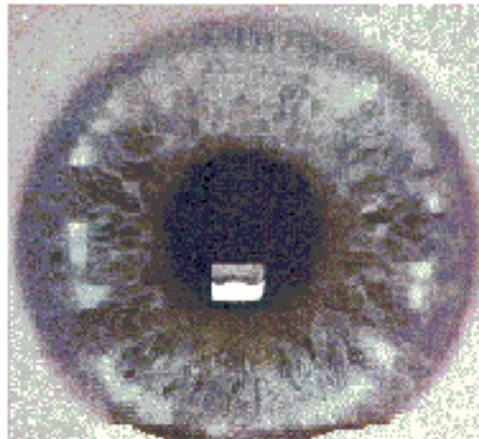
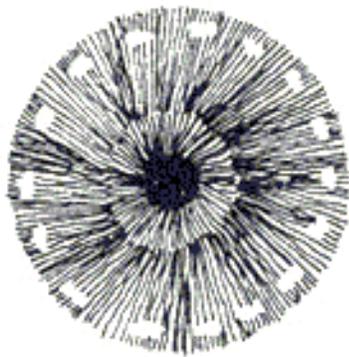


Figura 5: Rosario linfático

En la siguiente imagen podemos apreciar un rosario linfático en una fotografía de un iris real y como se trabajaría con ella en el sistema resultado de nuestro trabajo. Del lado izquierdo de la pantalla se ve la imagen del iris que lo contiene.

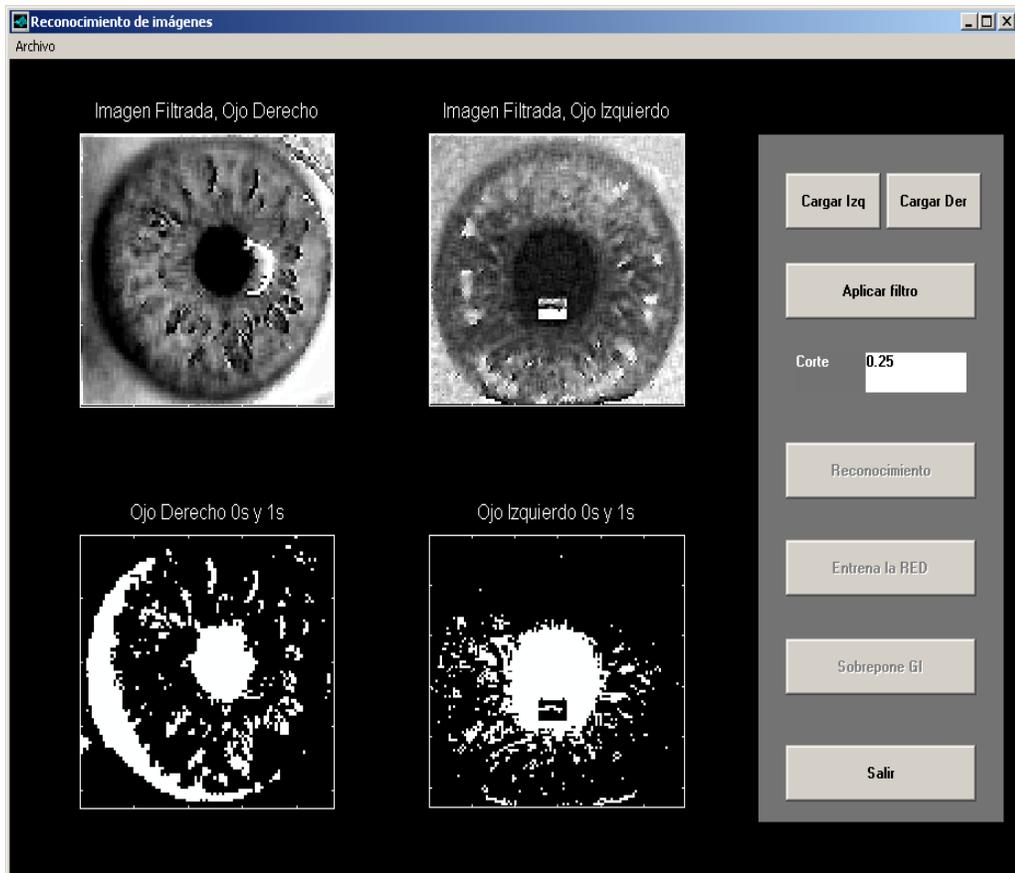


Figura 6: Pantalla de nuestro sistema que muestra un iris con rosario linfático

Congestión linfática o circulatoria

Similar a la lesión anterior (rosario linfático), esta lesión varía solamente en que la nube que se presenta es más uniforme y no está definida por semicírculos.

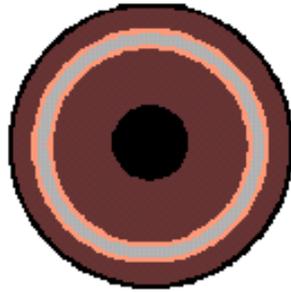


Figura 7: Congestión linfática o circulatoria

Anillo escamoso

Esta lesión es fácilmente identificable, ya que se manifiesta mediante un anillo perfectamente definido en la periferia y sobre el iris. Su color es azul y tiene un grado de obscuridad mayor que el resto del iris. La presencia de este anillo indica toxinas y desbalance en el metabolismo de la piel.

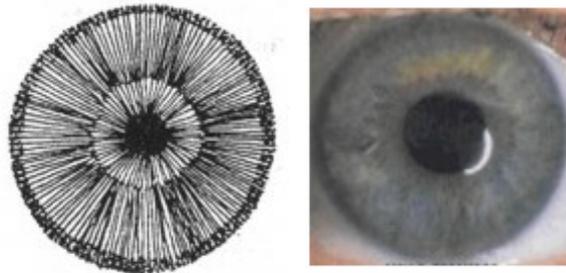


Figura 8: Anillo Escamoso

Arco senil

Es un anillo opaco en la porción más alta del iris (perteneciente al área del cerebro). La diferencia entre el anillo escamoso y el arco senil es que éste último no está definido en el iris sino en la córnea o cristalino. Es el signo clásico de la “edad avanzada” y representa carencia circulatoria en el área de la cabeza. Se asocia frecuentemente a problemas de memoria y declinación de las funciones cerebrales.

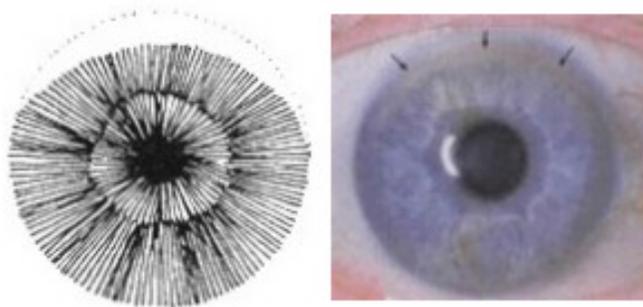


Figura 9: Arco Senil

Anillos nerviosos (estrés)

El estrés es un estado agudo de tensión o cansancio, físico o psíquico [8]. Los anillos nerviosos aparecen en forma perpendicular al radio del iris, son de color claro y producen pequeñas inflamaciones (mostradas en la parte derecha de la Figura 10). Indican una condición de ansiedad, tensión o estrés en las partes del cuerpo en que se encuentran, provocando rigidez y falta de circulación al sistema nervioso.

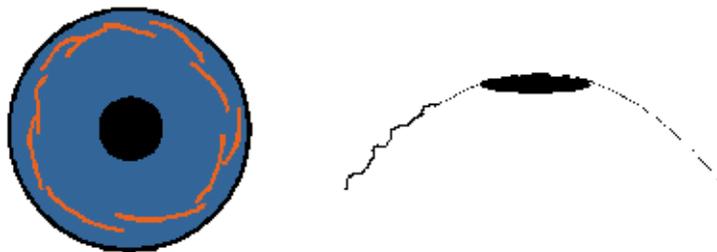


Figura 10: Anillos nerviosos

Radio solaris (*raddi solaris*)

Esta lesión se caracteriza por la presencia de líneas que van del estómago (en su representación en el iris) hacia la periferia del iris (pero no la tocan). Generalmente son de color oscuro (café) mezclado con anaranjado. Normalmente son convexas y se puede observar que producen inflamación en el área donde el rayo pasa. Indican absorción de toxinas formando canales desde el colon hasta los órganos que toca el rayo.

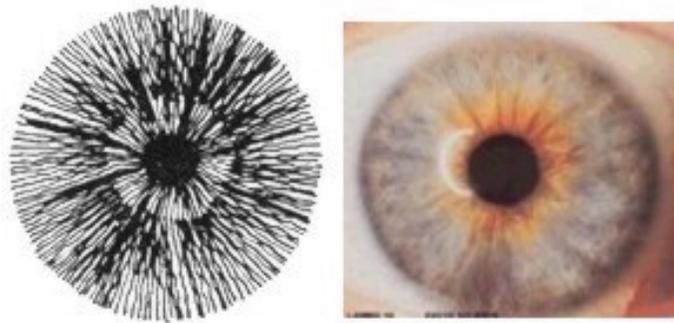


Figura 11: Radio solaris

La siguiente figura muestra un iris que contiene varios tipos de lesiones. Las más importantes son: un anillo escamoso denso al rededor del iris y los *raddi solaris* que se localizan principalmente en la parte superior.

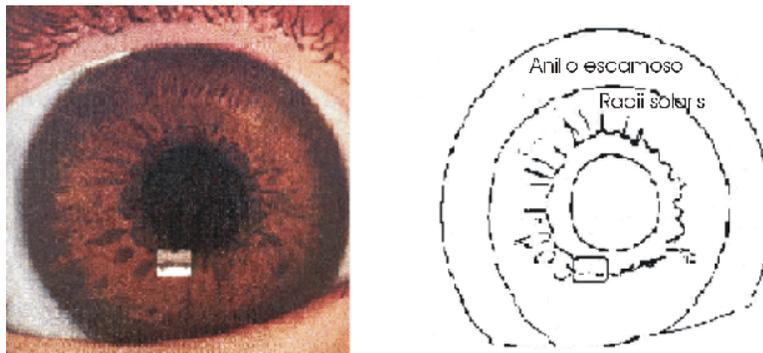


Figura 12: Ejemplo de un iris con radio solaris y anillo escamoso

Divertículos

Los divertículos son lesiones similares a los rayos solares, pero son de mayor alcance porque tienen un grosor considerablemente más grande que los de la lesión anterior.

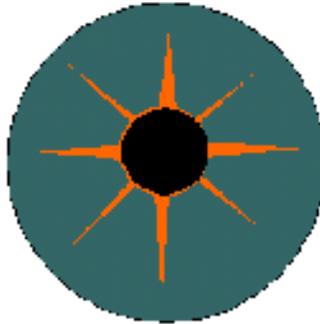


Figura 13: Divertículos

Anillo de absorción

El anillo de absorción es un círculo rojizo que aparece alrededor de la pupila. Aunque no es de un grosor considerable está perfectamente definido y es fácilmente identificable. Este anillo indica la deficiencia gastrointestinal para absorber los nutrientes. El oscurecimiento de esta área indica deficiencias en esa capacidad.

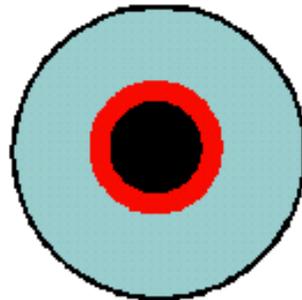


Figura 14: Anillo de absorción

Psoras o hiperpigmentación

Las psoras son manchas muy homogéneas en el color y totalmente contrastantes con el iris. Son heterogéneas en su forma, pero pueden identificarse porque regularmente no se encuentran en forma radial a las líneas del iris, sino en que aparecen aleatoriamente. Las áreas del tejido son débiles y se presentan con colonias de toxinas, existe desgaste en el cuerpo y es un signo hereditario.

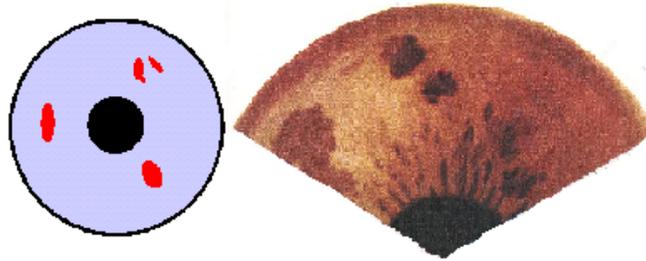


Figura 15: Psoras

Colesterol/sodio

La presencia de colesterol o sodio en el cuerpo puede identificarse en el iris por la aparición de un anillo en la periferia. Es un depósito definido en la cornea. Indica acumulamiento crónico y absorción de sales inorgánicas o sustancias grasas. Se asocia con endurecimiento de las arterias y presión alta.

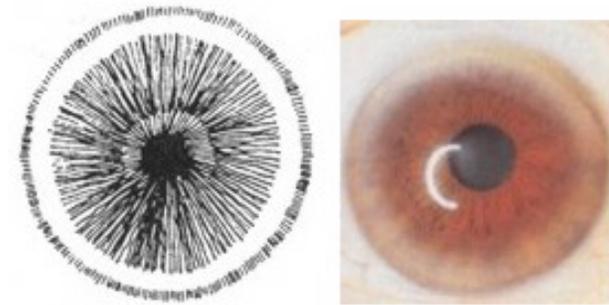


Figura 16: Colesterol/sodio

Sobre acidificación

Esta lesión se caracteriza por una blancura predominante que cubre toda el área del tejido del iris. Indica acidificación crónica, resultado de desbalances en la dieta alimenticia que contiene alto contenido de ácidos, como azúcares, trigo y productos lácteos.

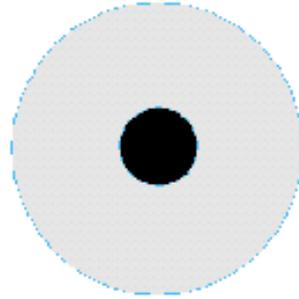


Figura 17: Sobre acidificación

Congestión de senos faciales

Esta lesión presenta una serie de líneas similares a los divertículos. Sin embargo, solamente aparecen en la parte superior del iris. Van desde la pupila hacia la periferia del iris y atraviesan el estómago.

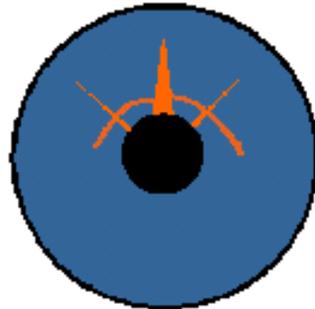


Figura 18: Congestión de senos faciales

Anemias (extremidades)

La característica de esta lesión es la aparición de semicírculos que pueden estar definidos tanto en la parte superior como en la inferior -o en ambas- del iris. Estas líneas indican falta de irrigación sanguínea hacia el cerebro (cuando se encuentran en la parte superior) y son características de senitud, o falta de irrigación sanguínea hacia las extremidades (manos, brazos, piernas y pies) que se encuentran regularmente frías, cuando aparece en la parte inferior.

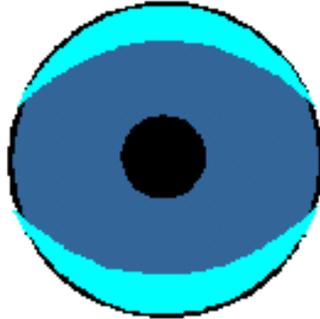


Figura 19: Anemias

Estómago sobre acidificado

El área que representa al estómago en el iris es un círculo que está alrededor de la pupila y puede no estar perfectamente definido en todos los casos. Cuando está sobre acidificado este anillo presenta un color blanco intenso mucho mayor que cualquier otra parte del iris. Indica un exceso de ácido hidrocloreídrico.

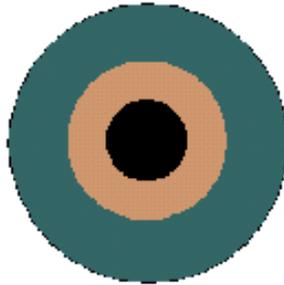


Figura 20: Estómago sobre acidificado

Estómago infra acidificado

Esta lesión es de características similares a la anterior. En esta aparece un círculo que rodea la pupila en la zona del estómago y es de color oscuro. Indica un desbalance en la química estomacal en el que la proteína digestiva de ácido hidrocloreídrico es insuficiente. Indica además carencia de sodio orgánico y una potencial anemia proteica.

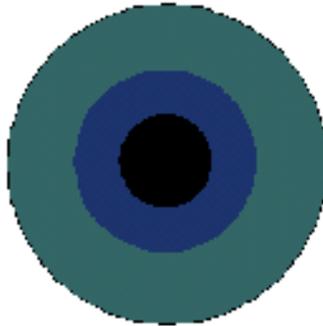


Figura 21: Estómago infra acidificado

2.2 Lesiones relacionadas con los intestinos

En la gráfica iridológica de la Figura 4 puede observarse que el órgano que sigue al estómago radialmente son los intestinos. Estos órganos se encuentran representados tanto en el ojo izquierdo como en el derecho. A continuación se presentan algunas de las lesiones en el iris ligadas con afectaciones en estos órganos.

Intestino estrecho

Aunque con algunas variantes en cada iris, la forma del intestino es fácilmente identificable y similar a la que aparece en la Figura 22, puede identificarse en el iris, debido a que tiene un color contrastante que es, generalmente, más claro. El intestino estrecho es una lesión donde su correspondencia en el iris aparece rodeando estrechamente la pupila.

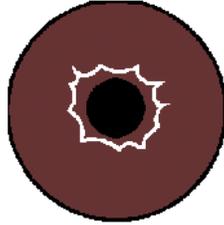


Figura 22: Intestino estrecho

Intestino espástico

Esta lesión es la opuesta de la anterior.

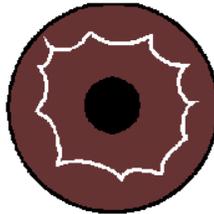


Figura 23: Intestino espástico

Intestino espástico y estrecho

La Figura 24 muestra la combinación de las dos lesiones anteriores (espástico y estrecho).

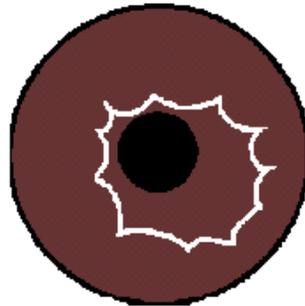


Figura 24: Intestino espástico y estrecho

Prolapsos

Los prolapsos son también lesiones en el iris. Se manifiestan por la cercanía del intestino a la pupila, ya sea en la parte superior o en la parte inferior del iris.

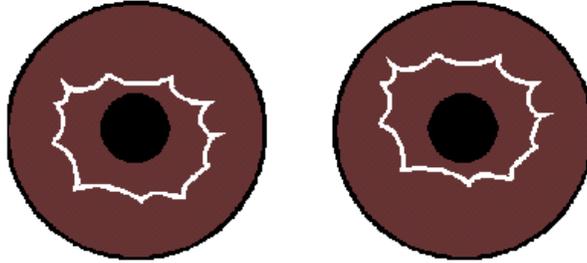


Figura 25: Prolapsos

2.3 Lesiones abiertas y cerradas

Existen otros tipos de lesiones en el iris que se caracterizan porque sus fibras radiales se encuentran separadas o expandidas. Su estructura puede ser cerrada o abierta. Una *lesión cerrada* (Figura 26 centro) es una mancha en forma de óvalo o rombo que tiene un contorno bien definido. Varía en color de acuerdo al avance de la lesión (o *estado*, tratado en la sección 2.4). Son paralelas a las líneas radiales del iris que van desde la pupila a la periferia. Estas lesiones se presentan en cualquier parte del iris a partir de la línea del estómago. Pueden aparecer en forma conjunta, (varias a la vez) y reciben el nombre de *racimos*. Además pueden presentarse dentro o junto a una *lesión abierta*. Cuando una *lesión cerrada* es muy pequeña y de color negro se denomina **cripta** y son símbolo inequívoco de un tumor en el área donde aparecen porque representan un órgano muerto.

