

**RECONOCIMIENTO DE CARACTERES ÓPTICOS OCR POR MEDIO DE  
CORRELACIÓN Y REDES NEURONALES**



**AUTOR:  
MAURICIO DURÀN GÒMEZ  
ID: 73731**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA  
ESCUELA DE INGENIERÍA Y ADMINISTRACIÓN  
FACULTAD DE INGENIERÍA ELECTRÓNICA  
FLORIDABLANCA  
2009**

**RECONOCIMIENTO DE CARACTERES ÓPTICOS OCR POR MEDIO DE  
CORRELACIÓN Y REDES NEURONALES**



**AUTOR:  
MAURICIO DURÁN GÓMEZ  
ID: 73731**

**Trabajo de Grado para Optar al Título de Ingeniero Electrónico**

**DIRECTOR DEL PROYECTO  
ING. JUAN CARLOS VILLAMIZAR**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA  
ESCUELA DE INGENIERÍA Y ADMINISTRACIÓN  
FACULTAD DE INGENIERÍA ELECTRÓNICA  
FLORIDABLANCA  
2009**

Nota de Aceptación:

---

---

---

---

Firma del Director del Proyecto

---

Firma del Docente Calificador

---

Firma del Docente Calificador

Floridablanca, 8 de Septiembre de 2009

## **AGRADECIMIENTOS**

Este proyecto lo dedico en especial a Dios por permitirme culminar esta etapa de mi vida, por ofrecerme la virtud de la paciencia, del conocimiento y de la perseverancia para poder superar los obstáculos presentados para culminar mi meta.

De igual forma, doy gracias a mis padres y a mi hermano por darme el apoyo incondicional y por motivarme a alcanzar este logro.

A mi director de proyecto, Ingeniero Juan Carlos Villamizar por apoyarme e interesarse de manera significativa en el proceso para la construcción del proyecto de grado y por aportar todos sus conocimientos y experiencia necesaria para realizar un trabajo de calidad.

## CONTENIDO

	<b>Pág.</b>
INTRODUCCIÓN.....	12
DELIMITACIÓN.....	13
ANTECEDENTES.....	14
OBJETIVOS.....	15
1. CONCEPTOS BÁSICOS DE LAS IMÁGENES.....	16
1.1 CLASES DE IMÁGENES.....	16
1.2 TIPOS DE IMÁGENES.....	16
1.2.1 Imágenes RGB.....	17
1.2.2 Imágenes de intensidad.....	18
1.2.3 Imágenes binarias.....	18
1.2.4 Imágenes indexadas.....	18
1.3 LECTURA Y ESCRITURA DE IMÁGENES.....	19
1.4 FUNCIONES PARA LA CONVERSIÓN DE IMÁGENES.....	21
1.4.1 Herramientas para la conversión entre tipos de datos.....	21
1.5 ACCESO A PÍXELES Y PLANOS EN IMÁGENES.....	23
1.5.1 Colores de una imagen.....	24
1.5.2 Representación de colores.....	25
1.6 FILTRAJE ESPACIAL.....	26
1.6.1 Función medfilt2.....	28
1.7 SUBMUESTREO DE IMÁGENES.....	28
1.8 CONECTIVIDAD DE COMPONENTES (PÍXELES) EN LA IMAGEN.....	28
2. RECONOCIMIENTO ÓPTICO DE CARACTERES.....	30
2.1 RECONOCIMIENTO DE CARACTERES MANUSCRITOS.....	30
2.2 IMPLEMENTACIÓN DE UN SISTEMA OCR.....	31
2.2.1 Medición de características.....	31
2.2.2 Funciones para la extracción de bordes.....	32
2.2.3 Taxonomía de un OCR.....	33
2.2.4 Formas de adquisición de datos.....	33
2.2.5 Dependencia e independencia del escritor.....	34
2.2.6 Tamaño del léxico.....	35
2.2.7 Aislamiento de los caracteres.....	36
2.2.8 Extracción de características de imágenes.....	37
2.3 PROCESAMIENTO DE IMÁGENES.....	38
2.3.1 Binarización.....	38
2.3.2 Cierre de contornos.....	39
2.3.3 Operadores morfológicos.....	39