La *lesión abierta* tiene forma triangular no tiene base porque este tipo de lesión está siempre sobre el contorno del iris (periferia), en algunas ocasiones también se encuentra abierta en la parte superior. Al igual que en la *lesión cerrada* no abarca el área del estómago y su color depende del *estado* en que se encuentre (Figura 26 derecha).

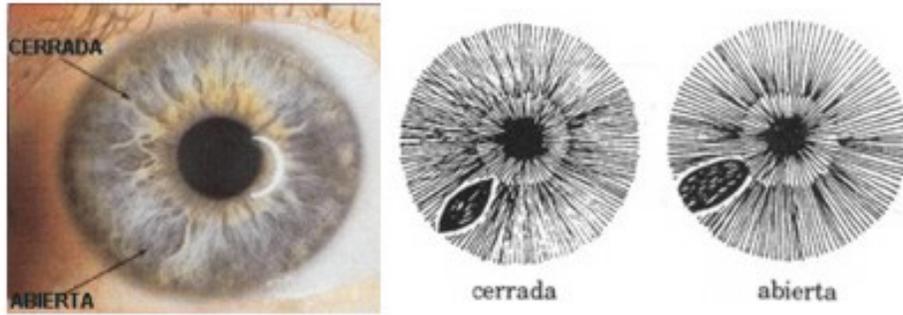


Figura 26: Lesiones cerradas y abiertas

2.4 Estados de las lesiones

Se denomina *estado*, al avance que presenta una lesión ya sea abierta o cerrada. Existe una clasificación básica de 4 tipos: *agudo*, *sub-agudo*, *crónico* y *degenerativo*, pero puede encontrarse en la bibliografía clasificaciones más detalladas[10]. Se mide toda inflamación en el iris por una escala del 1 al 5, considerando el 3 como el promedio. Si un paciente está en el estado 1 de inflamación y de toxicidad, consideramos que está en un estado subagudo. Si está en el 4 entonces puede hallarse en un estado crónico y cuando llega al 5 consideramos que está en camino a una degeneración.

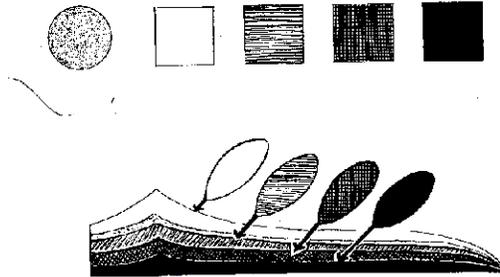


Figura 27: Estado de las lesiones[2]

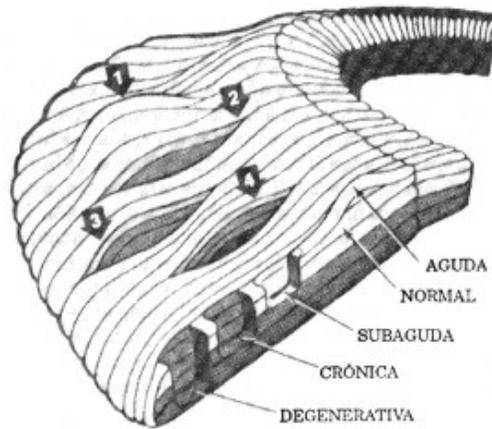


Figura 28: Vista lateral de los estados de una lesión[2]

2.4.1 La crisis curativa o la ley de curación de Hering

Constantine Hering fue un médico homeópata de cierto renombre en la Europa del siglo XIX. Como resultado de su trabajo como médico clínico, formuló una ley de curación que sostiene que la enfermedad se cura mediante un proceso de reversión. “Toda curación procede de adentro hacia fuera, de la cabeza hacia abajo y en orden inverso a aquel en que se fueron manifestando los síntomas del padecimiento”. Es el proceso por el cual pasa el organismo cuando la sangre se limpia y reconstituye químicamente por medio de la nutrición. La Figura 29 muestra las líneas que se forman cuando comienza el proceso curativo.



Figura 29: Líneas de curación

2.4.2 Agudo (en relieve)

Cuando una lesión se encuentra en este estado, existen hilos o fibras que atraviesan la lesión. El área donde está definida es de color blanco, se dice que hay un exceso de actividad (hiperactividad). Se observa inflamación en el área afectada, mostrada en la parte derecha de la Figura 30. En este estado el organismo puede restablecerse asimismo si la persona está sana.



Figura 30: Estado agudo

2.4.3 Sub-agudo (yace exactamente bajo la superficie)

Las fibras que se encuentran en la lesión tienen un color que va del verde al café y la lesión tiene una tonalidad más oscura que la pigmentación predominante del iris. La acción del metabolismo disminuye al igual que la inflamación que se observa en el estado anterior.



Figura 31: Estado sub-agudo

2.4.4 Crónico (profundo, debajo de la superficie)

Cuando una lesión llega a este estado, tiene un color negro en más de un 25%. Ya no hay inflamación sino que la lesión es convexa. Esto indica baja actividad (crónica) en el tejido.



Figura 32: Estado crónico

2.4.5 Degenerativo (muy profundo, en el fondo)

La lesión en este estado es de color negro en un 75% y existe destrucción del tejido del iris así como carencia total de sensibilidad en las áreas afectadas.



Figura 33: Estado degenerativo

3 Capítulo 3. Representación de imágenes

En la actualidad existen varios métodos para representar imágenes. Debido a que éstas suelen requerir demasiado espacio para su representación y almacenamiento las investigaciones se centran en representaciones de imágenes mediante algunos de sus componentes o características principales que permitan, posteriormente, reconstruirlas. En este capítulo investigamos distintas formas de representación, las características que las definen y algunos métodos para su manipulación y tratamiento, como son: conversiones a niveles de gris, métodos de adelgazamiento. Estos métodos se aplican a imágenes del iris, para filtrar la información relevante en el proceso de reconocimiento.

En las secciones de la 3.1 y 3.2 presentamos las imágenes digitales y la representación morfológica que son dos formas de representación de imágenes. En las secciones 3.3 y 3.4 explicamos las operaciones fundamentales para hacer transformaciones morfológicas, estas permiten extraer características de imágenes y son empleadas a menudo por los algoritmos de filtrado que se presentan en el capítulo 4. En las secciones subsecuentes abordamos los métodos de adelgazamiento de imágenes digitales, que son fundamentales en el reconocimiento de patrones.

3.1 Imágenes digitales[11]

Sean i y r , las funciones de *iluminación* y *reflexión* respectivamente, definidas como $i : [a, b] \times [c, d] \rightarrow \mathbb{R}^+$ (donde a, b, c, d son números reales tales que $a < b$ y $c < d$) y $r : [a, b] \times [c, d] \rightarrow [0, 1]$. Definimos una *imagen digital* como la función bidimensional $f : [a, b] \times [c, d] \rightarrow \mathbb{R}^+$ derivada del producto de $i \times r$, tomando la siguiente forma:

$$f = ir$$

en el intervalo:

$$\begin{aligned}0 < i < \infty \\ 0 < r < 1\end{aligned}$$

por lo que las cotas para f son:

$$0 < f < \infty$$

3.2 Representación morfológica y residual

La representación de imágenes digitales debe tomar en cuenta el significado geométrico de características como la forma y tamaño cuya relación define el concepto de *escala* (el enfoque no lineal para representaciones multiescalares puede ser consultado en [12]). La *escala* es el patrón más pequeño (generado con un patrón prototipo de tamaño unitario) que pueda “caber” dentro de una señal. Sin embargo, la representación de una imagen bi-dimensional puede verse como una serie de componentes simples que se obtienen recursivamente mediante una función de *transformación constructiva*, este esquema de representación es muy general. La información escalar se encuentra inmersa en la representación mediante una selección apropiada de la *función de transformación*. Morfológicamente, la descomposición de una forma (MSD Morphological Shape Decomposition) [13] es un caso particular del esquema de representación mencionado[14]. Las transformaciones morfológicas son importantes porque permiten reducir la información necesaria para representar una imagen.

3.3 Definiciones fundamentales para transformaciones morfológicas de imágenes

La morfología matemática se basa en la teoría de conjuntos y en la topología. En la sección 3.1 presentamos la definición de imagen digital, podemos tener una definición alternativa de este concepto a partir de que la imagen está formada por un conjunto (o colección) de coordenadas discretas denominadas *pixeles*. Sin embargo, un pixel no es considerado estrictamente una coordenada puntual sino más bien como un cuadrado unitario cuyo centro coincide

con sus coordenadas. En este sentido, podemos pensar que las coordenadas que identifican un pixel pertenecen a Z^2 . En el ejemplo que aparece en la Figura 34 observamos dos conjuntos de pixeles A y B .

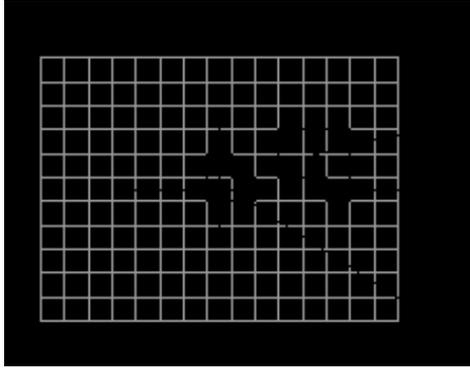


Figura 34: Imagen binaria con dos objetos, los conjuntos A y B

3.4 Transformaciones morfológicas

Las operaciones fundamentales que podemos asociar a una image, vista como un conjunto de pixeles, son las básicas de la teoría de conjuntos: unión, intersección, complemento $\{\cup, \cap, ^c\}$ más la *traslación* que definimos a continuación:

Traslación: Dado un vector x y un conjunto A . La traslación, $A + x$, se define como:

$$A + x = \{a + x \mid a \in A\}$$

Conjunto de operaciones básicas de Minkowski[15]. Con los conceptos anteriores podemos definir la *adición* y la *substracción*. Los elementos individuales que componen un conjunto B son pixeles; pero también pueden ser vistos como vectores (formados por las coordenadas asociadas a los pixeles). Así, dados los conjuntos de pixeles A y B podemos definir las siguientes operaciones:

$$\begin{array}{ll}
\text{Adición de Minkowski} & A \oplus B = \bigcup_{\beta \in B} (A + \beta) \\
\text{Substracción de Minkowski} & A \ominus B = \bigcap_{\beta \in B} (A + \beta)
\end{array}$$

Derivadas de las operaciones de Minkowski tenemos las cuatro transformaciones morfológicas que se definen a continuación:

$$\begin{array}{ll}
\text{Dilatación:} & D(A, B) = A \oplus B = \bigcup_{\beta \in B} (A + \beta) \\
\text{Erosión:} & E(A, B) = A \ominus (-B) = \bigcap_{\beta \in B} (A - \beta) \quad \text{donde } -B = \{-\beta \mid \beta \in B\} \\
\text{Apertura:} & O(A, B) = A \circ B = D(E(A, B), B) \\
\text{Cierre:} & C(A, B) = A \bullet B = E(D(A, -B), -B)
\end{array}$$

Como ya mencionamos todas estas operaciones son necesarias para la extracción de características, com: la *extracción del perímetro*.

3.5 Muestreo espacial y niveles de gris

Una forma alternativa más de definir una *imagen digital* es como una función $f(x, y)$. Sea A un conjunto finito de pixeles, la función se define como:

$$f(x, y) = \begin{cases} 1 & \text{si } (x, y) \in A \\ 0 & \text{en otro caso} \end{cases}$$

Como el conjunto A es finito esta función puede ser representada por una matriz de $N \times M$ elementos en la forma que se muestra a continuación:

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(1, 0) & \dots & f(N-2, 0) & f(N-1, 0) \\ f(0, 1) & f(1, 1) & \dots & f(N-2, 1) & f(N-1, 1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ f(0, M-2) & f(1, M-2) & \dots & f(N-2, M-2) & f(N-1, M-2) \\ f(0, M-1) & f(1, M-1) & \dots & f(N-2, M-1) & f(N-1, M-1) \end{bmatrix}$$

En la práctica una imagen no se encuentra determinada únicamente por la presencia o ausencia de píxeles sino por niveles de gris, así es que, de una manera más general, podemos pensar que el contradominio de la función $f(x,y)$ es el intervalo $[0,1]$. Bajo esta interpretación el *histograma* de una imagen es una función que indica la ocurrencia de cada nivel de gris. Puede entenderse también como la probabilidad de que un valor de gris determinado aparezca en la imagen. Los histogramas condensan la información de las imágenes; pero tienen el inconveniente de que por sí solos no aportan información alguna sobre la localización espacial de los píxeles.

3.6 Relaciones entre píxeles: vecindad, conexidad

Un *pixel* p de coordenadas (x, y) presenta un total de cuatro vecinos en el plano vertical y horizontal, las coordenadas de estos cuatro vecinos se muestran en la Figura 35.

	$x, y - 1$	
$x - 1, y$	x, y	$x + 1, y$
	$x, y + 1$	

Figura 35: Vecindad $N_4(p)$

Este conjunto de píxeles se denomina *vecindad de tipo 4* del pixel p , y se representa por $N_4(p)$. Además podemos considerar la existencia de otros cuatro vecinos asociados a las diagonales, cuyas coordenadas se muestran en la Figura 36:

$x - 1, y - 1$		$x + 1, y - 1$
	x, y	
$x - 1, y + 1$		$x + 1, y + 1$

Figura 36: Vecindad $ND(p)$

Los cuales se representan por $ND(p)$. La unión de $N_4(p)$ con $ND(p)$ define la *ocho vecindad* del pixel p , representada por $N_8(p)$.

Decimos que dos píxeles están *conectados* si son *adyacentes* (vecinos) y sus niveles de gris satisfacen alguna relación específica (por ejemplo ser iguales).

Existen tres tipos de conexidad:

- 1 Conexidad-4. Dos pixeles p y q presentan una *conexidad-4* si q pertenece a $N_4(p)$.
- 2 Conexidad-8. Dos pixeles p y q presentan una *conexidad-8* si q pertenece a $N_8(p)$.
- 3 Conexidad- m . Dos pixeles p y q presentan una *conexidad- m* si:
 - a) El pixel q pertenece a $N_4(p)$, o
 - b) El pixel q pertenece a $ND(p)$ y $N_4(p) \cap N_4(q) = \emptyset$.

La *conexidad- m* es una modificación de la *conexidad-8* cuya utilidad consiste en eliminar las conexiones múltiples que aparecen en ocasiones cuando se utiliza la *conectividad-8*. La Figura 37 muestra un ejemplo de estas conexides.

1	1	1	1	1	1	1	1	1	
1	-	1	0	1	\		/	0	
0	0	1	0	0	\	1	0	0	1

Figura 37: *Conexidad-4* *Conexidad-8* *Conexidad- m*

En Figura 37, vemos que para ir desde el *pixel central* al inferior derecho utilizaríamos *conexidad-8*, sin embargo, para ir al pixel superior derecho se podría ir de dos formas, directamente o a través del pixel superior. Esta indeterminación se evita con la conexidad mixta. En este caso, el pixel superior derecho no está conectado al central ya que presenta una conexión con el pixel superior.

En la Figura 38 observamos que si se utiliza la *conexidad-4*, existirían 4 objetos, cuando de manera intuitiva se observa que es uno solo. Si usamos la *conexidad-8* desaparece el problema, pero ahora tanto el *fondo exterior*

como el *interior* están conectados. Por ello cuando queremos determinar si dos pixeles pertenecen al mismo objeto tomamos *conexidad-8* y para el fondo *conexidad-4* o viceversa.

Esta “dualidad” en las vecindades sirve para lograr que se satisfaga el *Teorema de la Curva Cerrada de Jordan*, que establece que una curva simple cerrada en el plano delimita dos regiones: *el interior* y *el exterior*.

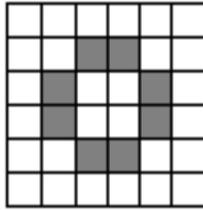


Figura 38: Ejemplo de conexidad

3.7 Distancia

Con la distancia queremos obtener el número mínimo de pasos elementales que se necesitan para ir de un punto a otro. Dados tres pixeles p , q y z , con coordenadas (x, y) , (s, t) y (u, v) respectivamente, podemos definir una función de distancia D si se cumple cumple que:

- $D(p, q) \geq 0, (D(p, q) = 0, \text{ si } p = q)$
- $D(p, q) = D(q, p)$
- $D(p, z) \leq D(p, q) + D(q, z)$

Las funciones de distancia usadas comúnmente son:

- La distancia euclidiana.
- La distancia "Manhattan".
- La distancia "tablero de ajedrez".

La *distancia euclidiana* entre dos pixeles p y q con coordenadas (x, y) y (s, t) respectivamente se define:

$$D_E(p, q) = \sqrt{(x - s)^2 + (y - t)^2}$$

con esta definición, las distancias del pixel central a los demás pixeles en la Figura 39 serán:

$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
2	1	0	1	2
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$

Figura 39: Distancia euclidiana

La *distancia euclidiana* involucra un gran número de cálculos y no toma en cuenta el concepto de vecindad que se ha descrito. Es por eso que se han definido otras distancias.

La *distancia "Manhattan"*, entre dos pixeles p y q con coordenadas (x, y) y (s, t) respectivamente se define como:

$$D = |x - s| + |y - t|$$

Ejemplos de esta distancia aparece en la Figura 40:

4	3	2	3	4
3	2	1	2	3
2	1	0	1	2
3	2	1	2	3
4	3	2	3	4

Figura 40: Distancia "Manhattan"

La distancia "tablero de ajedrez", entre dos pixeles p y q con coordenadas (x, y) y (s, t) respectivamente se define como:

$$D(p, q) = \max(x - s, y - t)$$

Se ejemplifica en la Figura 41:

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Figura 41: Distancia tablero de ajedrez

3.8 Métodos de Adelgazamiento

Para lograr el reconocimiento de patrones de las lesiones en las imágenes del iris se tiene la necesidad de reducir la cantidad de información a ser procesada al mínimo. En las primeras investigaciones sobre *métodos de adelgazamiento*[17] se definieron varios algoritmos[18] para obtener la representación adelgazada de ciertos patrones de caracteres. Los caracteres adelgazados se emplearon para reconocimiento de texto a partir de imágenes[19].

El diseño de algoritmos de adelgazamiento se basa en la esqueletización de los componentes de las imágenes. El término de "esqueleto" se emplea en general para denotar una representación de un patrón mediante una colección de curvas y arcos delgados (o casi delgados).