3. CONSTRUCCIÓN DE UN SISTEMA OCR .....	45
3.1 SEGMENTACIÓN.....	46
3.1.1 Estrategias de segmentación .....	47
3.2 SIMPLIFICACIÓN.....	47
3.3 EXTRACCIÓN DE CARACTERÍSTICAS .....	48
3.3.1 Extracción de caracteres y métodos de clasificación .....	48
3.4 DECISIÓN.....	49
4. MÉTODOS DE CLASIFICACIÓN.....	50
4.1 CORRELACIÓN.....	50
4.1.1 Correlación canónica.....	52
4.1.2 Correlación entre variables.....	52
4.1.3 Correlación en Matlab .....	54
4.1.4 Regresión lineal.....	54
4.2 DESARROLLO .....	55
4.3 REDES NEURONALES .....	59
4.3.1 Características de una red neuronal artificial .....	60
4.3.2 Funcionamiento de una RNA.....	61
4.3.3 Diseño y programación de una RNA .....	62
4.3.4 Estructura.....	62
4.3.5 Ventajas .....	63
4.3.6 Tipología de las RNA.....	63
4.3.7 Topología .....	64
4.3.8 Tipos de entrada de una RNA .....	67
4.3.9 Funciones de transferencia .....	67
4.3.10 Aprendizaje .....	70
4.4 DESARROLLO .....	70
4.4.1 Implementación de un OCR con redes neuronales con funciones de base radial .....	72
4.4.2 Como funcional una RBF-NN .....	74
4.4.3 Arquitectura.....	74
CONCLUSIONES .....	81
RECOMENDACIONES.....	83
BIBLIOGRAFÍA.....	84

## LISTA DE FIGURAS

	<b>Pág.</b>
<b>Figura 1.</b> Representación de imágenes RGB: imagen RGB y representación matricial.....	17
<b>Figura 2.</b> Representación de una imagen a escala de grises en Matlab.....	18
<b>Figura 3.</b> Representación de imágenes indexadas: imagen indexada original, parcial de la matriz imagen y matriz colormap .....	19
<b>Figura 4.</b> Representación binaria GRB.....	25
<b>Figura 5.</b> Representación binaria de una letra.....	26
<b>Figura 6.</b> Filtro de la mediana.....	27
<b>Figura 7.</b> Conectividad conexión de 8 y 4 .....	29
<b>Figura 8.</b> Bordeado de una letra.....	32
<b>Figura 9.</b> Esquema ilustrativo de algunos dispositivos de adquisición “online” Y “off-line”...	34
<b>Figura 10.</b> Texto de un solo escritor .....	35
<b>Figura 11.</b> Texto de varios escritores .....	35
<b>Figura 12.</b> Ejemplos de formas básicas de elementos estructurantes básicos .....	40
<b>Figura 13.</b> Elemento estructurante plano de tamaño 3x3 .....	40
<b>Figura 14.</b> Matriz original y submatrices.....	41
<b>Figura 15.</b> Dilatación de una imagen binaria mediante un elemento estructurante de tamaño 3x3.....	42
<b>Figura 16.</b> Dilatación de una señal unidimensional definida por la función f. Elemento estructurante de tamaño 3.....	42
<b>Figura 17.</b> Erosión de una imagen binaria mediante un elemento estructurante de tamaño 3x3 .....	43
<b>Figura 18.</b> Erosión de una señal unidimensional definida por la función f. Elemento estructurante de tamaño 3.....	43
<b>Figura 19.</b> Dilatación y erosión de una señal unidimensional definida por la función f. elemento estructurante de tamaño 3.....	44
<b>Figura 20.</b> Diagrama del reconocedor de caracteres ópticos (algoritmo del OCR) .....	45
<b>Figura 21.</b> Proceso global de la segmentación de imágenes .....	47
<b>Figura 22.</b> Diagrama de dispersión de los coeficientes de correlación .....	53
<b>Figura 23.</b> Coeficiente de correlación entre dos variables .....	54
<b>Figura 24.</b> Nubes de puntos y recta de mínimos cuadrados.....	55
<b>Figura 25.</b> Redimensionamiento de la imagen y separación de líneas .....	57
<b>Figura 26.</b> Red neuronal artificial .....	60
<b>Figura 27.</b> Comparación entre una red neuronal biológica y una red neuronal artificial.....	61
<b>Figura 28.</b> RNA (Perceptrón).....	65
<b>Figura 29.</b> Red Adaline .....	65
<b>Figura 30.</b> Red Backpropagation.....	66
<b>Figura 31.</b> Función escalón .....	68
<b>Figura 32.</b> Función lineal.....	68
<b>Figura 33.</b> Funciones sigmoidales.....	69
<b>Figura 34.</b> Etapas del proceso de desarrollo con el método de RNA.....	71
<b>Figura 35.</b> Imagen original de tamaño 500x379 píxeles imagen redimensionada a un tamaño de 70x50 píxeles.....	71

<b>Figura 36.</b> Reducción de la imagen a un tamaño de 7x5.....	72
<b>Figura 37.</b> a) El Perceptrón representa límites de decisión (lineales). b) Las RBF-NN representan clases (a través de las funciones base) .....	73
<b>Figura 38.</b> Arquitectura de una RBF-NN .....	73
<b>Figura 39.</b> Pantalla principal .....	76
<b>Figura 40.</b> Reconocimiento de caracteres a partir de una imagen (a) .....	76
<b>Figura 41.</b> Reconocimiento de caracteres a partir de una imagen (b) .....	77
<b>Figura 42.</b> Reconocimiento de texto manuscrito.....	77
<b>Figura 43.</b> Entrenamiento de redes neuronales .....	78
<b>Figura 44.</b> Parámetros para redes Backpropagation .....	79
<b>Figura 45.</b> Parámetros para red de base radial .....	79
<b>Figura 46.</b> Ejemplos de entrenamiento .....	80

## LISTA DE TABLAS

	<b>Pág.</b>
<b>Tabla 1.</b> Formatos y extensiones soportadas por Matlab.....	20
<b>Tabla 2.</b> Comandos para conversión de datos .....	22
<b>Tabla 3.</b> Mediciones realizadas por la función imfeature .....	31
<b>Tabla 4.</b> Funciones de transferencia.....	69

## RESUMEN GENERAL DE TRABAJO DE GRADO

**TITULO:** RECONOCIMIENTO DE CARACTERES ÓPTICOS OCR POR MEDIO DE CORRELACIÓN Y REDES NEURONALES.

**AUTOR:** MAURICIO DURÁN GÓMEZ

**FACULTAD:** FACULTAD DE INGENIERÍA ELECTRÓNICA

**DIRECTOR(A):** JUAN CARLOS VILLAMIZAR

### RESUMEN

Esta tesis tiene como objetivo general el estudio y la aplicación de un sistema que realice el reconocimiento de caracteres ópticos en imágenes previamente tomadas o capturadas. Se empieza el reconocimiento a partir de la digitalización del texto que puede ser a través de una cámara, un escáner o capturando la imagen del carácter escrito por medio de un lápiz óptico. Como primer paso, se realiza un tratamiento de la imagen capturada, pasando desde la binarización de la imagen hasta el filtrado que elimina el ruido que la distorsiona.

El sistema de reconocimiento desarrollado es dependiente del escritor, por lo que cada sistema de clasificación escogido para reconocer el carácter debe ser previamente adaptado para que pueda realizar el proceso de reconocimiento de la manera correcta. Sin embargo, se tiene en estos métodos de clasificación una gran variedad de ejemplos de entrenamiento, muestras y plantillas que permitirán en un alto índice, el reconocimiento de caracteres manuscritos de terceros en el programa.

Para el proyecto se utilizaron dos métodos de clasificación muy conocidos para relacionar imágenes y crear variables de similitud para la correcta etiquetación de los caracteres a reconocer. Estos dos métodos son los de correlación y la de redes neuronales. El método de correlación da como resultado un valor de relación entre los datos o patrones de las imágenes y dicha relación puede calcularse mediante diversos coeficientes que miden el grado de correlación entre las mismas. Por otro lado, las redes neuronales serán entrenadas de tal forma que el sistema pueda identificar de un conjunto de posibles soluciones la mejor de las salidas en relación al dato de entrada del sistema.

Una vez se hayan extraído y clasificados los patrones del carácter de entrada, queda por último la etiquetación del carácter de salida correctamente reconocido.

**PALABRAS CLAVES:** Reconocimiento de caracteres ópticos, correlación, redes neuronales.

## THESIS OVERVIEW

**TITLE:** OPTICAL CHARACTER RECOGNITION OCR THROUGH CORRELATION AND NEURAL NETWORKS

**AUTHOR:** MAURICIO DURÁN GÓMEZ

**FACULTY:** FACULTY OF ELECTRONIC ENGINEERING

**DIRECTOR:** JUAN CARLOS VILLAMIZAR

## ABSTRACT

This thesis aims to study the general and the implementation of a system to perform optical character recognition on images previously taken or caught. It starts from the recognition of digitized text that can be through a camera, scanner, or capturing the image of the character written by a stylus. As a first step, is a treatment of the captured image, moving from the binarization of the image to the filter that removes noise distortion.

The recognition system developed is dependent on the writer, so each classification system chosen to recognize the character must first be adjusted so that you can make the process of recognizing the right way. However it is these methods of classification in a variety of training examples, samples and templates that allow a high rate, the recognition of handwritten characters in the third.

The project will use two well-known classification methods to link images and create similar variables for the correct labeling of the characters to recognize. These two methods are the correlation and the neural network. The correlation method results in a value of relationships between data or patterns of images and this ratio can be calculated using various ratios that measure the degree of correlation between them. On the other hand neural networks are trained so that the system can identify a set of possible solutions the best of trips in relation to the input of the system.

Once you have collected and classified the patterns of character input, is by labeling the last character output correctly recognized.

**KEYWORDS:** Optical character recognition, correlation, neural networks.

## INTRODUCCIÓN

El reconocimiento de caracteres ópticos (Optical Character Recognition OCR) es un mecanismo que convierte caracteres alfanuméricos impresos o manuscritos en archivos o comandos ASCII legibles por máquina usando las propiedades ópticas del equipo y sin necesidad de digitar la información.