Los *algoritmos de adelgazamiento* eliminan sucesivamente capas de pixeles en el *borde* del patrón hasta que sólo quede el esqueleto. Un *pixel de borde* o *de contorno* es aquel que tiene al menos un pixel blanco en $N_4(p)$. La eliminación o retención de un pixel (negro) p dependerá de la configuración de pixeles en una *vecindad local* (ver sección 4.6) conteniendo a p . Estos algoritmos pueden clasificarse como:

- secuenciales o
- paralelos

En un *algoritmo secuencial*, cada pixel se examina para su eliminación en diferentes secuencias en cada iteración, y la eliminación de p en la n -ésima iteración depende de todas las operaciones que se hayan desarrollado anteriormente. Decimos que un pixel p se examina para su eliminación si es un pixel negro y los pixeles en una ventana de 3×3 se etiquetan de la siguiente forma:

x_4	x_3	x_2
x_5	p	x_1
x_6	x_7	x_8

Figura 42: Pixeles de $N8(p)$

Emplearemos x_i para denotar tanto al pixel como su valor 0 (blanco) ó 1 (negro). El número de pixeles negros en $N8(p)$ se denota por $b(p)$. Una secuencia de pixeles y_1, y_2, \dots, y_n se denomina una 8 - (4 -) *trayectoria* si y_{i+1} es un 8 - (4 -) vecino de y_i , $i=1, 2, \dots, n-1$. Un subconjunto Q de una imagen P es 8 - (4 -) *conexa* si para cada par de puntos x, y existe una 8 - (4 -) *trayectoria* desde x hasta y consistente de puntos en Q . En este caso, se dice que Q es una 8 - (4 -) *componente* de P . El orden de conexidad de la imagen P es el número de componentes de su complemento \bar{P} , y si este orden es 1 decimos que P es conexo simple, en otro caso P es conexo múltiple.

Un pixel p es 8 - (4 -) *eliminable* si al eliminarlo no cambia la 8 - (4 -) *conexidad* de P . Los pixeles que son candidatos para eliminarse son los *pixeles de contorno*. Los pixeles negros que no son de *contorno* decimos que son *pixeles interiores*. Se dice que un pixel p es *final* $b(p) = 1$.

La característica principal de los algoritmos de adelgazamiento es que durante su aplicación siempre se garantice la conexidad del objeto que se está adelgazando. Esta propiedad se expresa en términos del *número de cruzamientos*, *número de la conexidad* y *simplicidad del pixel*. Estos términos se definen a continuación.

Existen dos definiciones para el número de *cruzamientos de un pixel*. La primera de Rutovitz[20] propone que es útil medir la conexidad como el número de transiciones de un pixel blanco a un pixel negro y viceversa cuando los pixeles de $N8(p)$ son visitados en el orden mostrado en la Figura 42. Entonces definimos el número de cruces:

$$X_R(p) = \sum_{i=1}^8 |x_{i+1} - x_i|$$

donde $x_9 = x_1$ y esto es igual a dos veces el número de las *4-componentes* negras en $N8(p)$. La eliminación de p no afecta la *4-conexidad* si $X_R(p) = 2$ debido a que, en este caso, los pixeles negros en $N(p)$ son *4-conexos*. Un ejemplo donde p no es eliminable sería cuando tenemos los pixeles x_3, x_7 donde $X_R(p)=4$. De cualquier forma, dado que al separar los *4-componentes* siguen siendo *8-conexos*, el esqueleto obtenido empleando este método puede contener pixeles *8-eliminables*, este esqueleto suele denominarse *8-conexo imperfecto*.

La segunda definición llamada de Hilditch[21] especifica el *número de cruzamientos* $X_H(p)$ como el número de veces que uno cruza desde un pixel blanco a uno negro.

$$X_H(p) = \sum_{i=1}^4 b_i$$

donde

$$b_i = \begin{cases} 1 & \text{si } x_{2i-1} = 0 \text{ y } (x_{2i} = 1 \text{ o } x_{2i+1} = 1) \\ 0 & \text{en otro caso} \end{cases}$$

y $X_H(p)$ es igual al número de *8-componentes* negras de $N8(p)$ excepto cuando todos los $N4(p)$ son negros, en cuyo caso $X_H(p)=0$. Para ambas definiciones de *número de cruzamientos*, un pixel cuyos $N8(p)$ sean negros tendrá el número de cruces = 0 y lo denominamos *pixel aislado*. Si $X_H(p)=1$, la eliminación de p no cambia la *8-conexidad* del patrón.

Otra diferencia entre el *número de cruzamientos* de $X_H(p)$ y $X_R(p)$ es que la condición $X_H(p) = 1$ también implica que p es un *pixel de contorno*, mientras

que $X_R(p) = 2$ no garantiza esa condición debido a que p puede tener un vecino blanco en la esquina. Para evitar eliminar a p , en este caso (podría crearse un hueco), se requiere cumplir también otra condición, si $b(p) \leq 6$, se asegura que p es un *pixel de contorno*.

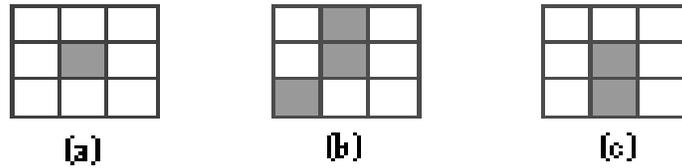


Figura 43: ejemplo de combinaciones de pixeles

La Figura 43(a) muestra un *pixel aislado*, en (b) podemos ver un *pixel de conexión* y en (c) un *pixel final*. En una imagen binaria, tomando una ventana de 3x3 podemos tener 512 posibles combinaciones.

3.8.1 Algoritmos de adelgazamiento secuenciales

Si se emplean algoritmos secuenciales, los puntos de contorno se examinan para eliminarse en un orden predeterminado, esto puede lograrse mediante el “barrido” de la imagen o mediante *algoritmos de seguimiento de contornos*. Los *algoritmos de seguimientos de contornos* pueden visitar solamente los pixeles que se encuentran en el borde del patrón.

Cuando se examina un pixel de contorno p , se elimina o retiene de acuerdo a la configuración de los $N8(p)$. Para lograr la eliminación secuencial de una línea completa en una iteración, un algoritmo secuencial “marca” el pixel a eliminar, y posteriormente, se eliminan todos los pixeles marcados al final de la iteración. Esto, generalmente, garantiza que sólo una capa de pixeles se elimine en cada ciclo.

Un pixel p considerado para eliminarse satisface todas las propiedades siguientes:

- p es un pixel negro.
- p no es un punto aislado o un punto final, por ejemplo, $b(p) \geq 2$.
- p es un pixel de contorno, p.e. existe al menos un pixel blanco en $N4(p)$.

Los algoritmos de adelgazamiento secuenciales pueden emplear el número de cruces (*cruzamientos*) de Rutovitz $X_R(p)$ o de Hildritch $X_H(p)$, para determinar la eliminación de un pixel. Empleando $X_H(p)$ se “barre” el patrón de izquierda a derecha y de arriba hacia abajo, y los pixeles se marcan para eliminarse tomando en cuenta las siguientes cuatro condiciones adicionales:

H1: Al menos un vecino negro de p no debe estar marcado

H2: $X_H(p) = 1$ al inicio de la iteración. Esto significa que el número de veces que se cruza sobre el pixel p es igual a uno.

H3: Si x_3 está marcado (ver Figura 42), se actualiza $x_3=0$, y $X_H(p)$ no cambia. Esto significa que si el pixel superior del punto p ya se había marcado como visitado x_3 se elimina y el número de cruces sobre p se mantiene sin cambios.

H4: Igual que H3, con x_5 sustituyendo a x_3 .

Marcar el pixel p como visitado.

Cuando se emplea $X_R(p)$ para determinar la eliminación de un pixel en un algoritmo tomamos una definición distinta para determinar los *pixeles de contorno*. Un *pixel de contorno* es un pixel negro que tiene al menos un pixel blanco en $N8(p)$. Esta condición junto con el uso de $X_R(p)$ requiere una condición adicional ($F = x_1x_3x_5x_7 = 0$). Para garantizar que no se crean huecos al eliminar el pixel se debe respetar lo siguiente:

- 1 Si $X_R(p) = 0$ u 8 , p no es eliminable.
- 2 Si $X_R(p) = 2$, p es eliminable si y solo si $F = 0$ y p no es un punto final.
- 3 Si $X_R(p) = 4$, p es eliminable si y solo si $F = 0$ y uno de sus cuatro pixeles de esquina es 0 con 1,s a ambos lados, lo cual es equivalente a $\sum_{i=1}^4 x_{2i-1}\bar{x}_{2i}x_{2i+1} = 1$.
- 4 Si $X_R(p) = 6$, p es eliminable si y solo si un pixel en $N4(p)$ es 0 y los otros tres unos pertenecen a diferentes *4-componentes*, o $\sum_{i=1}^4 x_{2i-1} = 3$.

3.9 Extracción del perímetro

Una de las características que define a una lesión (o a cualquier objeto) es su perímetro. El método exacto para la extracción del mismo consiste en buscar un pixel perteneciente al borde del objeto e ir siguiendo la forma de este encontrando los sucesivos vecinos mientras se cuenta. El gran inconveniente es su costo computacional. La morfología presenta una forma más rápida pero inexacta de obtener este perímetro. Basta realizar la resta entre la imagen original y la que resulta después de erosionarla:

$$P(A) = A - (A \ominus B)$$

y contar los pixeles que estan a nivel más alto (los que sobresalen), con lo que se obtiene el denominado perímetro interior. Si restamos la imagen dilatada a la original:

$$P(A) = (A \oplus B) - A$$

obtenemos el perímetro interior.

4 Capítulo 4. Filtrado de imágenes

Los algoritmos de filtrado están ligados a la eliminación de ruido en las imágenes. Esta relación se debe a que podemos considerar como ruido todas aquellas características que no nos interesan o que deseamos eliminar. Para filtrar adecuadamente las imágenes del iris, se realizó una investigación de los tipos de filtro existentes con el fin de utilizar la combinación que mejor se adaptara a las características propias de nuestras imágenes.

Durante el procesamiento de una imagen es común emplear descripciones estáticas de imágenes y sub-imágenes. El concepto de estática está íntimamente relacionado al concepto de distribución de probabilidad. Para una región dada, la cual podemos considerar como una imagen completa, podemos definir la *función de distribución de probabilidad del brillo* en la región, y la *función de probabilidad de densidad del brillo* en la región. Para definir ambas funciones empleamos una imagen digitalizada $a(x,y)$, (ver sección 3.1).

- La *función de distribución de probabilidad del brillo* $P(a)$, es la probabilidad de que el brillo seleccionado de una región sea menor o igual al valor del brillo a . Como a va de $-\infty$ a $+\infty$, $P(a)$ se incrementa de 0 a 1. $P(a)$ es no-decreciente y $\frac{dP(a)}{da} \geq 0$.
- La *función de probabilidad de densidad del brillo* es la probabilidad de que el brillo de una región caiga entre a y $a + \Delta a$, dada la función de distribución de probabilidad $P(a)$, podemos expresarla como $p(a)\Delta a$ donde $p(a)$ es la probabilidad de densidad de brillo:

$$p(a)\Delta a = \left(\frac{dP(a)}{da} \right) \Delta a$$

4.1 Ruido en las imágenes digitales

Todas las imágenes tienen cierta cantidad de ruido que se manifiesta generalmente en píxeles aislados cuyo valor de gris es diferente al de sus vecinos. El

ruido puede clasificarse en cuatro tipos:

- **Gausiano.** Produce pequeñas variaciones en la imagen. Siempre consideramos que el valor final de un pixel sería el ideal más una cantidad correspondiente al error, que puede describirse como una variable gaussiana.
- **Impulsivo (conocido como sal y pimienta).** El valor que toma el pixel no tiene relación con el valor ideal sino con el valor del ruido que toma valores muy altos o muy bajos. Se caracteriza porque el pixel toma un valor máximo, o un valor mínimo si se ha perdido la señal que lo define.
- **Frecuencia.** La imagen obtenida es la suma entre la imagen ideal y otra señal, la interferencia se caracteriza por ser una senoide de frecuencia determinada.
- **Multiplicativo.** La imagen obtenida es fruto de la multiplicación de dos señales.

4.2 Algoritmos de filtrado

Los algoritmos de filtrado son métodos cuyo fin es favorecer alguna característica presente en una imagen o eliminar otra que esté ocultando lo que se busca. Para llegar al objetivo se utilizan diversas operaciones y transformaciones matemáticas. Las **operaciones** que se pueden aplicar a una imagen digital para transformarla en otra (o a otra representación) se clasifican en tres categorías, presentadas en la tabla 1.

Operación	Descripción
Punto	El valor de salida de una coordenada específica, solo depende del valor de entrada de la misma coordenada
Local	El valor de salida de una coordenada específica, depende de los valores de salida de los $N8(p)$ de la misma coordenada
Global	El valor de salida de una coordenada específica, depende de todos los valores de la imagen de entrada

Tabla 1. Operaciones aplicables a una imagen

La Figura 44 muestra gráficamente la clasificación de operaciones considerando a y b como imágenes.

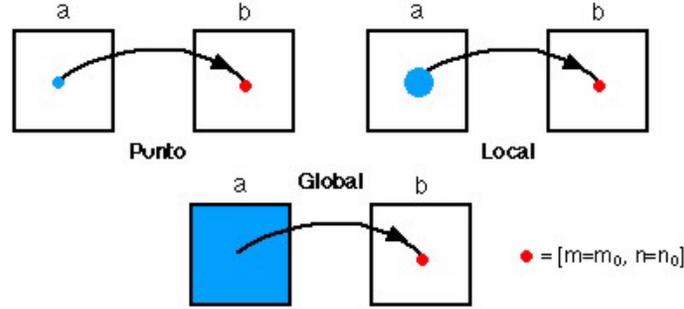


Figura 44: Ejemplo gráfico de operaciones en imágenes

Las transformaciones matemáticas son de dos tipos y pueden llevarse a cabo en el dominio del espacio: *convolución* y *correlación*, y en el dominio de la frecuencia: *transformada de Fourier*.

La *convolución*, denotada comunmente por \otimes , de una función $b(m, n)$ respecto a la función $a(m, n)$ genera una nueva función $c(m, n)$ tal que:

$$c[m, n] = a[m, n] \otimes b[m, n] = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} a[j, k]b[m - j, n - k]$$

Lo más normal es usar convoluciones de 3x3 elementos. Entonces la expresión anterior puede concretarse en:

$$c[m, n] = a[m, n] \otimes b[m, n] = \sum_{j=0}^2 \sum_{k=0}^2 a[j, k]b[m - j, n - k]$$

La *correlación* de $c(m, n)$ entre dos funciones $b(m, n)$ $a(m, n)$ es:

$$c[m, n] = a[m, n] \circ b[m, n] = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} a^*[j, k]b[m + j, n + k]$$

donde a^* es el complejo conjugado, sin embargo, en las imágenes digitales, al ser número reales a^* es igual a a .

La *transformada de Fourier* es una extensión de las series de Fourier² para señales no periódicas. Para ello se toma un nuevo coeficiente c'_k que es igual a:

$$c'_k = T_0 c_k = \int_0^{T_0} f(t) e^{jk\omega_0 t} dt$$

por otro lado:

$$f(t) = \frac{1}{T_0} \sum_{k=-\infty}^{\infty} c'_k e^{jk\omega_0 t} = \frac{\omega_0}{2\pi} \sum_{k=-\infty}^{\infty} c'_k e^{jk\omega_0 t}$$

ya que $\omega_0 T_0 = 2\pi$

Como las señales no son periódicas, su período abarca desde $-\infty$ hasta ∞ por lo que:

$$c'_k = F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega$$

considerando que:

$$\begin{aligned} k\omega_0 &= \omega \\ \omega_0 &\rightarrow d\omega \quad \text{cuando } T_0 \rightarrow \infty \end{aligned}$$

para el caso bidimensional discreto se define como:

$$\begin{aligned} F(u, v) &= \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) e^{-\frac{j2\pi(u x + v y)}{N}} \\ f(x, y) &= \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) e^{\frac{j2\pi(u x + v y)}{N}} \end{aligned}$$

Este tipo de operaciones son utilizadas por los algoritmos de filtrado y aunque existe una gran variedad de algoritmos, podemos hacer una clasificación general de la siguiente manera:

² puede expresarse como $f(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}$

4.2.1 Filtros lineales espaciales:

- 1 de paso alto
- 2 de paso bajo

4.2.2 Filtros no lineales:

- 1 RO (de orden no lineal)
1. Mediana (FM)

4.2.3 Filtros lineales espaciales

El procesamiento espacial lo constituyen aquellas técnicas que operan directamente sobre los valores de los píxeles de una imagen. Los algoritmos de filtrado se emplean para reducir el ruido de una imagen o para prepararla para una segmentación. La diferencia entre los *algoritmos lineales* y *no-lineales* es que los primeros se pueden analizar en el dominio de Fourier y los segundos no. Un *filtro lineal* produce salidas que son combinaciones lineales de la secuencia de entrada. Podemos definirlo matemáticamente de la siguiente forma:

$$y(n) = L_n a$$

donde a es la imagen de entrada, $y(n)$ es la salida en el tiempo n , y L_n es el filtro expresado como el valor real de la función para cada n .

Para transformar una imagen, los *filtros lineales* emplean diferentes funciones. Estas funciones se conocen como *función de difusión de un punto* (PSF - Point Spread Function) y su transformada de Fourier *función de transferencia óptica* (OTF- Optical Transfer Function) o simplemente función de transformación. La tabla 2 muestra algunas transformadas que empleamos en los filtros.

	Función de difusión de un punto (PSF)		Función de transformación (OTF)
T2.1	$R_{a,b}(x, y) = \frac{1}{4ab}u(a^2 - x^2)u(b^2 - y^2)$	F ↔	$\left(\frac{\text{seno}(2\pi a f_x)}{\pi a f_x}\right)\left(\frac{\text{seno}(2\pi b f_y)}{\pi b f_y}\right)$
T2.2	$R_{a,b}(x, y) \otimes R_{a,b}(x, y)$	F ↔	$\left(\frac{\text{seno}(2\pi a f_x)}{\pi a f_x}\right)\left(\frac{\text{seno}(2\pi b f_y)}{\pi b f_y}\right)^2$
T2.3	$P_a(r) = \frac{u(a^2 - r^2)}{\pi a^2}$	F ↔	$2\frac{J_1(2\pi a f)}{\pi a f}$
T2.4	$P_a(r) \otimes P_a(r)$	F ↔	$4\left(\frac{J_1(2\pi a f)}{\pi a f}\right)^2$

Tabla 2. Funciones de Transformación

- | | |
|-----------------|--------------|
| 1. Rectangular | 2. Piramidal |
| 3. Caja pequeña | 4. Cónica |

donde **F** representa la transformada de Fourier y \otimes la convolución (las operaciones $u(*)$ Función paso unidad y $J_1(*)$ Función Bessel están definidas en[22]).

También se hace una clasificación de los filtros lineales entre implementaciones basadas en *soporte rectangular para el filtro*, *soporte circular para el filtro*. En la Figura 45 mostramos un ejemplo de *filtro lineal uniforme*, (en un filtro uniforme la imagen de salida esta basada en un promedio local de la entrada del filtro, donde todos los valores de *soporte* del filtro tienen el mismo peso o valor), en (a) con *soporte rectangular*, y en (b) con *soporte circular*.

$$h_{rec}[j, k] = \frac{1}{25} \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array} \quad h_{cir}[j, k] = \frac{1}{21} \begin{array}{|c|c|c|c|c|} \hline 0 & 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 1 & 1 & 0 \\ \hline \end{array}$$

Figura 45: (a) Filtro rectangular ($J = K = 5$) (b) Filtro circular ($R = 2.5$)

El tamaño del filtro esta dado mediante un rectángulo acotado por $J \times K$ donde, sin perder generalidad, $J \geq K$. El tamaño de la imagen es de $N \times N$. La función de transferencia en la Figura 45 (a) esta definida en T2.1 y para (b) en T2.3.