El reconocimiento de caracteres ópticos se ha convertido en un reto de la tecnología contemporánea debido a las inmensas cantidades de aplicaciones que abarca el tema y a lo sofisticada que es su técnica. Su popularidad radica en su versatilidad y uso, que permite convertir material impreso, mecanografiado o escrito manualmente por medios computacionales sin necesidad de digitar la información. Por ejemplo, cuando se dispone de algún tipo de información en forma de documento impreso o manual y se desea procesarla mediante un computador, existen dos opciones: una primera consistiría en introducirla a través del teclado, labor larga y tediosa.

Otra posibilidad es automatizar esta operación por medio de un sistema de OCR compuesto de un software y hardware adecuado que reduciría considerablemente el tiempo de entrada de datos. En este caso, el hardware utilizado se reduce a un dispositivo de captura de imágenes (cámara, escáner o lápiz óptico) y a un computador personal en el que se ejecuta el software de reconocimiento desarrollado para dicho propósito.

El objetivo general de este proyecto es el de realizar el reconocimiento de caracteres ópticos por medio del método de correlación y el de redes neuronales, pero tras este objetivo se persigue la idea de permitir al hombre por medio de la tecnología OCR hacer sus labores y tareas de una manera más fácil, práctica y productiva, que es adonde el desarrollo de recursos e investigaciones en este tipo de campo apuntan.

## DELIMITACIÓN

Para este proyecto se han considerado ciertas restricciones que delimitan el estudio y desarrollo de la investigación. Estas restricciones son consideradas por separados en las técnicas de clasificación por correlación y por redes neuronales para el reconocimiento de caracteres ópticos.

Las siguientes restricciones fueron consideradas en el método por correlación para facilitar la operación de reconocimiento:

- Los caracteres a reconocer se pueden ubicar dentro del contexto de la imagen a través de renglones de caracteres o por medio de un carácter por imagen.
- Los renglones que forman el texto no podrán estar solapados ni inclinados. Deberán estar al menos separados por una línea en blanco.
- Los caracteres no pueden estar solapados o unidos.
- Las letras del texto pueden ser de cualquier tipo y tamaño.
- El sistema es dependiente del escritor.

En el caso del reconocimiento por redes neuronales se considera absolutamente necesario el adecuado entrenamiento del tipo de red escogida. De igual forma se tienen algunas limitaciones en el sistema por RNA:

- Los caracteres a reconocer pueden estar ubicados aleatoriamente en el contexto de la imagen pero no podrán estar solapados ni inclinados, deberán estar al menos separados por una línea en blanco.
- Los caracteres no pueden estar solapados o unidos.
- Las letras del texto pueden ser de cualquier tipo y tamaño.
- El sistema es dependiente del escritor.

## ANTECEDENTES

- En los antecedentes de este proyecto se puede resaltar a nivel nacional el desarrollo de la profesora Lucía Chacón Alvarado en el reconocimiento de caracteres ópticos aplicados en bibliotecas para convertir textos escritos en archivos digitales reduciendo recursos y tiempo para realizar la labor de conversión.

Reconocimiento de Caracteres Ópticos Aplicaciones en Bibliotecas

*M.I.L.S. Lucía Chacón Alvarado*

Profesora

Escuela de Bibliotecología

Documentación e Información

Universidad Nacional

- A nivel regional se realizó una ponencia de los estudiantes Roberto Carlos De Armas Domínguez y Lener Samuel Bautista Rodríguez de la Universidad Industrial de Santander UIS, cuyo título es “Reconocimiento automático de caracteres manuscritos y marcas, orientado al procesado de encuestas”<sup>1</sup> y allí se especifica que el proyecto es relevante para la toma de información del contenido de grandes volúmenes de formularios (encuestas).

*Roberto Carlos de Armas Domínguez*

*Lener Samuel Bautista Rodríguez*

Estudiantes

Ingeniería Electrónica Documentación e Información

Universidad Industrial de Santander (UIS)

- En la universidad Pontificia Bolivariana seccional Bucaramanga se han realizado investigaciones en algunos campos que son motivo de desarrollo en el presente proyecto, tal como el reconocimiento de placas de automóviles usando redes neuronales dirigido por el Ingeniero Juan Carlos Villamizar<sup>2</sup>.

*Anderson Ernesto Ballesteros Cárdenas*

*Zaira Liliana Caballero Moreno*

Egresados

Facultad de Ingeniería Electrónica

---

<sup>1</sup>DE ARMAS DOMÍNGUEZ, Roberto Carlos; BAUTISTA RODRÍGUEZ, Lener Samuel. Reconocimiento Automático de Caracteres Manuscritos y Marcas, orientado al procesado de Encuestas. Bucaramanga, 2003. Tesis (pregrado). Universidad Industrial de Santander (UIS). Escuela Ingeniería Eléctrica, Electrónica y Telecomunicaciones. Disponible en internet: <http://unired.uis.edu.co/bibresultados.do?instituciones=1&n=10&q=gualdrón&tidctos=2>, consultado el 18 de enero de 2009.