En un *filtro lineal triangular* la imagen de salida esta basada en un *promedio local* de la entrada del filtro, donde todos los valores de *soporte* del filtro tienen diferentes pesos. En la Figura 45 presentamos ejemplos de filtros *triangulares* con *soporte rectangular* y *circular*.

$$h_{rec}[j, k] = \frac{1}{81} \begin{array}{|c|c|c|c|c|} \hline 1 & 2 & 3 & 2 & 1 \\ \hline 2 & 4 & 6 & 4 & 2 \\ \hline 3 & 6 & 9 & 6 & 3 \\ \hline 2 & 4 & 6 & 4 & 2 \\ \hline 1 & 2 & 3 & 2 & 1 \\ \hline \end{array} \quad h_{cir}[j, k] = \frac{1}{25} \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 2 & 2 & 2 & 0 \\ \hline 1 & 2 & 5 & 2 & 1 \\ \hline 0 & 2 & 2 & 2 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ \hline \end{array}$$

Figura 46: (a) Filtro piramidal ($J = K = 5$) (b) Filtro cónico ($R = 2.5$)

Aunque el *soporte* de los filtros de la Figura 46 (a) y (b) es *rectangular* y *circular*, respectivamente, la *función de transferencia* que se emplea en los *filtros lineales triangulares* es diferente a la de los *filtros lineales uniformes*, por lo que se les denomina *piramidal* en el caso de (a) y su *función de transferencia* esta definida en T2.2 y para (b) *cónico* con su *función de transferencia* definida en T2.4.

4.2.4 Filtros lineales de paso alto

Las características visuales de reconocimiento son críticas para la existencia humana. El reconocimiento de las características de una imagen depende del nivel local de contrastes cercanos a esas características. Uno de los primeros pasos del reconocimiento de una imagen es la *identificación del borde (edge)* o *extracción del contorno*. Para resolver esta tarea, frecuentemente se recurre a esquemas de filtrado de paso alto. Desafortunadamente este esquema de filtrado amplifica el ruido que suele encontrarse en una imagen.

La intensidad local afecta la sensibilidad ocular del ruido en una imagen. Específicamente, el sistema visual humano es mucho menos sensible al ruido en las áreas brillantes de una imagen que en las áreas oscuras. Esta observación se atribuye a la ley de Weber[23]. Una aproximación simple consiste en intensificar o realzar las regiones brillantes en lugar de las regiones oscuras. Un método sencillo para lograr esto es asignar más peso al filtrado

alto haciendo este peso proporcional a la media local (debido a que el ruido es una variación sobre los niveles de gris normales y le corresponden frecuencias altas). Esta clase de filtros se conocen como *mean-weighted highpass filters* [24], [25], [26], [27], [28], [29].

La evidencia empírica también sugiere que el sistema visual es menos sensible al ruido en los contornos (regiones con frecuencias altas en la imagen). Este efecto se conoce como enmascaramiento (*masking*) [27].

Una limitante de este tipo de filtros es que la estructura del filtro está mezclada. Esto significa, que la escala de la media local o detector de contornos está mezclada. El usuario debe especificar una vecindad local para la media o explícitamente definir cuál es la media de un contorno local.

4.2.5 Filtros pesados de paso alto

El método estándar para realzar una imagen se conoce como "*unsharp masking*" [23]. En este método la imagen original se modifica mediante la sustracción de una señal proporcional a una versión reducida de la imagen original. De manera equivalente, una señal proporcional a una versión filtrada de la imagen original puede agregarse a esa misma imagen.

Denotemos al filtro lineal de paso alto mediante H y sea $f(x,y)$ la imagen a manejar. Consideremos

Fórmula 1:

$$g = f + Hf$$

La imagen g resulta de agregar a la imagen filtrada la imagen original. Como resultado se enfatizan los contornos y la estructura general de la imagen. Alternativamente, supongamos que tenemos una imagen con ruido f y un filtro R para restaurarla. Debemos considerar la diferencia entre f y Rf como un filtrado, es decir, $Hf = Rf - f$. Entonces tenemos el mismo método definido en la Fórmula 1.

El realce o el restablecimiento de una imagen g puede ser insatisfactorio si el ruido en la imagen original f es amplificado por H . La ley de Weber y el efecto

de enmascaramiento sugiere el siguiente método para el realce de imágenes:

Sea L un filtro lineal que se “ajusta” a un tipo específico de figura de una imagen “local”. Emplear el término “local” significa que la imagen de salida Lf en el punto (x,y) depende sólo de la vecindad local de f cerca de (x,y) . El enfoque de “ajuste” significa $|Lf(x,y)|$ es grande si las características de una imagen local (como un borde o una región) está cercana a (x,y) en f , un *filtro pesado de paso alto* (*weighted high pass*) se define por:

$$H_w f(x, y) = |Lf(x, y)|^p Hf(x, y), \quad p \geq 1$$

4.2.6 Filtros lineales de paso bajo

Un *filtro de paso bajo* disminuye la ganancia de frecuencias altas, que es donde se encuentra el ruido; el filtro más sencillo e intuitivo es el que tiene coeficientes unidad para todos los elementos.

1	1	1
1	1	1
1	1	1

Figura 47: Coeficientes del filtro con valores unidad

Si multiplicamos el resultado por un noveno obtenemos la media de todos los pixeles, por lo que el ruido disminuye. Sin embargo, éste tipo de filtro presupone que la influencia de todos los pixeles es igual. Otra consideración es, que cuanto más alejado esté un pixel del *pixel central*, su valor será menor. Se tiene entonces la máscara:

1	1	1
1	2	1
1	1	1

Figura 48: Media de los coeficientes

Si se quiere dar más importancia al *pixel central* p que a sus vecinos, y a los $N_4(p)$ que a los $N_8(p)$ (estos conceptos se encuentran definidos en la sección 3.6) la configuración quedaría de la siguiente forma:

1	2	1
2	4	2
1	2	1

Figura 49: Coeficientes para el filtro tomando en cuenta sus vecinos

Especificada de manera general por:

1	b	1
b	b ²	b
1	b	1

Figura 50: Generalización de los coeficientes del filtrado

Debiendo ser la ganancia de todas ellas la unidad para que la imagen no cambie.

Filtros no lineales de paso bajo (RO)

Los *filtros no lineales* para imágenes basados en un orden estático, los cuales se denominan *filtros RO* (de Orden no Lineal), han sido desarrollados para aplicaciones donde el filtrado lineal es inadecuado[30], [31], [32]. Los *filtros lineales* suelen fallar en la preservación de los contornos y suprimen impulsivamente los componentes de ruido. De cualquier forma, los filtros basados en orden estático emplean solamente un *orden lineal* de la secuencia de entrada para producir las salidas. Los *filtros RO* son robustos porque preservan la información de contornos.

El filtro que calcula la mediana *FM*, es aún mejor que los filtros *RO*. Pero independientemente de que es fácil implementarlo y preserva bien los contornos, tiene limitantes en cuanto al manejo del ruido que puede tener repercusiones[33]. Para corregir esta limitante en [30] se presentan filtros *L* basados en combinaciones lineales. Un filtro *L* puede designarse como un criterio óptimo para una gran variedad de distribuciones de ruido. Como caso concreto de filtros *L*, el movimiento promedio de un filtro es óptimo para el ruido Gaussiano y el *FM* es cercano al óptimo de un filtro *L* para el ruido de Laplace.

Una de las desventajas principales de los filtros *L* es que no consideran la correlación temporal de una secuencia de entrada. Como resultado de un

filtrado L , la señal a filtrar se supone como una constante representativa que se corrompe mediante la agregación de ruido y no pueden retenerse los contenidos de espectro de la señal de entrada. Se han realizado múltiples trabajos para poder combinar las características de los filtros lineales y los filtros no lineales basados en ordenes estáticos para restablecer señales de frecuencias selectivas de señales no estacionarias. [33], [34], [35], [36].

5 Capítulo 5. Resumen de redes neuronales y arquitectura tentativa

El proceso de reconocimiento de lesiones se hizo mediante dos métodos diferentes basados en el uso de redes neuronales. El objetivo fue realizar una evaluación comparativa y así decidir las condiciones apropiadas para cada método. Adicionalmente se estudiaron métodos de entrenamiento aplicables a ambas redes. Las redes empleadas son la de Hopfield y el “perceptron” multicapa; a éstas se les denomina “de alimentación directa”. En ambos casos se utilizaron el algoritmo de entrenamiento propio de cada red que consiste, esencialmente, en el ajuste de algunos parámetros internos, sin embargo, existe un algoritmo de entrenamiento basado en el método de “retropropagación” y modificado por Marquardt-Levenberg que puede ser utilizado para ambos enfoques y en general, para redes de *alimentación directa*. Se utilizaron tanto el algoritmo de cada red como el de Marquardt-Levenberg. Se analizaron estas tres redes porque existen demostraciones formales de que funcionan como clasificadores de patrones estáticos[37], [38], [39].

5.1 Antecedentes y Notación básica

Las redes neuronales artificiales están basadas en el comportamiento que tienen las neuronas biológicas. Las redes de neuronas biológicas transmiten señales electroquímicas a través de trayectorias neuronales. Cada neurona recibe señales de otras, mediante una unión llamada *sinapsis*. Algunas señales excitan la neurona y otras tienden a inhibirla. Cuando el efecto acumulativo excede un umbral, la neurona envía una señal hacia otras neuronas. Los modelos de neuronas artificiales exhiben estas sencillas características. Cada neurona artificial recibe un conjunto de entradas y cada una de estas entradas se multiplica por un *peso*, de forma análoga a la sinapsis. La suma de todas las entradas “pesadas” determina el grado de acumulación, al que denominamos *nivel de activación*. Notacionalmente, cada entrada p_i (también suele denotarse por x_i) se modula mediante un peso w_i y la entrada total se expresa como:

$$\sum_i p_i w_i$$

o de forma vectorial, $\mathbf{P} \cdot \mathbf{W}$ donde $\mathbf{P} = [p_1, p_2, \dots, p_n]$ y $\mathbf{W} = [w_1, w_2, \dots, w_n]$. Posteriormente se procesa la señal de entrada mediante una *función de activación* o *transferencia* para producir la señal de salida que se transmite sólo si esta señal es diferente de 0. La *función de transferencia* puede ser una función de límites duros o una función suave como las funciones sigmoidales o tangente hiperbólica (en la Tabla 3 presentada en la siguiente sección mostramos un resumen de las posibles funciones a utilizar en una red).

Una red neuronal se representa gráficamente mediante un conjunto de *nodos* (que pueden ser rectángulos) y flechas, conceptos fundamentales de teoría de gráficas. Un *nodo* corresponde a una neurona, y una flecha corresponde a una *conexión* con una dirección que muestra el flujo entre neuronas, por lo tanto, existen unas neuronas conectadas al sistema de entrada y otras conectadas al sistema de salida (esto puede observarse en la Figura 53 de la siguiente sección). Las redes neuronales pueden reconocer, clasificar, convertir y aprender patrones. Un *patrón* es una descripción cuantitativa o cualitativa de un objeto, un concepto, o un evento. Una *clase de patrones* es un conjunto de patrones que comparten algunas propiedades, en nuestro trabajo generamos una clase de patrones con las lesiones del iris que queremos reconocer. El *reconocimiento de patrones* se refiere a la categorización de datos de entrada en clases identificables mediante el reconocimiento de propiedades significativas o atributos de los datos.

Las redes neuronales pueden tener una arquitectura paralela-distribuida con un gran número de nodos y conexiones. Cada punto de conexión de un nodo a otro se asocia mediante un *peso*. Explicamos de forma detallada estos conceptos y mostramos ejemplos en las secciones de la 5.1.1 a la 5.1.7 La construcción de una red neuronal involucra las siguientes tareas:

- Determinar las propiedades de la red: la *topología (conexidad) de la red* o *arquitectura*, los *tipos de conexiones*, el *orden de las conexiones* y el *rango de los pesos*.
- Determinar las propiedades de los nodos: el *rango de activación* y la *función de activación* o *transferencia*.

- Determinar el sistema dinámico: *el esquema de utilización de los pesos, la fórmula de activación-cálculo y la regla de aprendizaje.*

Propiedades de la red. La topología de la red se refiere a su marco de trabajo o su esquema de interconexión. Normalmente especificamos este marco de trabajo mediante el número de *capas* y el número de *nodos* por capa. Los tipos de capas incluyen:

- La *capa de entrada*: a los *nodos* de esta capa se les denomina *unidades de entrada*, muestran la información que será procesada a través de la red. En nuestro caso las lesiones que deseamos identificar.
- La *capa oculta*: a los *nodos* de esta *capa* se les denomina *unidades ocultas*, no son directamente observables, establecen la no-linearidad de la red.
- La *capa de salida*: aquí llamamos a los *nodos unidades de salida*, cada *unidad de salida* representa una *clase de objeto* o lesión.

En lo que se refiere al esquema de *interconexión*, una red puede ser *alimentación directa* o *recurrente* y sus conexiones pueden ser *simétricas* o *asimétricas*. Mostramos estas definiciones a continuación:

- Redes de *alimentación directa*: todos los puntos de conexión van en una sola dirección (desde la *capa* de entrada hacia la *capa* de salida).
- Redes *recurrentes*: contienen conexiones hacia atrás o ciclos.
- *Conexiones simétricas*: si hay un punto que conecte al nodo i con el nodo j , entonces también existe una conexión desde el nodo j al nodo i , y todos los *pesos* asociados con las dos conexiones son iguales, es decir, $w_{ji} = w_{ij}$.
- *Conexiones asimétricas*: si las conexiones no son *simétricas*, como se definió en el punto anterior, entonces son asimétricas.

Una *conexión de orden-superior*, es una conexión de combina entradas de más de un nodo, por multiplicación. El número de entradas determina el orden de la conexión. El *orden* de una red neuronal es la *conexión de orden-superior*. Asumimos que una red neuronal es de primer orden, a no ser que se mencione otra cosa.

Las conexiones de los pesos pueden ser números reales o enteros. Pueden estar confinados a un rango. Son ajustables durante el proceso de entrenamiento de la red, pero a algunos se le puede modificar su valor de forma deliberada. Cuando el proceso de entrenamiento de la red concluye, todos los pesos se han modificado.

Propiedades de los nodos. Los niveles de activación de los nodos pueden ser discretos (0 ó 1) o continuos en el rango [0,1]. Esto depende de la *función de transferencia* que se utilice. Si es una *función de límite-duro*, entonces los *niveles de activación* están acotados por los valores de 0 (o -1) y 1. Para una función sigmoideal, los *niveles de activación* están limitados a un rango continuo de reales [0,1].

Sistema dinámico. El esquema de inicialización de los *pesos* es específico del modelo particular de la red neuronal elegida. En la mayoría de los casos los *pesos* son valores aleatorios pequeños en el rango de los número reales.

La *regla de aprendizaje* es uno de los atributos más importantes que debe ser especificado en la red neuronal. La *regla de aprendizaje* determina cómo adaptar las conexiones de los *pesos* para optimizar su rendimiento e indica cómo calcular los valores de ajuste de los *pesos* durante cada ciclo del proceso de entrenamiento.

Las siguientes secciones muestran ejemplos más detallados de todos los conceptos que tratamos.

5.1.1 Neurona de una sola entrada.

En una red neuronal que tenga una neurona y un solo valor de entrada (mostrada en la Figura 51) la entrada escalar p se multiplica por el *peso* es-

calar w para formar wp , uno de los términos se envía a la sumatoria. La otra entrada, 1, se multiplica por el *sesgo o umbral* b y se envía a la sumatoria. El valor de salida n pasa a través de una *función de transferencia* f , lo que produce la salida escalar a de la neurona.

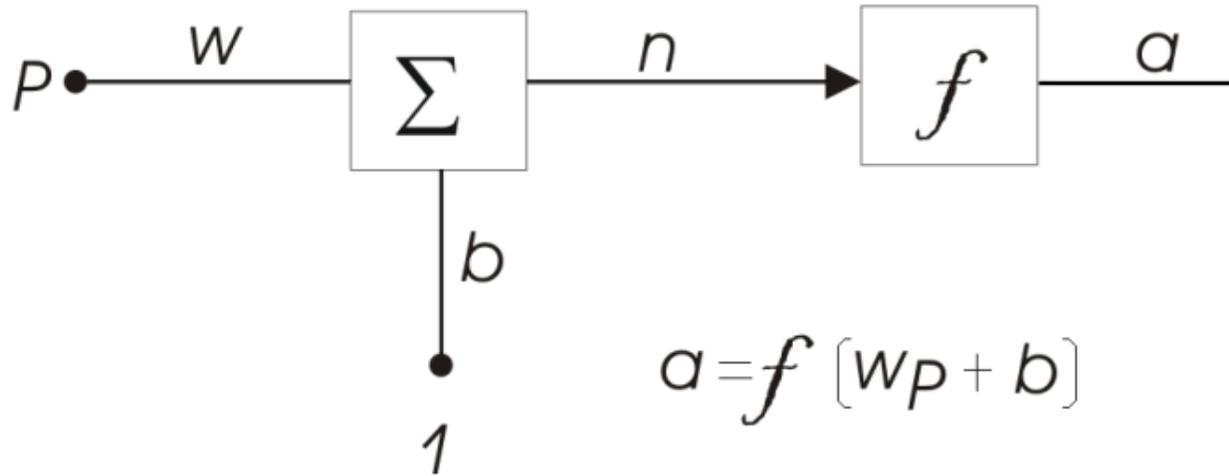


Figura 51: Neurona de una sola entrada

Las redes neuronales utilizan funciones para definir el valor que tomará la salida. En la Tabla 3 presentamos un resumen de las *funciones de transferencia* que pueden emplearse y la Figura 52 muestra la función de *límite duro* en el plano.

Nombre	Relación Entrada/Salida	Gráfica
Límite Duro	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$	
Límite Duro Simétrico	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$	
Lineal	$a = n$	
Saturación lineal	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$	
Saturación lineal Simétrica	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$	
Logaritmo sigmoideal	$a = \frac{1}{1+e^{-n}}$	
Tangente hiperbolica sigmoideal	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
Lineal positiva	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$	
Competitiva	$a = 1$ neurona con máximo n $a = 0$ todas las demás neuronas	

Tabla 3: Funciones de transferencia

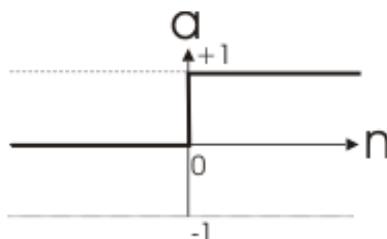


Figura 52: Diagrama de la función de Límite duro

Generalmente, las redes neuronales tienen neuronas con más de un valor de entrada. La Figura 53 muestra una neurona con R entradas, a las entradas

individuales p_1, p_2, \dots, p_R se les asigna un peso correspondiente a su elemento $w_{1,1}, w_{1,2}, \dots, w_{1,R}$ de una matriz de pesos \mathbf{W} . [40]

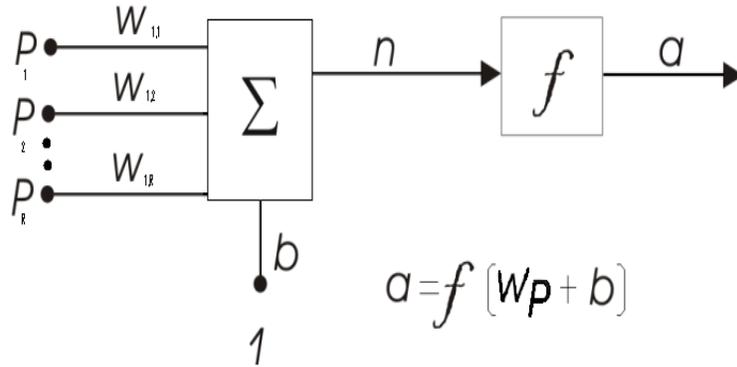


Figura 53: Neurona de R entradas

La neurona tiene un umbral b , que se suma junto con las entradas de los pesos para formar el valor de entrada de la red n :

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b$$

Esta expresión puede escribirse en forma matricial de la siguiente forma:

$$n = \mathbf{W}\mathbf{p} + b$$

donde la matriz \mathbf{W} para el caso de una neurona de una sola entrada tiene un solo renglón, por lo tanto, podemos escribir la salida de la neurona como:

$$a = f(\mathbf{W}\mathbf{p} + b)$$

5.1.2 Índices de los valores de los pesos.

La convención empleada en la asignación de índices de los elementos de la matriz de pesos es la siguiente: el primer índice indica a qué neurona se le aplica ese *peso*. El segundo índice indica la fuente de la señal de alimentación (valor de p) de la neurona. Por lo tanto, los índices en $w_{1,2}$ indican que este peso representa la conexión para la primer neurona desde la entrada p_2 . Esta convención es útil cuando tenemos una red neuronal con más de una neurona.

5.1.3 Arquitecturas de red.

Capas. En una red con una sola capa de S neuronas, como la mostrada en la Figura 54, cada una de las R entradas se conecta a cada neurona y la *matriz de pesos* ahora tiene S renglones.

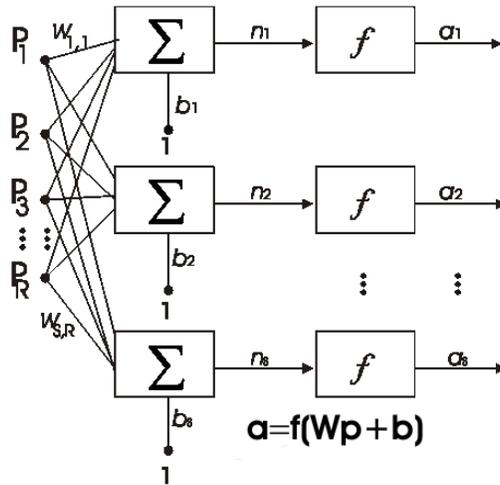


Figura 54: Capa de S neuronas

La *capa* incluye la *matriz de pesos*, las sumatorias, el *vector de sesgo* \mathbf{b} , la *función de transferencia* y el *vector de salida* \mathbf{a} . Cada elemento del *vector de entrada* \mathbf{p} se conecta a cada neurona a través de la *matriz de pesos* \mathbf{W} . Cada neurona tiene un *sesgo* b_i , una sumatoria, una *función de transferencia* f y una salida a_i . Agrupando las salidas, formamos el *vector de salida* \mathbf{a} . Es común que el número de entradas a una capa sea diferente del número de neuronas, es decir que $R \neq S$. Y las funciones de transferencia pueden ser diferentes en una misma capa.

Si empleamos varias capas en una red, utilizamos un super-índice para especificar el número de capa. Este número se agrega a los pesos $w_{i,j}^c$, los valores de los sesgos b_i^c , las entradas a la red n_i^c , las funciones de transferencia f^c y los valores de salida a_i^c , donde c representa el número de capa, se convierten en los valores de entrada para cada neurona de la siguiente capa o la salida de toda la red si c representa la última capa.

5.1.4 Reglas de aprendizaje de las redes neuronales.

La *regla de aprendizaje* de una red neuronal se refiere al procedimiento mediante el cual se modifican los valores de los *pesos* y los valores de los *sesgos* de una red, a este procedimiento también se le conoce como *algoritmo de entrenamiento*. El propósito de la *regla de aprendizaje* es entrenar la red para optimizar o mejorar alguna tarea. Existen muchos tipos de *reglas de aprendizaje* de las redes. Estos tipos caen en tres categorías: *aprendizaje supervisado*, *aprendizaje no-supervisado* y *aprendizaje reforzado* o *gradiente*.

5.1.5 Aprendizaje supervisado

La *regla de aprendizaje* esta basada en un conjunto de ejemplos (el conjunto de entrenamiento) del comportamiento propio de la red:

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \dots, \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

donde \mathbf{p}_q es una entrada de la red y \mathbf{t}_q es su correspondiente salida correcta (salida objetivo). Cuando las entradas se aplican a la red, las salidas de la red se comparan con las salidas objetivo. La regla de aprendizaje se utiliza para ajustar los *pesos* y los *sesgos* para “mover” las salidas de la red cerca de la salida objetivo.

5.1.6 Aprendizaje no-supervisado

En este tipo de aprendizaje, los *pesos* y los *sesgos* se modifican en respuesta solamente de las entradas de la red, sin considerar salidas objetivo.

5.1.7 Aprendizaje reforzado o gradiente

Es similar al *aprendizaje supervisado*, excepto que, en lugar de proveer la salida correcta para cada entrada de la red, el algoritmo solamente da un *grado*. El *grado* (o *puntuación*) es una medida del rendimiento de la red sobre alguna secuencia de entradas.

5.2 Ejemplos de redes neuronales

5.2.1 Red de Hopfield

La red de Hopfield es *supervisada* y normalmente toma como entrada datos binarios[41]. Lo más apropiado para esta red son entradas en representación binaria exacta de imágenes en blanco y negro (-1, 1). Cuando las entradas son continuas (como las imágenes del iris), existe un problema fundamental de representación que se resuelve convirtiendo los valores analógicos a valores binarios. Topológicamente esta red es totalmente conexa, es decir, cada neurona contiene una liga con las demás neuronas existentes[42].

La red de Hopfield tiene la capacidad de aprender patrones y reconocerlos aún cuando estos patrones sean difusos[43]. Esta red actualiza los patrones mediante la comparación del patrón dado contra los que tiene almacenados y se elige el patrón que tenga mayor similitud, es decir, aquel con el que existe mayor coincidencia entre píxeles. Como consecuencia es imposible, en general, realizar el reconocimiento cuando las imágenes de prueba³ corresponden a rotaciones o traslaciones del patrón original.

Para la identificación de lesiones se empleó una red totalmente conexa y su número de neuronas estuvo determinado por la dimensión de la matriz con los datos de entrada, éstos debieron convertirse previamente de forma continua a discreta. Una vez definida la arquitectura de la red, con los datos que se mencionaron, se hizo el entrenamiento el algoritmo de aprendizaje de Hopfield.

5.2.2 Algoritmo de la red de Hopfield

El paso 1 del algoritmo que se presenta a continuación es el proceso de entrenamiento de la red, los patrones que debe aprender se le presentan uno a uno, se les asigna un peso a los bits de la muestra y finalmente se almacenan. La matriz entrenamiento t_{ij} representa la magnitud de la correlación entre

³Las imágenes de prueba son aquellas que se desea reconocer una vez que la red ha sido entrenada.

los patrones i y j (indica qué tan similares son las muestras -patrones- entre sí). Cuando t_{ij} es grande y con valor positivo, significa que una imagen de un patrón de prueba que tenga en el componente i -ésimo el valor a (1 o -1), entonces es altamente probable que el componente j -ésimo tenga valor a . Pero si T_{ij} es grande y con valor negativo, entonces, si el elemento i de una imagen de prueba tiene valor a entonces es altamente probable que el elemento j -ésimo tenga valor $-a$. N es la la dimensión del vector de entrada y M es el número de elementos que tiene la muestra. Una vez que la red ha sido entrenada, comienza el proceso de reconocimiento mediante la presentación del patrón a identificar (paso 2) y se compara bit a bit contra las similitudes que tiene almacenadas la matriz $t_{i,j}$ (paso3). La salida de la red es la muestra con la que el patrón a identificar tiene mayor coincidencia.

1. Asigna pesos a las conexiones

$$t_{ij} = \begin{cases} \sum_{s=0}^{M-1} x_i^s x_j^s & i \neq j \\ 0 & i = j, 0 \leq i, j \leq N - 1 \end{cases}$$

En esta fórmula t_{ij} es el peso de la conexión desde el nodo i al nodo j y x_i^s , que puede ser +1 ó -1, es el elemento i del patrón s .

2. Paso inicial para identificar un patrón desconocido

$$\mu_i(0) = x_i, 0 \leq i \leq N - 1$$

En esta fórmula $\mu_i(t)$ es la salida del nodo i en el tiempo t y x_i , que puede ser +1 o -1 es el elemento i del patrón de entrada.

3. Itera hasta converger

$$\mu_j(t+1) = f_h \left[\sum_{i=0}^{N-1} t_{ij} \mu_i(t) \right], 0 \leq j \leq N - 1$$

La función f_h es la función de *límite duro simétrico*, presentada en la Tabla 3. El proceso se repite hasta que los nodos de salida se mantengan sin cambio durante varias iteraciones. Los nodos de salida representan el ejemplar del patrón que mejor coincidencia tiene con la

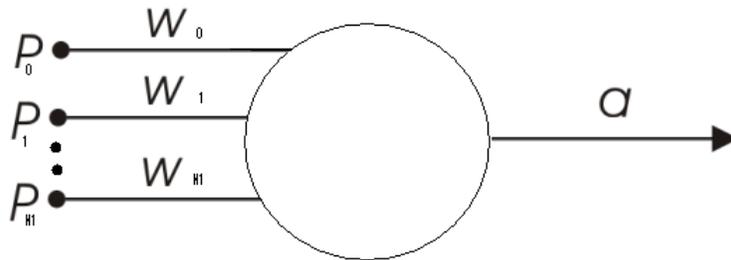
entrada (patrón) desconocida.

4. Se repite a partir del paso 2

5.3 Perceptron

5.3.1 Perceptron de una sola capa

El perceptron de una sola capa[44] es interesante porque tiene la habilidad de aprender a reconocer patrones simples. La Figura 55- muestra un perceptron que puede decidir si una entrada pertenece a una de dos clases (denotadas por A y B). La neurona calcula la sumatoria de los pesos con los valores de entrada correspondientes, le resta un valor θ (llamado *umbral*) y envía el resultado a una función de límite duro, que decide si la salida toma el valor de +1 o -1. La regla de decisión indica que el patrón corresponde a la clase A si el valor de salida es +1 y a la clase B si el valor de salida es -1. El perceptron forma dos regiones de decisión separadas por un hiperplano. Estas regiones se muestran en la Figura 56. Existen solamente dos salidas y el hiperplano es una línea. En este caso las salidas que se encuentren arriba de la línea límite corresponden a la clase A y las que se encuentren debajo pertenecen a la clase B. La ecuación de la línea límite dependerá de las conexiones de los pesos y del valor del umbral.



$$a = f_h(\sum_{i=0}^{N-1} w_i p_i - \theta) \quad a = \begin{cases} +1 \Rightarrow \text{clase A} \\ -1 \Rightarrow \text{clase B} \end{cases}$$

Figura 55:

Perceptron de una capa

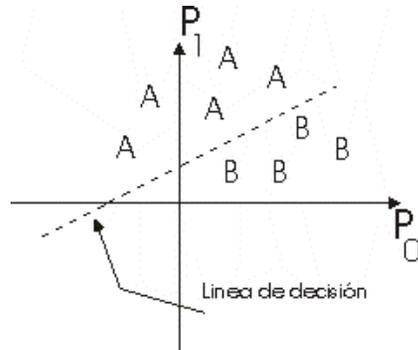


Figura 56: Regiones en el hiperplano

Las *conexiones de los pesos* y el *valor del umbral* del perceptron pueden “adaptarse” o “arreglarse” empleando diferentes algoritmos. El *algoritmo de convergencia* original del perceptron para ajustar *pesos*, es el siguiente: Como primer paso se inicializan los valores de los pesos y el valor del umbral con números aleatorios pequeños que sean diferentes de 0. El nuevo valor de entrada formado de N elementos continuos se toma como entrada a la neurona y la salida se calcula como en la Figura 56. Las conexiones de los pesos se adaptan, sólo cuando ocurre un error, empleando la fórmula que aparece en el paso 4 del *procedimiento de convergencia del perceptron*. Esta fórmula incluye un término de ganancia (η) que se encuentra en el rango de 0.0 a 1.0 y controla la proporción de la adaptación. Este término de ganancia debe ajustarse para satisfacer los requerimientos conflictivos de una rápida adaptación de los cambios reales de las distribuciones y promedio de entradas anteriores, para permitir hacer estimaciones estables de los valores de los pesos.

5.3.2 Procedimiento de convergencia del “perceptron”

1. Inicializa los pesos y el umbral

Inicializar $w_i(0)$ ($0 \leq i \leq N - 1$) y θ a valores aleatorios pequeños. Aquí $w_i(t)$ es el peso de la entrada i en el tiempo t y θ es el umbral en el nodo de salida.

2. Se presenta una nueva entrada y la salida deseada

Se presentan nuevos valores continuos como entrada $p_0, p_1 \dots p_{N-1}$ en conjunto con la salida deseada $d(t)$.

3. Calcula la salida actual

$$a(t) = f_h \left(\sum_{i=0}^{N-1} w_i(t) p_i(t) - \theta \right)$$

4. Adapta los pesos

$$w_i(t+1) = w_i(t) + \eta[d(t) - a(t)]p_i(t), 0 \leq i \leq N-1$$

$$d(t) = \begin{cases} +1 & \text{si la entrada corresponde a la clase A} \\ -1 & \text{si la entrada corresponde a la clase B} \end{cases}$$

En estas ecuaciones η es una fracción de ganancia positiva menor que 1 y $d(t)$ es la salida correcta deseada de la entrada actual.

5. Repetir a partir del paso 2.

5.3.3 Perceptron multicapa

El “*Perceptron*” *multicapa* es una red neuronal *supervisada* que puede tomar como entrada valores continuos o discretos (binarios) [45] y es indiscutiblemente la arquitectura de red neuronal más utilizada. El “*perceptron*” *original* (o simplemente “*perceptron*”) tiene tres capas de neuronas: unidades sensoriales, unidades asociativas y la unidad de respuesta (Figura 57), a la capa de unidades asociativas se le conoce como *capa oculta*. Un “perceptron” emplea activaciones binarias para las unidades sensoriales y asociativas y una activación de +1, 0 ó -1 para la unidad de respuesta[46]. A las conexiones entre las unidades sensoriales y las asociativas se le asignan aleatoriamente *pesos* fijos que toman valores de +1, 0 ó -1.

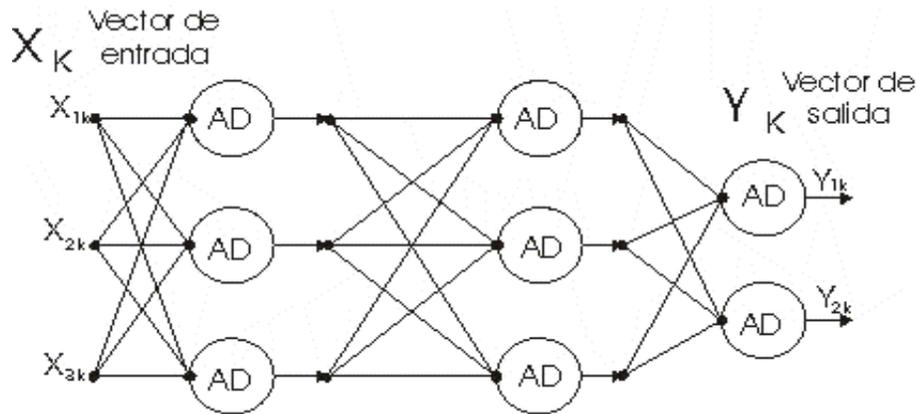


Figura 57: Red neuronal *directa* de 3 capas

La *función de activación* para cada unidad asociativa es la *función binaria escalonada* con un *umbral* θ arbitrario, por lo tanto, el conjunto de señales que van desde las unidades asociativas hacia la unidad de salida (respuesta) es una señal binaria (0 o 1).

Los *pesos* desde las *unidades asociativas* hacia la *unidad de respuesta* se ajustan mediante la regla de aprendizaje (procedimiento de convergencia) del “perceptron”. Para cada patrón de entrenamiento que se presente como entrada, la red debe calcular la respuesta que dará la unidad de salida determinando si ocurrió o no un error en el patrón dado (mediante la comparación de la salida que fue calculada contra el valor deseado). Una señal de error denota que los pesos deben alterarse en la dirección indicada por el valor actual deseado. Deberán ajustarse solamente los pesos de las conexiones que envíen una señal diferente de 0 hacia la unidad de salida puesto que estas señales contribuyen al error. En la Figura 57 las neuronas con la letra AD indican que son adaptativas, basadas en los valores de los *pesos*.

5.4 Algoritmo de entrenamiento Marquardt-Levenberg

Desde que el algoritmo de entrenamiento *retropropagación* [47] se presentó por primera vez, hubo considerables investigaciones y han aparecido un gran número de métodos para acelerar la convergencia de este algoritmo. Las investigaciones realizadas caen en dos categorías. La primera involucra los

desarrollos de técnicas ad hoc. Estas técnicas incluyen ideas tales como la variación del tiempo de aprendizaje. La segunda categoría de investigación está enfocada a técnicas de optimización numérica estándar.

Los enfoques más populares de la segunda categoría, utilizan los métodos del gradiente conjugado o quasi-Newton (secante)[48]. Los métodos quasi-Newton son considerablemente más eficientes, pero sus requerimientos de almacenamiento y computo están arriba del cuadrado del tamaño de la red. Mientras que *retropropagación* es un algoritmo de pasos descendentes, el algoritmo de Marquardt-Levenberg [49] es una aproximación al método de Newton. Para comprender el funcionamiento de este algoritmo se presentará primero el algoritmo de entrenamiento de *retropropagación* y posteriormente la modificación introducida por Marquardt.

5.4.1 Algoritmo retropropagación

Considere una red directa multicapa como la presentada en la Figura 58.

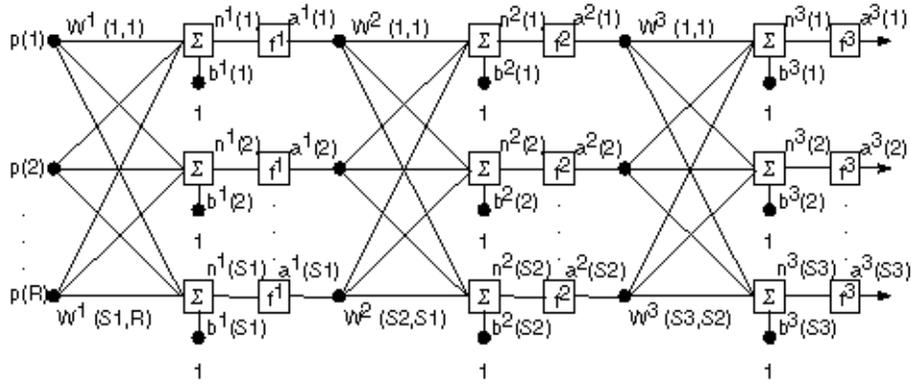


Figura 58: Red neuronal “feedforward”

La entrada de la red del nodo i en la capa $k + 1$ es

$$n^{k+1}(i) = \sum_{j=1}^{S_k} \omega^{k+1}(i, j) a^k(j) + b^{k+1}(i). \quad (1)$$

donde S es el número de nodos (neuronas) en la k -ésima capa. La salida para el nodo i deberá ser:

$$a^{k+1}(i) = f^{k+1}(n^{k+1}(i)). \quad (2)$$

Para una red de M capas, el sistema de ecuaciones en forma matricial está dado por:

$$\vec{a}^0 = \vec{p} \quad (3)$$

$$\underline{a}^{k+1} = \underline{f}^{k+1}(W^{k+1}\underline{a}^k + \underline{b}^{k+1}), k = 0, 1, \dots, M - 1. \quad (4)$$

La tarea de la red es aprender asociaciones entre conjuntos específicos de parejas de entradas-salidas $\{(\underline{p}_1, \underline{t}_1), (\underline{p}_2, \underline{t}_2), \dots, (\underline{p}_Q, \underline{t}_Q)\}$, donde Q es el número de muestras o patrones de aprendizaje.