<sup>2</sup> BALLESTEROS CÁRDENAS, Anderson E.; CABALLERO MORENO, Zaira L. Reconocimiento de Placas de Automóviles usando Redes Neuronales. Bucaramanga, 2006. Trabajo de grado. Universidad Pontificia Bolivariana (UPB). Facultad de Ingeniería Electrónica.

Universidad Pontificia Bolivariana (UPB).

## **OBJETIVOS**

### **OBJETIVO GENERAL**

- Realizar el reconocimiento de caracteres manuscritos por medio del método de correlación y redes neuronales.

### **OBJETIVOS ESPECÍFICOS**

- Tomar caracteres manuscritos por medio de lápiz ópticos.
- Implementar el reconocimiento de caracteres manuscritos por medio de la correlación.
- Implementar el reconocimiento de caracteres manuscritos por medio de redes neuronales.
- Probar diferentes topologías de redes neuronales en el reconocimiento de caracteres manuscritos.

### CONCEPTOS BÁSICOS DE LAS IMÁGENES

En Matlab una imagen a escala de grises es representada por medio de una matriz bidimensional de  $m \times n$  elementos en donde  $n$  representa el número de píxeles de ancho y  $m$  el número de píxeles de largo. Por otro lado, una imagen de color RGB es representada por una matriz tridimensional  $m \times n \times p$ , donde  $m$  y  $n$  tienen la misma significación que para el caso de las imágenes de escala de grises mientras  $p$  representa el plano, que para RGB puede ser 1 para el rojo, 2 para el verde y 3 para el azul<sup>3</sup>.

#### 1.1 CLASES DE IMÁGENES (Según el tipo de datos de sus elementos)

<b>Double:</b>	Doble precisión, números en punto flotante que varían en un rango aproximado de $-10^{308}$ a $10^{308}$ (8 bytes por elemento).
<b>uint8:</b>	Enteros de 8 bits en el rango de [0,255] (1 byte por elemento).
<b>uint16:</b>	Enteros de 16 bits en el rango de [0, 65535] (2 bytes por elemento).
<b>uint32:</b>	Enteros de 32 bits en el rango de [0, 4294967295] (4 bytes por elemento).
<b>int8:</b>	Enteros de 8 bits en el rango de [-128, 127] (1 byte por elemento).
<b>int16:</b>	Enteros de 16 bits en el rango de [-32768, 32767] (2 bytes por elemento).
<b>int32:</b>	Enteros de 32 bits en el rango de [-2147483648,2147483647] (4 bytes por elemento).
<b>Single:</b>	Número en punto flotante de precisión simple, con valores aproximadamente en el rango de $-10^{38}$ a $10^{38}$ (4 bytes por elemento).
<b>Char:</b>	Caracteres (2 byte por elemento).
<b>Logical:</b>	Los valores son 0 ó 1 (1 byte por elemento). <sup>4</sup>

#### 1.2 TIPOS DE IMÁGENES

- **Imágenes RGB:** Una imagen RGB representa el color de cada píxel como un conjunto de tres valores que contienen la intensidad de rojo, verde y azul para conformar el color.
- **Imágenes de intensidad:** Representa una imagen en escala de grises.
- **Imágenes binarias:** El valor de intensidad solo admite valores 1 ó 0.

---

<sup>3</sup> CUEVAS JIMENEZ, Erik Valdemar y ZALDIVAR NAVARRO, Daniel. Visión por Computador utilizando Matlab y el Toolbox de Procesamiento Digital de Imágenes [online]. [Consultado el 15 de mayo de 2009]. Disponible en internet: <proton.ucting.udg.mx/tutorial/vision/cursovision.pdf>.

<sup>4</sup> GARCÍA PÉREZ, David. Grupo de Visión Artificial. Manejo Básico de Imágenes con Matlab [online]. [Consultado el 15 de mayo de 2009]. Disponible en internet: <http://gva1.dec.usc.es/~antonio/docencia/20042005tci/practicas/apuntes-practica1.pdf>

- **Imágenes indexadas:** En una imagen indexada, sólo se usan un conjunto limitado de colores discretos, usualmente 256 o menos. Estos colores forman un "mapa de colores" de la imagen, y a cada punto de la imagen se le asigna un color de este mapa.