El índice de rendimiento (especificado en la *regla de aprendizaje reforzado* o *gradiente*) para la red es:

$$V = \frac{1}{2} \sum_{q=1}^Q (\underline{t}_q - \underline{a}_q^M)^T (\underline{t}_q - \underline{a}_q^M) = \frac{1}{2} \sum_{q=1}^Q \underline{e}_q^T \underline{e}_q. \quad (5)$$

donde \underline{a}_q^M es la salida de la red cuando la q -ésima entrada \underline{p}_q se presenta, y $\underline{e}_q = \underline{t}_q - \underline{a}_q^M$ es el error para la q -ésima entrada. Para el algoritmo estandar de retropropagación se utiliza una regla de aproximación por pasos descendentes. El *índice del rendimiento* está dado en forma aproximada por

$$\hat{V} = \frac{1}{2} \underline{e}_q^T \underline{e}_q. \quad (6)$$

donde la suma total de los cuadrados se sustituye por el error cuadrático de un solo par entrada/salida. El algoritmo de aproximación de pasos descendentes (gradiente) es:

$$\Delta\omega^k(i, j) = -\alpha \frac{\partial \hat{V}}{\partial \omega^k(i, j)} \quad (7)$$

$$\Delta b^k(i) = -\alpha \frac{\partial \hat{V}}{\partial b^k(i)} \quad (8)$$

donde α es el factor de ganancia, en el rango $0 \leq \alpha \leq 1$ [45]. Se define

$$\delta^k(i) \equiv \frac{\partial \hat{V}}{\partial n^k(i)} \quad (9)$$

como la sensibilidad del índice de rendimiento a los cambios en la entrada de la red del nodo i en la capa k . Ahora se demostrará, empleando (1), (6) y (9), que

$$\frac{\partial \hat{V}}{\partial \omega^k(i,j)} = \frac{\partial \hat{V}}{\partial n^k(i)} \frac{\partial n^k(i)}{\partial \omega^k(i,j)} = \delta^k(i) a^{k-1}(j) \quad (10)$$

$$\frac{\partial \hat{V}}{\partial b^k(i)} = \frac{\partial \hat{V}}{\partial n^k(i)} \frac{\partial n^k(i)}{\partial b^k(i)} = \delta^k(i). \quad (11)$$

También puede demostrarse que la sensibilidad satisface la siguiente relación de recurrencia

$$\underline{\delta}^k = \dot{F}^k(\underline{n}^k) W^{k+1T} \underline{\delta}^{k+1} \quad (12)$$

donde

$$\dot{F}^k(\underline{n}^k) = \begin{bmatrix} \dot{f}^k(n^k(1)) & 0 & \cdots & 0 \\ 0 & \dot{f}^k(n^k(2)) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \dot{f}^k(n^k(Sk)) \end{bmatrix} \quad (13)$$

y

$$\dot{f}^k(n) = \frac{df^k(n)}{dn}. \quad (14)$$

Esta relación de recurrencia se inicializa en la última capa

$$\underline{\delta}^M = -\dot{F}^M(\underline{n}^M)(\underline{t}_q - \underline{a}_q). \quad (15)$$

El algoritmo de aprendizaje procede como sigue: primero, propaga la entrada hacia adelante empleando (3) - (4); posteriormente, la propaga hacia atrás con (15) y (12); y finalmente, actualiza los pesos con (7), (8), (10) y (11).

5.4.2 Modificación de Marquardt-Levenberg

Supongamos que tenemos una función $V(\underline{x})$ que queremos minimizar con respecto al parámetro del vector \underline{x} , entonces el método de Newton será

$$\Delta \underline{x} = - [\nabla^2 V \underline{x}]^{-1} \nabla V(\underline{x}). \quad (16)$$

donde $\nabla^2 V \underline{x}$ es la matriz Hessiana y $\nabla V \underline{x}$ es el gradiente. Si asumimos que $V \underline{x}$ es una suma de funciones cuadráticas

$$V(\underline{x}) = \sum_{i=1}^N e_i^2(\underline{x}) \quad (17)$$

puede mostrarse que

$$\nabla V(\underline{x}) = J^T(\underline{x}) \underline{e}(\underline{x}) \quad (18)$$

$$\nabla^2 V(\underline{x}) = J^T(\underline{x}) J(\underline{x}) + S(\underline{x}) \quad (19)$$

donde $J(\underline{x})$ es la matriz jacobiana

$$J(\underline{x}) = \begin{bmatrix} \frac{\partial e_1(\underline{x})}{\partial x_1} & \frac{\partial e_1(\underline{x})}{\partial x_2} & \dots & \frac{\partial e_1(\underline{x})}{\partial x_n} \\ \frac{\partial e_2(\underline{x})}{\partial x_1} & \frac{\partial e_2(\underline{x})}{\partial x_2} & \dots & \frac{\partial e_2(\underline{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N(\underline{x})}{\partial x_1} & \frac{\partial e_N(\underline{x})}{\partial x_2} & \dots & \frac{\partial e_N(\underline{x})}{\partial x_n} \end{bmatrix} \quad (20)$$

y

$$S(\underline{x}) = \sum_{i=1}^N e_i(\underline{x}) \nabla^2 e_i(\underline{x}). \quad (21)$$

Para el método de Gauss-Newton se asume que $S(\underline{x}) \approx 0$, y la actualización de la ecuación (16) será

$$\Delta \underline{x} = [J^T(\underline{x})J(\underline{x})]^{-1} J^T(\underline{x})\underline{e}(\underline{x}). \quad (22)$$

La modificación de Marquardt-Levenberg al método de Gauss-Newton es:

$$\Delta \underline{x} = [J^T(\underline{x})J(\underline{x}) + \mu I]^{-1} J^T(\underline{x})\underline{e}(\underline{x}). \quad (23)$$

El parámetro μ se multiplica por algún factor (β) y cualquier paso provoca un incremento en $V(\underline{x})$. Cuando un paso disminuye a $V(\underline{x})$, μ se divide entre β . Note que cuando μ es grande el algoritmo se vuelve de pasos desendentes (con paso $1/\mu$), mientras que para μ pequeña, el algoritmo se convierte en el de Gauss-Newton.

El paso clave en este algoritmo es el calculo de la matriz jacobiana. Para el problema de coincidencia con la red neuronal, los términos en la matriz jacobiana pueden calcularse por una modificación simple al algoritmo retropropagación. El índice de rendimiento para este problema está dado por (5). Es fácil ver que es equivalente en su forma a (17), donde $\underline{x} = [\omega^1(1, 1)\omega^1(1, 2) \cdots \omega^1(S1, R)b^1(1) \cdots b^1(S1)\omega^2(1, 1) \cdots b^M(SM)]^T$, y $N = Q \times SM$. El algoritmo de retropropagación estándar calcula los términos como

$$\frac{\partial \hat{V}}{\partial \omega^k(i, j)} = \frac{\partial \sum_{m=1}^{SM} e_q^2(m)}{\partial \omega^k(i, j)}. \quad (24)$$

Para los elementos de la matriz jacobiana en el algoritmo de Marquardt es necesario calcular los términos como

$$\frac{\partial e_q(m)}{\partial \omega^k(i, j)}. \quad (25)$$

Estos términos pueden calcularse empleando el algoritmo estándar de retropropagación con una modificación en la última capa

$$\Delta^M = -\dot{F}^M(\underline{n}M). \quad (26)$$

Se puede ver que cada columna en la matriz en (26) es un vector sensible jacobiana que debe propagarse hacia atrás a través de la red para producir

un renglón en la jacobiana.

Resumen del método

La modificación de Marquardt al algoritmo de retropropagación procede como sigue:

1. Presentar todas las entradas a la red y calcular sus salidas correspondientes (empleando (3) y (4)), y los errores ($e_q = t_q - a_q^M$). Calcular la suma de los cuadrados de los errores sobre todas las entradas ($V(x)$).
2. Calcular la matriz jacobiana (empleando (26), (12), (10), (11) y (20)).
3. Resolver (23) para obtener Δx .
4. Recalcular la suma cuadrática de los errores empleando $x + \Delta x$. Si esta nueva suma de cuadrados es menor que la calculada en el paso 1, entonces reducir μ mediante β , actualizar $x = x + \Delta x$, y regresar al paso 1. Si la suma de los cuadrados no se reduce, entonces incrementar μ mediante β e ir al paso 3.
5. Se asume que el algoritmo converge cuando la norma del gradiente (18) es menor que algún valor predeterminado, o cuando la suma de los cuadrados se reduce a alguna meta de error.

Las redes “perceptron” y Hopfield son redes directas cuyo número de nodos se definió una vez que se especificó la longitud de la matriz que contiene los patrones de entrenamiento. Esta matriz como se explicó en la descripción del proyecto, es derivada de la segmentación de la imagen del iris. De esta manera, una vez que se defina la arquitectura de la red directa se probó con el algoritmo de entrenamiento de Marquardt-Levenberg.

6 Capítulo 6. Algoritmos de tratamiento de la imagen del iris

Con el fin de obtener componentes independientes extraídos del iris que representen las lesiones que contiene la imagen y que constituyan los patrones para el entrenamiento de la red, aplicamos varios algoritmos de manipulación de imágenes:

- Conversión de la imagen a escala de grises
- Normalización de los valores que componen la imagen
- Filtrado de paso bajo para la imagen del iris
- Descomposición y conversión de la imagen a valores de 0 y 1
- Adelgazamiento de los elementos encontrados que contengan valores de uno
- Obtención de componentes conexas en elementos individuales (lesiones encontradas)

Una vez que se aplican todos los algoritmos de manipulación de la imagen del iris mencionados, tenemos un conjunto de matrices de tamaño 6 x 6 que conforman las componentes conexas de la imagen del iris. Estas matrices nos permitieron entrenar varias redes neuronales. El algoritmo de entrenamiento de la red seleccionada se presenta al final del capítulo. El resultado del entrenamiento de la red se almacenó en un archivo de texto que puede cargarse para reconocimientos posteriores.

Para especificar a qué parte del cuerpo afecta alguna lesión, se identificó la pupila posicionándonos al centro de la imagen del iris, esto permite ubicarla para posteriormente emplear un algoritmo de *seguimiento de contornos*[50], [51], [52], que nos indica el área que ocupa, estos algoritmos fueron necesarios debido a que el tamaño de la pupila varía de acuerdo a la iluminación con que fue tomada la fotografía del iris. A partir de la localización de la

pupila, dividimos toda la imagen del iris en 12 secciones que corresponden a los ángulos 0° , 30° , 60° , etc. De acuerdo con las técnicas iridológicas del capítulo 2.

6.1 Algoritmos de manipulación de la imagen del iris previos al entrenamiento de la red

6.1.1 Conversión de la imagen a escala de grises

Las imágenes del iris con las que contamos se encuentran en un espacio *RGB*, este espacio está basado en la combinación de tres señales de luminocidad cromática distinta: el rojo, el verde y el azul (Red, Green, Blue). Para realizar la conversión a tonos de gris primero podemos modificar el contraste. Denotemos a la imagen del iris mediante una matriz X de dimensiones $M \times N$, para filtrarla debe convertirse a tonalidades de grises y posteriormente a valores de 0 y 1. Esto permite trabajar con rangos de frecuencias menores a los que se emplean cuando la imagen es a colores. Para convertir la imagen X de color a escala de grises se deben eliminar los matices y la información de la saturación mientras se mantiene la iluminación. Los valores en X están en el rango 0 (negro) a 1 (blanco).

La conversión a escala de grises es una de las técnicas más sencillas de realce de características de una imagen que puede realizarse empleando la siguiente función:

$$y = f(x)$$

donde x : representa los datos de la entrada original y y : la salida de datos convertida

La conversión más sencilla es la *lineal*, mostrada en el plano mediante la Figura 59 y emplea la siguiente fórmula para la *modificación del contraste*:

$$Y = \frac{y_{\max} - y_{\min}}{X_{\max} - X_{\min}} (X - X_{\min}) + y_{\min}$$

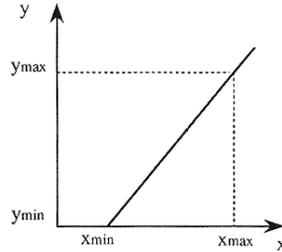


Figura 59: Ubicación de los elementos en el plano para la modificación del contraste

Para eliminar la *luminocidad* y el *color* de las imágenes del iris las transferimos del espacio *RGB* al *YIQ* para el sistema *NTSC* (relacionado con los sistemas de color para la TV), el objetivo es obtener *Y* que contiene la iluminación correspondiente a la escala de grises. La transformación emplea la siguiente matriz:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & -1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

Por lo que el algoritmo simplemente consiste en la división de dos matrices de dimensión $N \times 3$:

$$yiq = rgb/T$$

donde \tilde{N}

T es una matriz de dimensión 3×3 que contiene valores constantes que representan los colores rojo, verde y azul.

rgb : contiene la información de los colores de la imagen en el espacio HSV^4 , es una matriz de N renglones por 3 columnas que contiene valores entre 0 y 1, por ejemplo: $[0 \ 0 \ 0]$ para el negro, $[1 \ 1 \ 1]$ para el blanco,

⁴ *Hue Saturation Value*

[1 0 0] para el rojo puro , [.5 .5 .5] para el gris.

$yiiq$ contiene la escala de gris que se aplica a los elementos de la imagen original para lograr la conversión.

6.1.2 Algoritmo de normalización de la imagen de iris

Una vez convertida la imagen del iris a escala de grises la normalizamos mediante el un sencillo algoritmo tomando valores mínimos y máximos. El algoritmo empleado se presenta a continuación:

1. Asignar a las variables enteras min y max el valor de 0 y 1 respectivamente
2. Obtener en m y n las dimensiones de la matriz de la imagen f
3. Para $i = 1, m$

Para $j = 1, n$

$$g[i,j] = (f[i,j] - \min) / (\max - \min)$$

6.1.3 Algoritmos de filtrado de la imagen del iris.

La imagen se filtra utilizando dos algoritmos, el primero permite una mayor definición de contornos y se decidió emplearlo partiendo de la investigación realizada en el capítulo 5. El segundo algoritmo intensifica los tonos de los contornos y nos da como salida la matriz B :

- Algoritmo “Weiner” de paso bajo
- Algoritmo de intensificación de la imagen

El algoritmo “Weiner” procesa la intensidad de la imagen X empleando un *punto informado*⁵. La *vecindad*⁶ de tamaño $i \times j$ se emplea para estimar el promedio local de la imagen, es decir, la desviación estándar. Para el filtrado de las imágenes del iris, se empleó una *vecindad* = 3x3 (el término *vecindad* representa un vector con los valores 3 y 3), el vector *tamaño*(X) = [m n] donde m y n representa el número de renglones y columnas de X respectivamente, la función *ceros*() rellena una matriz de *tamaño*(X) con valores de 0. La notación [q : r] especifica un rango de valores que va de q hasta r . Tomando en cuenta estas definiciones el algoritmo se divide en los siguientes pasos:

1. Expansión de la matriz X

1.1 Obtiene la dimensión de la matriz de la imagen X y la almacena en *tamaño*(X)

1.2 $A_2 \leftarrow \text{ceros}(\text{tamaño}(X) + \text{vecindad} - 1)$

1.3 $A_2([\text{vecindad}[1]-1] / 2] + [1 : m], [(\text{vecindad}(2)-1)/2 + [1 : n]]) \leftarrow X$

2. $cx \leftarrow$ Conversión de los vecinos de la matriz X a columnas

3. Calcula la media de los vecinos de cx y los almacena en la matriz: *media_v*

3.1 $\text{media}_v \leftarrow \text{sum}(cx) / \text{prod}(\text{vecindad})$

⁵Consideramos un punto \mathbf{p} como el valor de la matriz $X[i,j]$

⁶La vecindad de un punto informado \mathbf{p} son los valores de la matriz $I[i,j]$ adyacentes a \mathbf{p} . El tamaño de la vecindad está dado por el punto informado \mathbf{p} más sus 8-vecinos. Es decir, si definimos una vecindad de 3X3 entonces tenemos 8-vecinos, v_1, \dots, v_8 y el punto informado \mathbf{p} .

v_4	v_3	v_2
v_5	\mathbf{p}	v_1
v_6	v_7	v_8

4. Calcula la varianza de los vecinos y la almacena en la matriz V de *tamaño*(X) el símbolo ($.*$) se emplea para denotar multiplicación de matrices

$$4.1 \ V \leftarrow \text{sum}(cx .* cx) / \text{prod}(\text{vecindad})$$

$$4.2 \ V \leftarrow V - (\text{media}_v .* \text{media}_v)$$

5. Calcula la frecuencia del ruido r

$$5.1 \ r \leftarrow \text{sum}(\text{sum}(V)) / \text{prod}(\text{tamaño}(V))$$

6. Filtra la imagen por pixeles y la almacena en la matriz B ! ($./$) se emplea para denotar división de matrices

$$6.1 \ B \leftarrow \text{media}_v + (\max(0, V - r) ./ \max(r, V)) .* (A_2 - \text{media}_v)$$

6.1.4 Algoritmo de intensificación de la imagen

Convierte la matriz B resultante del algoritmo de filtrado a una matriz In (intensificada). La matriz In contiene valores en el rango de 0 a 1.

1. $\text{Limites} \leftarrow [\min(\min(cx)) \ \max(\max((cx)))]$

2. $In \leftarrow \max(0, \min((x - \text{Limites}(1)) / \text{Limites}(2) - \text{Limites}(1)), 1))$

6.2 Ejemplo del procesamiento de las imágenes obtenidas

A continuación se presentan varias imágenes obtenidas como resultado de la aplicación de los algoritmos de: conversión de la imagen a escala de grises, normalización de la imagen y filtrado. Estas imágenes muestran el procesamiento general interno del sistema por pasos y no se pretende que funcionen como un manual de usuario. El programa que emplearía un usuario normal no presenta esta información, debido a que a los iridólogos no consideran apropiado, en general, descomponer o segmentar el iris de un paciente porque ven al iris de manera integral.

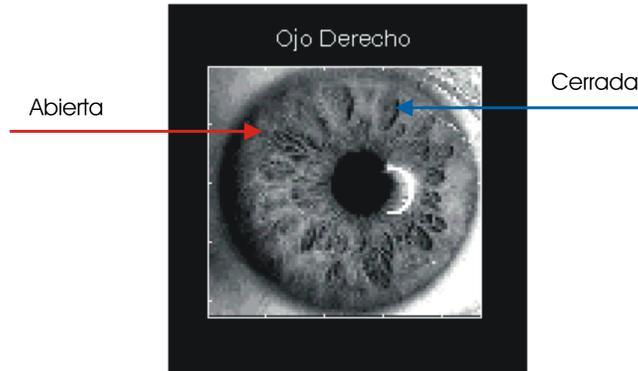


Figura 60: Imagen Original

La Figura 60 muestra un iris con lesiones cerradas (en su mayoría) y una lesión sub-aguda abierta, marcada con la flecha roja (izquierda), la flecha azul (derecha) señala una lesión cerrada. Debido a sus características esta imagen sirvió para obtener patrones de lesiones abiertas y cerradas con las que entrenamos la red neuronal. Diferenciar este tipo de lesiones es importante porque son muy parecidas y son las que más problemas presentan en el proceso de entrenamiento de la red y su posterior reconocimiento por el sistema.

6.3 Descomposición de la imagen

Como resultado del filtrado e intensificación de la imagen definidos en las secciones 6.1.1 a 6.1.3, obtenemos una imagen derivada de la original. La imagen derivada se encuentra en escala de grises, donde cada una de las lesiones está definida mediante cuatro valores:

b : blancos	n : negros
x_2 : valores cercanos a b	x_1 : valores cercanos a n

Figura 61: Posibles valores de la imagen

Los valores de b definen *medio contorno* de la lesión (del lado izquierdo o de lado derecho, pero siempre el mismo en una imagen dada) y no forman parte de ella, sino que definen su límite. Los valores de n se encuentran siempre adyacentes a los valores de b y forman parte de la lesión. Los valores de x_1 definen la lesión y los valores de x_2 definen el *medio contorno* opuesto a los valores de b . Como ejemplo tendríamos la lesión de la Figura 62.

		n	x_2			
	b	n	x_1	x_2		
	b	n	x_1	x_1	x_2	
	b	n	x_1	x_1	x_2	
	b	n	x_1	x_1	x_2	
		b	n	x_1	x_2	
			b	n	x_2	

Figura 62: Lesión característica en escala de grises

La imagen se convierte a valores de 0 y 1 empleando un valor al que llamamos *corte* y debe proporcionarse durante el filtrado. Si sabemos que los valores de los puntos que pertenecen a una lesión se encuentran en el rango de 0 a 0.5 (porque los puntos con valores entre 0.5 y 1 son contornos) entonces el valor del corte debe ser menor o igual a 0.5. Mientras más cercano a 0 sea el valor del corte podemos garantizar que el punto pertenece a la lesión y debido a que la imagen se filtró con un algoritmo para frecuencias bajas, entonces tomamos como valor del corte el punto medio que es de 0.25.

Una vez convertida la imagen a valores de 0 y 1 y con la finalidad de obtener una mejor claridad visual definimos las lesiones con valores de 1 y el resto de la imagen con valores de 0, con lo que, la lesión de la Figura 62 presentaría la forma de la Figura 63:

		b	n			
	n	b	b	n		
	n	b	b	b	n	
	n	b	b	b	n	
	n	b	b	b	n	
		n	b	b	n	
			n	n	n	

Figura 63: Lesión con valores de b y n

La Figura 64 muestra el resultado del procesamiento de la imagen previa a su descomposición. En la imagen podemos identificar las lesiones, la pupila y la sombra que presentaba la imagen original como valores blancos (unos), es posible notar que gracias a los algoritmos de tratamiento del iris, pudimos eliminar el reflejo de la luz de la fotografía (“lunita”) que pesentan una gran cantidad de imágenes del iris, por emplear iluminación para obtener la fotografía.



Figura 64: Imagen convertida a valores de 0 y 1

Las lesiones convertidas a valores de 0 y 1 sobre la imagen original se muestran en la Figura 65 a) y b). Podemos observar que en este punto los patrones de una lesión abierta y cerrada son distintos y esto nos permite hacer una discriminación adecuada.

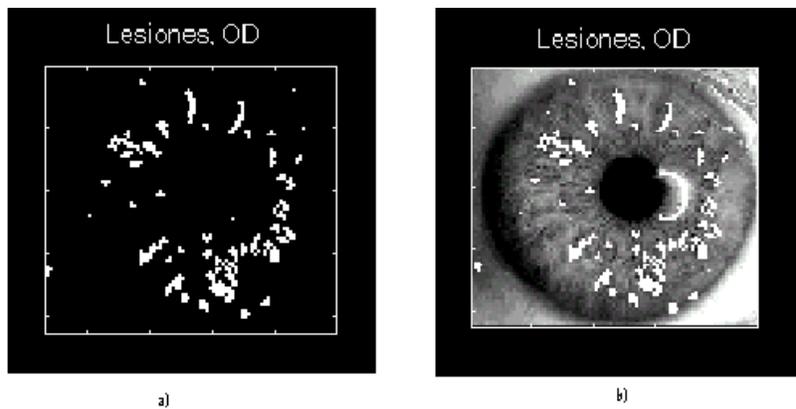


Figura 65: Lesiones sobre el iris

Conservar las lesiones en escala de grises resulta de vital importancia, esto nos permite especificar el *estado de una lesión* (tratado en el capítulo 2) que se presentará como parte del diagnóstico que emite el sistema. En este punto contamos con 3 imágenes: original, en escala de grises y con valores binarios (0 y 1).

6.4 Adelgazamiento de las lesiones

Como puede observarse en la Figura 65 a) aparecen una serie de puntos aislados que no se consideran como lesiones, además de este problema está el inconveniente de que en ciertas imágenes del valor de corte provoca que algunas lesiones independientes aparezcan como si se tratara de una sola. Por lo tanto empleamos un algoritmo de *adelgazamiento* o *esqueletización*.

En este punto, tenemos identificadas las lesiones como conjuntos de puntos blancos; por lo que, el primer paso del algoritmo consiste en identificar un punto \mathbf{p} con la característica mencionada y tomamos una vecindad de 3×3 , es decir tenemos 8-vecinos ($N8(p)$) que pueden ser de ambos colores (blancos y negros). Para adelgazar la imagen contamos el número de vecinos blancos a los que llamamos vB y el número de vecinos negros llamados vN . Si el número de $vB < 2$ y $vN \geq 4$ entonces eliminamos \mathbf{p} . Esto se deduce de la

observación de que si el número de vecinos blancos es menor que $2p$ es un punto aislado o es una esquina del contorno de una lesión. Sin embargo, es importante considerar la vecindad negra porque puede ser que la lesión que se está analizando sea angosta y el algoritmo de adelgazamiento la eliminaría.

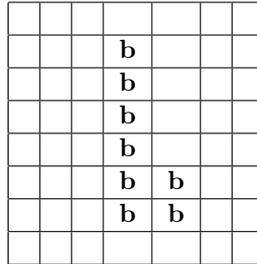


Figura 66: Ejemplo de una lesión angosta

6.5 Componentes conexas

El último paso de tratamiento de descomposición de la imagen consiste en generar una lista de lesiones que permitan entrenar la red neuronal o que representen los patrones a identificar. Empleamos el siguiente algoritmo para identificación de componentes conexas:

Recupera lesión (i,j) !encuentra el primer valor de 1 (blanco de la imagen)

1. define una lista L
2. apila el elemento (i,j)
3. marcar (i,j) como elemento visitado
4. Mientras la pila no esté vacía
 5. desapila (i,j)
 6. para cada pixel (i, j) blanco adyacente al pixel i,j

7. si (i, j) no está marcado entonces
8. marcar (i, j)
9. apilar (i, j)
10. inserta (i, j) en L
11. regresa L



Figura 67: Componentes conexos en la imagen del iris

La Figura 67 muestra las lesiones que se aislaron en escala de grises. Cada lesión es un componente almacenado en la lista L generada mediante el algoritmo de componentes conexos.

Una vez que aplicamos todos los algoritmos para el tratamiento de la imagen del iris, tenemos una lista de lesiones independientes con la que se entrena la red. La especificación de los valores iniciales y los patrones objetivo se presentan en la sección siguiente.

6.6 Valores iniciales del algoritmo de entrenamiento de la red neuronal

Para entrenar una red neuronal de retropropagación requerimos definir el conjunto de patrones objetivo ya que el entrenamiento de este tipo de red es supervisado, por lo tanto, para el entrenamiento además de definir los valores de los pesos y los umbrales proporcionamos para cada patrón a identificar el patrón objetivo asociado. Esto permite ajustar los valores de los pesos y los valores de los umbrales definidos en cada capa de la red. Los patrones objetivo son los siguientes:

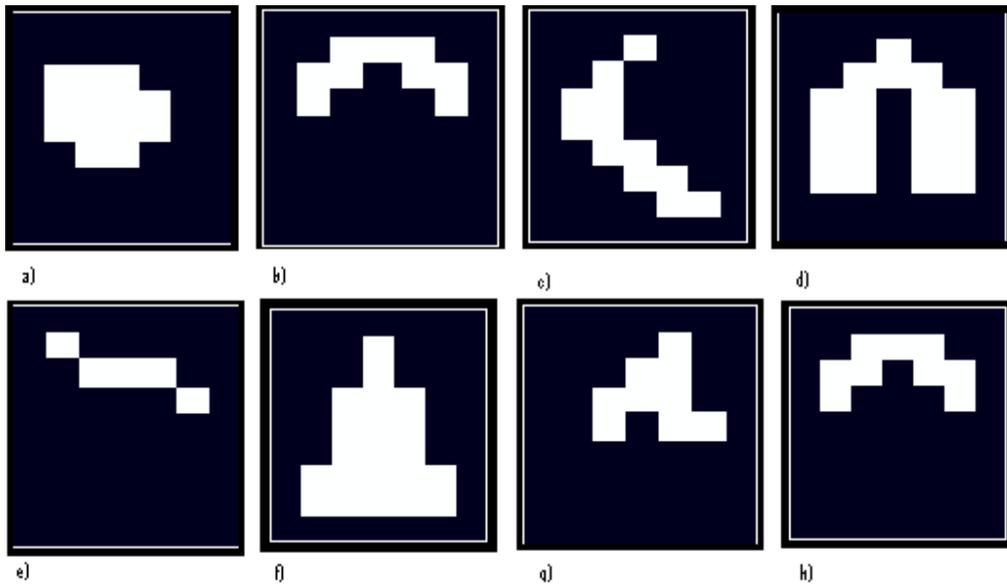


Figura 68: Conjunto de patrones objetivo

En la Figura 68 se muestra el conjunto de patrones objetivo en el inciso a) se muestra el rosario linfático, b) arco senil, c) lesión cerrada, d) lesión abierta, e) anillo nervioso, f) radio solaris, g) psora y h) anemia.

Una vez que se definen los patrones objetivo especificamos el número de capas que tendrá la red, para este caso, empleamos una sola capa oculta. El número de neuronas de la capa oculta fué de 40 y, la capa de salida consistió

de 8 neuronas, que corresponden a cada patrón objetivo.

Los valores de los pesos y los valores de umbral para cada capa se almacenan en un vector (son valores aleatorios) junto con la definición de la *función de salida* utilizada.

Para el entrenamiento de la red empleamos la función de *Tangente Hiperbólica Sigmoidal* por las características propias de esta función que puede consultarse en la tabla 3 del capítulo 5. Además de estos parámetros requerimos especificar la frecuencia del progreso de la red df , el número máximo de pasos del entrenamiento me , la meta para el error global eg y el factor de aprendizaje lr . Estos cuatro valores se almacenan en un vector tp , que se pasa como otro parámetro al algoritmo de entrenamiento. Los valores utilizados fueron:

$$\begin{aligned}df &= 1; \\me &= 8000 \\eg &= 0.005 \\lr &= 1\end{aligned}$$

El número máximo de pasos para el entrenamiento y la frecuencia del progreso de la red se ajustan regularmente a prueba y error derivándose del valor meta de error definido, es decir, si no alcanzamos este valor durante el proceso de aprendizaje debemos modificar df y me .

En los primeros entrenamientos consideramos valores inferiores de 8000 para me , pero no se alcanzaba la meta. El el valor 0.005 como meta para el error garantiza la identificación de las lesiones. Una vez definidos todos los valores iniciales se entrena la red con el método definido en la sección 5.4.

El código del algortimo de la red neuronal se presenta en el Anexo A, la descripción general del algortimo de aprendizaje de la red neuronal y el algoritmo de inferencia son los siguientes:

Algoritmo de aprendizaje de la red neuronal (descripción general)

Para n pasos del entrenamiento:

1. Inicializar los valores de los pesos y el valor de $i = 1$.
2. Presentar el patrón i -ésimo a la red en la capa de entrada
3. Obtener los niveles de activación de las unidades de salida, empleando el algoritmo de inferencia (descrito en el siguiente apartado). Si el rendimiento de la red se llega al valor predefinido (o al criterio para detenerse), entonces salir.
4. Actualizar los pesos mediante la regla de aprendizaje de la red.
5. Si $i = n$, entonces actualiza $i = 1$, en otro caso, incrementar i en 1. Ir al paso 2.

Algoritmo de inferencia

1. Presentar el patrón a la red en la capa de entrada
2. Calcular los niveles de activación de los nodos a través de la red
3. Si los niveles de activación de todas las unidades de salida se han calculado, entonces salir.

Los valores de los pesos y umbrales, así como el vector df , se almacenan en un archivo de texto y se cargan cada vez que se desea realizar el reconocimiento de una lesión.

En la Figura 69 mostramos el resultado del proceso de entrenamiento de la red neuronal de nuestro sistema.

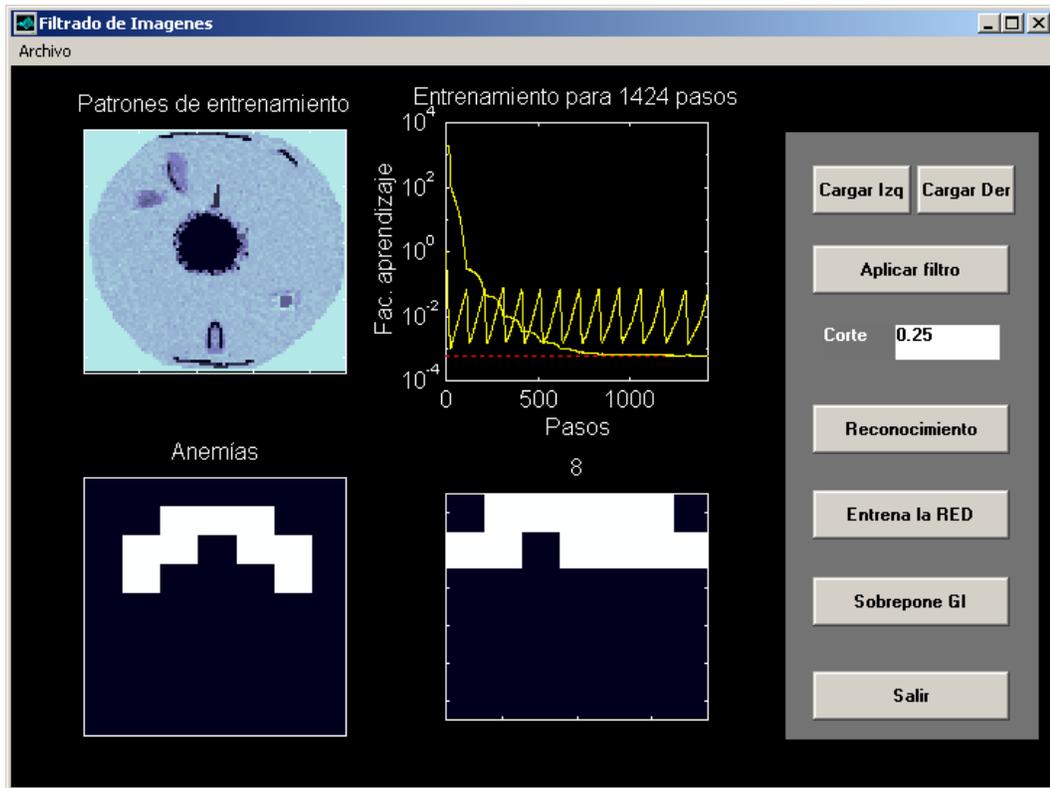


Figura 69: proceso de entrenamiento de la red neuronal

6.7 Ubicación de las lesiones en la gráfica iridológica

Para identificar el área en la que se encuentra una lesión dividimos el iris en doce secciones a partir del área de la pupila, las secciones corresponden en la gráfica iridológica a los ángulos 0° , 30° , 60° , 90° , etc. y cada una agrupa distintos órganos, en la página 2 del diagnóstico (presentado en la siguiente sección) se pueden consultar los órganos de cada área. En la tabla 5 presentamos esta relación.

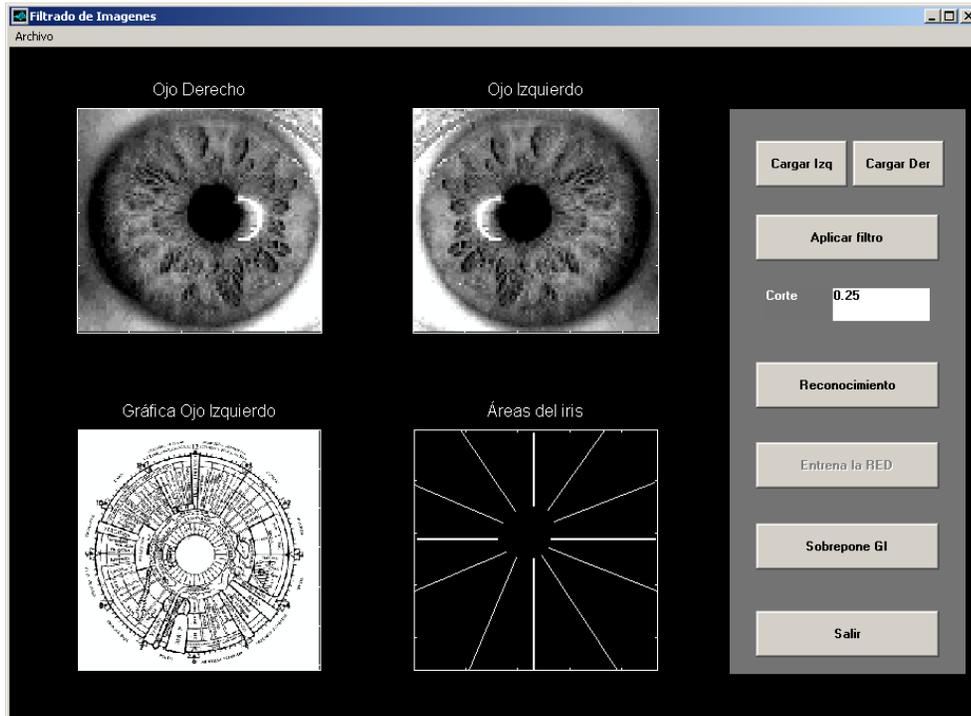


Figura 70: Secciones de áreas del iris

En la parte inferior de la Figura 70 se muestra la segmentación del iris. Consideramos que una lesión se encuentra en algún área si el 60% de ella cae dentro del área en cuestión. Generamos dos matrices ml y ma , ml contiene el número de lesión y sus coordenadas x, y de inicio y ma contiene todas las coordenadas contenidas en cada área de la gráfica iridológica partiendo de la identificación de la pupila del iris que se está analizando, a partir de la pupila consideramos 12 áreas que corresponden a los ángulos $0^\circ, 30^\circ, 60^\circ, 90^\circ, \dots, 360^\circ$. Para determinar el área en la que se encuentra cada lesión tomamos sus coordenadas de inicio y las buscamos en la matriz ma al encontrarlas detenemos el proceso y continuamos con la siguiente lesión. Este proceso se realiza por barrido, es costoso computacionalmente hablando, ya que implica generar una matriz nueva que contiene las coordenadas de las áreas para cada iris que se analiza, pero posee la ventaja de una mejor exactitud. Una vez localizada la lesión obtenemos el porcentaje que esta ocupando para marcarla como identificada y perteneciente a esa área. Esto permite que no se duplique la información de cada lesión.