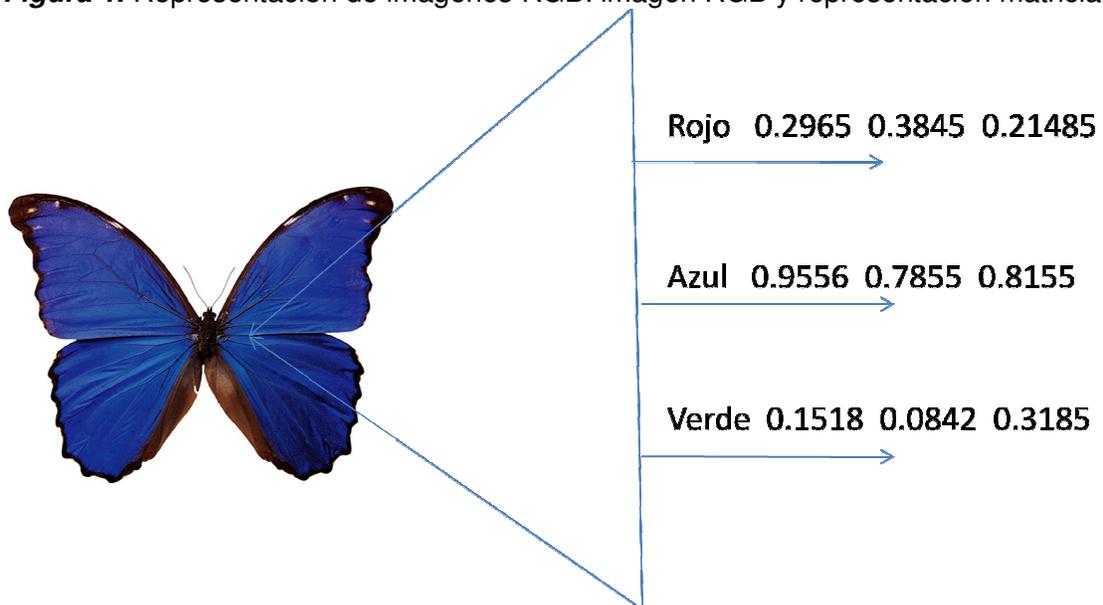
**1.2.1 Imágenes RGB.** Debido a la estructura del ojo humano, hay tres ejes en los sistemas que pretenden mostrar el color. Una imagen RGB representa el color de cada píxel como un conjunto de tres valores que contienen la intensidad de rojo, verde y azul para conformar el color. Estas intensidades son almacenadas directamente en matrices imagen. En Matlab, las componentes de color rojo, verde y azul de la imagen RGB residen en una matriz  $m \times n \times 3$ , donde  $m$  y  $n$  son el número de filas y de columnas de la imagen, y la tercera componente representa la dimensión, indicando que consta en este caso de 3 planos, conteniendo los colores de rojo, verde y azul de cada píxel.<sup>5</sup>

Por ejemplo, para determinar el color del píxel (28, 37) se ha de buscar en la tripleta RGB almacenada en (28, 37, 1.3). Suponiendo que el valor contenido en (28, 37, 1) fuera 0.5748, el de (28, 37, 2) fuera 0.1754 y por último, el de (28, 37, 3) fuera 0.6254, el color del píxel sería:

0.5748 0.1754 0.6254

Una matriz imagen que almacena un tipo RGB puede ser de tipo double, en cuyo caso el margen de los valores es [0,1], o de tipo uint8, y entonces el margen corresponde a [0,255].

**Figura 1.** Representación de imágenes RGB: imagen RGB y representación matricial

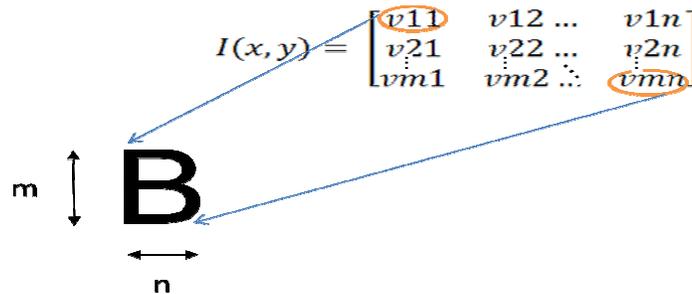


*Fuente:* Autor del proyecto

<sup>5</sup> HERVELLA AZOUZI, Samira. Editor de Imágenes basado en Regiones. Aplicación en entorno Matlab [online]. I.T.Telecomunicaciones,esp. Sonido e Imagen EUETIT.-Terrassa 2005-2006. [Consultado el 15 de mayo de 2009]. Disponible en internet: <<http://upcommons.upc.edu/pfc/bitstream/2099.1/3860/1/54960-1.pdf>>.

**1.2.2 Imágenes de Intensidad.** “Matlab almacena una imagen de intensidad o en escala de grises como una matriz simple, en la cual, cada elemento corresponde con un píxel de la imagen. La matriz puede ser de clase double, caso en el cual dicha matriz contiene valores en el margen [0,1], o de clase uint8, en cuyo caso el margen es [0,255]”<sup>6</sup>.