Área	Órganos	Ángulo
1. Cerebro sensorial	Sistema de locomoción Mentalidad innata Centro del equilibrio Medula	90
2. Cuello	Mastoides Oído Cuello Corazón	60
3. Pulmón	Corazón Bronquiolos pulmón	30
4. Torax	Timo Pleura Torax Costillas Pl. Solar	0
5. Abdomen Superior	Brazo, mano Bazo Diafragma Abd. Superior Ovario, Testículo	330
6. Abdomen Inferior	Pelvis Peritoneo Pared abdominal Ingle	300
7. Pelvis	Muslo, rodilla, pie Riñón Escroto, perineo ano, recto Vagina, utero	270

Tabla 5: Relación Área/Órganos Ojo Izquierdo

Tabla 5, continuación...

8. Espalda baja	Vejiga Espalda baja Espalda media	240
9. Espalda alta	Espalda alta Omoplato Esofago	210
10. Garganta	Traquea C. Vocales Tiroides Faringe Laringe Amigdalas	180
11. Cara	Mandibula inferior Lengua, Boca Nariz Mandibula superior Ojo Frente, Sien	150
12. Cerebro Motriz	Desarrollo mental Centro de la palabra Ego, Presión arterial Area 5 sentidos	120

El algoritmo es el siguiente:

- 1.- Generar la matriz ml que contiene las coordenadas de inicio de cada lesión en la gráfica del iris
- 2.- Obtener el centro de la imagen e identificar la pupila

3.- Generar una matriz nueva ma que contenga 12 renglones y almacenar en cada columna las coordenadas de las áreas

4. Para cada renglon de ml

4.1.- Buscar las coordenadas de la lesión en ma

4.2 - Almacenar en la matriz mr la información de la lesión y el área que le corresponde.

Cuando concluimos el proceso anterior presentamos el diagnóstico.

6.8 Diagnóstico

Derivada de la ubicación de las lesiones identificadas en la gráfica iridológica obtenemos una lista de lesiones donde ya definimos el tipo de lesión, tenemos 8 tipos distintos que se muestran en la Figura 68. Además conocemos su ubicación en la gráfica del iris y especificamos 12 áreas: sensorial, cuello, pulmón, torax, abdomen superior, abdomen inferior, pelvis, espalda baja, espalda alta, garganta, cara, motriz. Por último el grado de la lesión se obtiene calculando el promedio de las tonalidades de grises que se encuentran presentes en la lesión de acuerdo a la siguiente tabla:

Estado	Coloración	Porcentaje
Agudo	Blanco	75% - 100%
Sub-agudo	Gris a gris oscuro	50% - 75%
Crónico	Negro	25% al 75%
Degenerativo	Negro	arriba del 75%

Tabla 6: Estados de las lesiones

Tomado en cuenta la Tabla 6 obtenemos como último paso un diagnóstico similar al mostrado en la Figura 71.

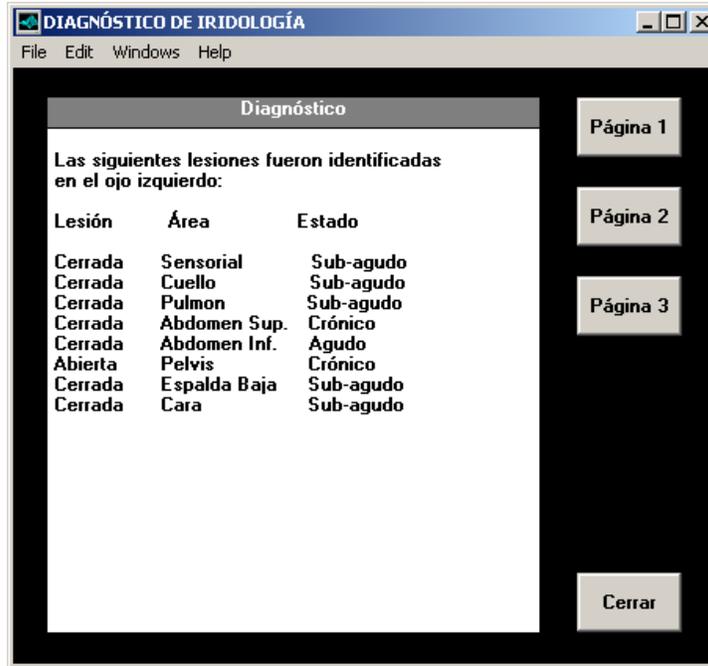


Figura 71: Diagnóstico final

El resultado presentado debe tener la característica de sencillez y limpieza que oculta la alta complejidad de la solución del problema. Tomamos en cuenta el perfil de usuario que desea utilizar el sistema y los intereses de los iridólogos por obtener información concreta en poco tiempo.

6.9 Utilización de la herramienta de diagnóstico

El software que utilizamos para implementar el sistema de diagnóstico es MATLAB 5.0 en su versión para Microsoft Windows pero existen versiones para Unix. Puede ejecutarse en cualquier computadora con un procesador Intel o compatible desde 386 en adelante. Requiere para su instalación 15MB de espacio libre en disco duro y 8 MB de memoria extendida.

Para realizar programas en MATLAB se emplea cualquier editor de texto o el block de notas. Los archivos se graban con la extensión .m. Programamos el sistema de diagnóstico con el lenguaje que proporciona MATLAB y presenta

la ventaja de que utiliza poco espacio en disco y requiere poca memoria, lo que compensa la carga generada por las imágenes.

Para entrenar la red neuronal seleccionamos un conjunto de imágenes que almacenamos en un solo archivo con formato BMP. El archivo se llama “PatEn0.bmp” y se encuentra en el mismo directorio donde guardamos los archivos del sistema.

La herramienta MATLAB no genera archivos ejecutables por lo que para emplear el sistema se requiere instalar el compilador que tiene un costo muy bajo en el mercado, además existen versiones especiales para estudiante con un costo aún menor. Se debe copiar la carpeta que contiene todos los archivos denominada “Entrena” y ejecutar el programa principal con el nombre “filtro”, en este programa empleamos varias funciones que diseñamos y es el encargado de invocarlas. MATLAB no proporciona un entorno de desarrollo directo por lo que, la interfaz del sistema está diseñada con funciones matemáticas.

Una vez que ejecutamos nuestro sistema de diagnóstico, podemos cargar las imágenes del iris que se quieren analizar presionando un botón que proporciona la interfaz , tanto para el ojo izquierdo como para el ojo derecho. Se deben proporcionar las imágenes en formato BMP y pueden obtenerse directamente con una cámara digital o tomando las fotos y digitalizandolas en cualquier dispositivo para este propósito.

El diagnóstico generado y las imágenes del iris se pueden imprimir desde la interfaz hacia cualquier dispositivo de impresión.

7 Capítulo 7. Conclusiones

Una vez que realizamos la implantación del software, tomamos una muestra de 50 imágenes del iris, aplicamos filtrado, adelgazamiento y separamos todas las lesiones. Posteriormente almacenamos en un archivo de texto la información de las matrices que contenían las lesiones para entrenar la red. Con la red entrenada es posible almacenar los valores de los pesos y de los umbrales para realizar en cualquier momento una identificación posterior de lesiones.

La idea original fue manipular el conjunto de imágenes para entrenar una sola red neuronal con los patrones obtenidos y posteriormente, reconocer todos los tipos de lesiones que teníamos clasificados. Sin embargo, con las primeras pruebas pudimos concluir que existen ciertas lesiones, las que rodean al iris de manera circular, como la congestión linfática o circulatoria, el anillo escamoso, el anillo de absorción y el colesterol o sodio, que tienen el mismo patrón y su tamaño crece considerablemente debido a que rodean todo el iris, por lo que no fue posible almacenarlas en una matriz de 6×6 . La solución para este tipo de lesiones es almacenarlas en otra matriz y entrenarlas con otra red que tiene las mismas características que la red que utilizamos para entrenar el 2 grupo de lesiones. Este grupo lo presentamos en el capítulo 6 sección 6.6 Figura 6.8. Por lo que el proceso de entrenamiento se realizó en 2 fases, básicamente realizamos clases de patrones y entrenamos una misma red con distintos parámetros.

Sin embargo, podemos concluir en este punto, que las redes neuronales actuales son lo suficientemente robustas para identificar y clasificar conjuntos grandes de patrones.

La investigación más fuerte de este trabajo se realizó en el área de filtrado de imágenes. La gran cantidad de algoritmos existentes hace que sea más difícil la selección de alguno de ellos, pero presenta la ventaja de que solucionan una amplia gama de problemas y se pueden mezclar para dar resultados idóneos que se apeguen a los requerimientos de filtrado de distintas imágenes. La primera utilización de estos algoritmos se realizó a prueba y error sin obtener los resultados que necesitábamos, por lo que la investigación de los tipos y familias así como su funcionamiento y uso facilitó la obtención de los algo-

ritmos de filtrado requeridos para las imágenes del iris. Esta investigación desencadenó otra concerniente a la representación de imágenes digitales y los tipos de operaciones que se les pueden realizar.

Para utilizar el sistema dividimos el proceso en dos partes empleando una contraseña de usuario para activar diferentes botones. El primer proceso consiste en el entrenamiento de la red y para utilizarlo se requiere de los patrones originales (imágenes de lesiones) y de los patrones objetivo. El segundo proceso consiste en identificar las lesiones del iris en una imagen de algún paciente para posteriormente ubicar cada lesión identificada en la gráfica iridológica. El resultado que se muestra en pantalla debe ser simple y claro tomándose en cuenta el perfil del usuario.

La obtención de las imágenes para entrenar la red, fue otro de los mayores obstáculos, debimos obtener un conjunto de imágenes del iris (por lo menos 2 diferentes) de cada tipo de lesión que además permitiera “ver” claramente la lesión que se quería identificar. El programa en el que implementó el sistema es Matlab, muchas de las funciones que incluye este programa para “cargar” imágenes y manipularlas, fallan si la resolución de los archivos (en formato .bmp) es muy grande, por lo que tuvimos que buscar imágenes que no se degradaran demasiado al utilizar una resolución baja. Si se emplea una cámara digital (especial para tomar imágenes de iris) la resolución puede ajustarse sin ningún problema.

El proceso para resolver el problema fue complejo. Cada etapa requirió de una investigación, diseño de algoritmos y posterior uso.

Los resultados obtenidos son correctos ya que se cumplió el objetivo, manipular imágenes del iris y entrenar una red que permitió su identificación y posterior ubicación en la gráfica iridológica.

References

- [1] Lau Califford, *Neural Networks, Theoretical Foundations and Analysis*, A selected print volume, IEEE Press, 1991.
- [2] Jensen Bernard, *Science and Practice in the healing arts*, Vol. II, Bernard Jensen publisher, 1982.
- [3] Hopcroft Joe, “Introduction to Computerized Iris Analysis”, *magazine of scientific iris*, 1998.
- [4] Lippmann Richard P., An Introduction To Computing with Neural Nets, *IEEE ASSP magazine*, April 1991, pp. 4-22.
- [5] Kohonen Teuvo, The Self-Organizing Map, *IEEE*, vol. 78, no. 9, sept. 1990, pp. 1464-1480.
- [6] Hopfield, J., Neuronal networks and physical systems with emergent collective properties. *Proceedings of the National Academy of Sciences of the USA.*, 79:2554-2588.
- [7] Hagan Martin T., Mohammad B. Menhaj, Training Feedforward Networks with Marquardt algorithm, *IEEE Transation on neural networks*, vol 5 No. 6 Nov. 1994.
- [8] *Diccionario enciclopédico*, Ediciones Grijalbo, S.A. 1995.
- [9] Jensen Bernard, *Science and Practice in the healing arts*, Vol. I, Bernard Jensen publisher, 1982.
- [10] Jensen Bernard, *Términos en iridiología*, Bernard Jensen publisher, 1993.
- [11] De la Escalera Arturo, *Visión por computador, Fundamentos y métodos*, Prentice Hall, España, 2001.

- [12] P. Maragos, "Pattern spectrum and multiscale representation". *IEEE Trans. Patter Anal. Macchine Intell.*, vol. 11, no. 7, pp. 701-716, July 1989.
- [13] I. Pitas and A. N. Venetsanopoulos, "Morphological shape decomposition", *IEEE Trans. Patter Anal. Macchine Intell.*, vol. 12, no. 1, pp. 38-45, 1990.
- [14] Coltuc Dinu and I. Pitas, "Morphological residual representation of signal and images" *IEEE Tansc. of image processing. Macchine Intell.*, vol. 5, no. 11, pp. 1569-1572, nov. 1996.
- [15] Serra. J. "*Image Analysis and Mathematical Morphology*", Academic Press, London, 1992.
- [16] Ortíz Campos José Francisco, *Matemáticas 2*, Publicaciones culturales, México, 1999.
- [17] G.P. Dinnen, "Programming pattern recognition", in *Proc. West. Joint Comput. Conf. (New York)*, 1995, pp. 94-100.
- [18] R.A. Kirsch, L. Cahn, C. Ray, and G.J. Urban "Experiments in processing pictorial information with a digital computer", in *Proc. East. Joint Comput. Conf. (New York)*, 1957, pp. 221-229.
- [19] T.M Alcorn and C.W. Hoggar, "Pre-processing of data for character recognition", *Marconi Rev.*, vol. 32, pp. 61-81, 1969
- [20] D. Rutovitz, "Pattern Recognition", *J. Roy. Stat. Soc.*, vol 129, Series A, pp. 504-530, 1966
- [21] C. J. Hilditch, "Linear skeletons from square cupboards", in *Machine Intell. (B. Meltzer and D. Michie, Eds.)* New York: Amer. Elsevier, 1968, pp. 325-347, vol. 3
- [22] Oppenheim, A.V., R.W. Schafer, and T.G. Stockham, Jr., Non-Linear Filtering of Multiplied and Convolved Signals. *Proc. IEEE*, 1968. 56(8): p. 1264-1291
- [23] A. K. Jain, *Fundamentos del procesamiento de imagenes digitales*, Prentice Hall, 1989

- [24] S.K. Mitra, S. Thurhofer, M. Lightstone, and Norbert Strobel, "Two-dimensional Teager operators and their image processing applications", *Proc. 1995 IEEE Workshop on Nonlinear Signal and Image Processing*, pp. 959-962, June 1995
- [25] S. Mo and V.J. Mathews, "Adaptative binarización of document images", *Proc. 1995 IEEE Workshop on Nonlinear Signal and Image Processing*, pp. 967-970, June 1995
- [26] M. Nowak, Eves Rib, *St. Martins Press*, New York, 1980.
- [27] L.A. Olzak and J. P. Thomas, "Seeing spatial patters", in *Handbook of perception and Human Performance* (K.R. Boff, L. Kaufman, and J.P. Thomas, eds.), ch. 7, New York: John Wiley and Sons, 1986. pp. 963-966, June, 1995.
- [28] G. Ramponi, "A simple cubic operator for sharpening an image", *Proc. 1995 IEEE Workshop on Nonlinear Signal and Image Processing*, pp. 963-966, June 1995
- [29] S. Thurnhofer, "*Quadratic Volterra Filters for Edge Enhacement and Their Applications in Image Processing*", Ph.D. dissertation, University of California, Santa Barbara, 1994.
- [30] A.C. Botik, T.S. Huag and D.C. Munson, Jr. "A generalization of order statics", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp 1342-1349, Dec. 1983.
- [31] Y. H. Lee and S.A. Kassam, "Generalized median filterin and related non-linear filtering techniques", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp 672-683, June. 1985.

- [32] R.C. Hardie and K.E. Barner, "Rank Conditioner Rank Selection Filters for Signal Restoration", *IEEE Trans. Image Processing*, vol. 3 no. 2, pp. 192-206, Mar. 1994.
- [33] Y. H. Lee and S.Y. Park, "A frequency selective nonlinear filter based on a generalized median filter", in *Proc. Conf. Inform. Sci. Syst.* (Baltimore, MD), pp 384-389, Mar. 1985.
- [34] S.Y. Park and Y. H. Lee, "Double Smoothing of Images Using Median and Weiner Filters", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-37, no. 6, pp 943-946, June 1989.
- [35] P.P. Gandhi and S.A. Kassam, "Design and performace of combination filters for signal restoration", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-39, pp 1524-1540, July 1991.
- [36] A. Papoulis, *Probability, Random variables and stochastic Processes*, 2nd Ed., McGraw-Hill, 1984.
- [37] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.
- [38] P. M. Grant and J. P. Sage, "A Comparison of Neural Network and Matched Filter Processing for Detecting Lines in Images", in *J. S. Denker (Ed.) AIP Conference Proceedings 151, Neural Networks for Computing*, Snowbird Utah, AIP, 1986.
- [39] R.P. Lippmann, B. Gold, and M.L. Malpass, "A Comparison of Hamming and Hopfield Neural Nets for Pattern Classification", *MIT Lincoln Laboratory Technical Report*, TR-769.
- [40] Hagan, Martin T., Demuth Howard B., Beale Mark., *Neural Network Design*. PWS Publishing Company., 1998.
- [41] Hopfield, J., Neuronal networks and physical systems with emergent collective properties. *Proceedings of the National Academy of Sciences of the USA*, 79:2554-2588.
- [42] Hopfield, J. (1984). "Neurons with graded response have collective computational properties like those of two-state neurons". *Proceedings of the National Academy of Sciences of the USA*, 81:3088-3092.
- [43] Hopfield, J and Tank, D. (1985). "Neural computation of decisions in optimization problems". *Biological Cybernetics*, 52:141-152.
- [44] R. Rosenblatt, *Principles of Neurodynamics*, New York, Spartan Books (1959).

- [45] Lippmann Richard P., An Introduction To Computing with Neural Nets, IEEE ASSP magazine, April 1991, pp. 4-22.
- [46] Widrow, Bernard and Lehr, Michael A. 30 Years of Adaptative Neural Networks: Perceptron, Madaline, and Backpropagation, *IEEE, Vol. 78, no. 9*, sept. 1990, pp. 1415-1442.
- [47] Rumelhart, D. E., Hinton, G. E., Williams, R. J., "Learning Representations by Back-Propagation Errors", *Nature, vol. 323*, pp. 533-536, 1986.
- [48] R. Battiti, "First -and second order methods for learning: Between steepest descent and Newton's method," *Neural computation*, vol. 4, no. 2, pp 141-166, 1992.
- [49] Marquardt, D., "An algorithm for least squares estimation of non-linear parameters", *J. Soc. Ind. Appl. Math.*, pp. 431-441, 1963.
- [50] Van Vliet, L.J. and B.J.H. Verwer, "A Contour Processing Method for Fast Binary Neighbourhood Operations", *Pattern Recognition Letters*, 1988. 7(1): p. 27-36.
- [51] Canny, J., "A Computational Approach to Edge Detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986. PAMI-8(6): p. 679-698.
- [52] Marr, D. and E.C. Hildreth, "*Theory of edge detection*". Proc. R. Soc. London Ser. B., 1980. 207: p. 187-217.