**Figura 2.** Representación de una imagen a escala de grises en Matlab



Fuente: Autor del proyecto

**1.2.3 Imágenes binarias.** Son imágenes que solamente tienen valor 0 ó 1. Se representan en Matlab a partir de arreglos lógicos. En una imagen binaria, cada píxel puede tomar uno de dos posibles valores discretos. Básicamente estos valores podrían ser respuesta en verdadero (matriz bidimensional de 1's) o falso (matriz bidimensional de 0's). Una imagen binaria puede ser considerada una clase especial de imagen de intensidad, conteniendo solo los valores correspondientes 1 y 0 lógico. Una imagen binaria puede ser almacenada en una matriz de clase double o clase uint8. Como es lógico suponer en una imagen de estas condiciones es mucho más fácil encontrar y distinguir características estructurales.

**1.2.4 Imágenes indexadas.** “Una imagen indexada almacena una imagen como dos matrices. La primera matriz tiene el mismo tamaño que la imagen y un número para cada píxel. La segunda matriz se denomina mapa de color y su tamaño puede diferir del de la imagen”<sup>7</sup>. Los números de la primera matriz son una instrucción de qué color usar según el mapa de color (colormap), de modo que se utiliza un conjunto limitado de colores discretos que usualmente varían entre 256 o menos. El colormap es una matriz mx3 de clase double. Cada fila de la matriz colormap especifica los valores de rojo, verde y azul (RGB) de un determinado color:

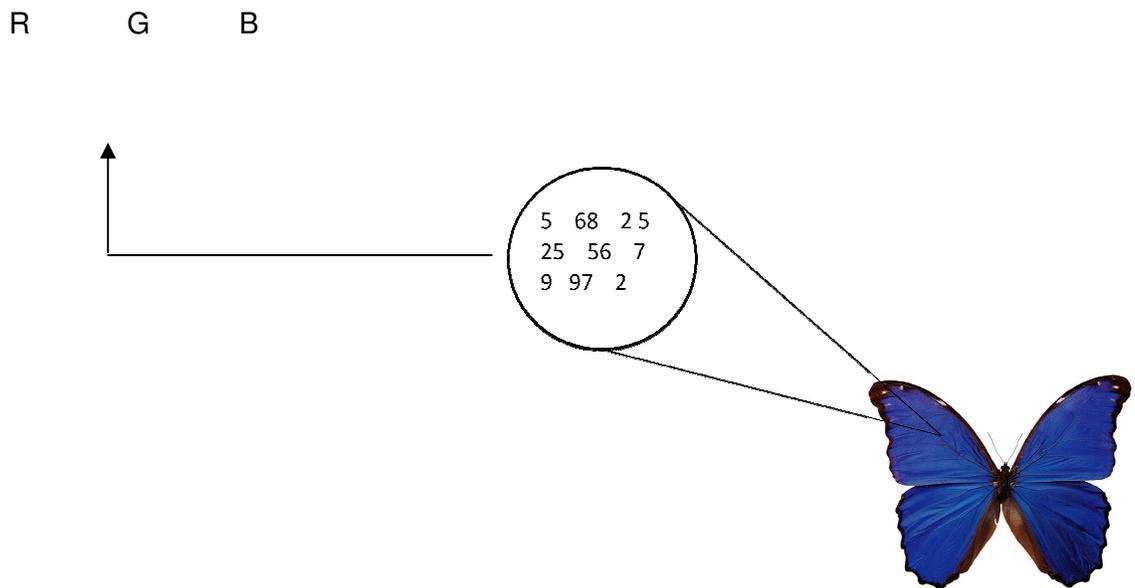
Color = [R G B]; R, G y B son valores reales dentro del margen 0 (negro) a 1 (máxima intensidad). La figura 3 ilustra la estructura de una imagen indexada, en donde los píxeles en la matriz de la imagen son representados por enteros, los cuales son punteros o índices a los valores de un color almacenado en la matriz colormap. Si la imagen indexada fuera una imagen en niveles de gris y no en color todos los valores de R, G y B tendrían el mismo nivel de intensidad por lo que los i, j de la matriz colormap serían iguales dentro de la misma fila.

<sup>6</sup> Ibíd., p. 36.

<sup>7</sup> Disponible en internet: <http://www.udb.edu.sv/Academia/Laboratorios/biomedica/Imagenes%20Medicas/guia1IM.pdf>, consultado el 9 de mayo de 2009.

La relación entre los valores de la matriz imagen y su correspondencia con los colores en la matriz colormap dependen de si la matriz imagen es de clase double o uint8. Si la matriz imagen es de clase double, el valor 1 apunta a la primera fila del colormap, el valor 2 a la segunda, y así progresivamente. Si es de clase uint8, se produce un desplazamiento de tal manera que el valor 0 apunta a la primera fila del colormap, el valor 1 a la segunda, etc.

**Figura 3.** Representación de imágenes indexadas: imagen indexada original, parcial de la matriz imagen y matriz colormap.



Fuente: Autor del proyecto

### 1.3 LECTURA Y ESCRITURA DE IMÁGENES

“Es posible utilizar la función de Matlab *imread* para leer imágenes desde ficheros con el formato gráfico BMP, HDF, JPEG, PCX, TIFF y XWD, además del formato propio MAT. Para escribir imágenes en ficheros, Matlab aporta la función *imwrite*, que soporta los mismos formatos que *imread*”<sup>8</sup>.

Para leer imágenes contenidas en un archivo al ambiente de Matlab se utiliza la función *imread*, cuya sintaxis es:

`imread ('nombre del archivo')`

Donde el nombre del archivo es una cadena de caracteres conteniendo el nombre completo de la imagen con su respectiva extensión. Los formatos y extensiones de imágenes que soporta Matlab son los mostrados en la tabla.1.

<sup>8</sup> HERVELLA AZOUZI, Op. Cit., p. 37.

**Tabla 1.** Formatos y extensiones soportadas por Matlab

FORMATO	EXTENSION
TIFF	.tiff
JPEG	.jpg
GIF	.gif
BMP	.bmp
PNG	.png
XWD	.xwd

*Fuente:* CUEVAS JIMENEZ, Erik Valdemar y ZALDIVAR NAVARRO, Daniel. Visión por Computador utilizando Matlab y el Toolbox de Procesamiento Digital de Imágenes [online]. [Consultado el 15 de mayo de 2009]. Disponible en internet: <proton.ucting.udg.mx/tutorial/vision/cursovision.pdf>.

Para introducir una imagen guardada en un archivo con alguno de los formatos especificados en la tabla anterior solo tiene que usarse la función *imread* y asignar su resultado a una variable que representará a la imagen. De tal forma que si se quisiera introducir la imagen contenida en el archivo *data.jpg* a una variable para su procesamiento en Matlab, entonces se tendría que escribir en línea de comandos:

> >image=imread('data.jpg'), Con ello la imagen contenida en el archivo *data.jpg* quedará contenida en la variable *image*. Una vez que la imagen está contenida en una variable de Matlab es posible utilizar las funciones para procesar la imagen. Por ejemplo, una función que permite encontrar el tamaño de la imagen es *size* (variable).

> >[m, n]=size(image), En donde *m* y *n* contendrán los valores de las dimensiones de la imagen. Para grabar el contenido de una imagen en un archivo se utiliza la función *imwrite* (variable, "nombre del archivo"), en donde *variable* representa la variable que contiene a la imagen y *nombre del archivo* el nombre del archivo con su respectiva extensión de acuerdo a la tabla 1.

Suponiendo que la variable *image2* contiene la imagen que nos interesa grabar en el archivo *dato2.jpg* tendríamos que escribir:

> >imwrite(image2, 'data2.jpg'), Después que se realiza un procesamiento con la imagen, es necesario desplegar el resultado obtenido, la función *imshow* (variable) permite desplegar la imagen en una ventana en el ambiente de trabajo de Matlab. Si la variable a desplegar por ejemplo, es *face* al escribir en la línea de comandos<sup>9</sup>:

> >imshow(face)

---

<sup>9</sup> CUEVAS y ZALDIVAR, Op. Cit., p. 5.

## 1.4 FUNCIONES PARA LA CONVERSIÓN DE IMÁGENES

Para las conversiones a los diferentes tipos de imágenes, se han utilizado herramientas pertenecientes al Toolbox de procesamiento de Imagen de Matlab. Los formatos para las diferentes funciones son los siguientes:

- Escala de grises: La función *rgb2gray* convierte imágenes RGB o colormap en escala de grises.

$I = \text{RGB2GRAY}(RGB)$ , El formato de representación de color ofrecido por las imágenes RGB resulta no apropiado para aplicaciones en las cuales el cambio de iluminación es un problema. Matlab dispone de funciones especiales para realizar cambios entre modelos de color y para convertir imágenes de color a escala de grises, tal como lo permite hacer la función *rgb2gray*. La función *rgb2gray* cambia una imagen en formato RGB a escala de grises y el resultado lo guarda en un variable.

$\text{NEWMAP} = \text{RGB2GRAY}(\text{MAP})$ , Esta función también da la posibilidad de devolver un colormap en escala de grises equivalente al valor de entrada MAP.

*RGB*: Es la imagen de entrada.

*Map*: El colormap de entrada.

- Binaria: Esta función *im2bw* convierte imágenes RGB, en escala de grises o indexadas a imágenes binarias tomando un valor de umbral.

$\text{BW} = \text{IM2BW}(I, \text{LEVEL})$ , La imagen es convertida a blanco y negro, utilizando un valor de umbral automático.

$\text{BW} = \text{IM2BW}(X, \text{MAP}, \text{LEVEL})$ , Convierte imágenes indexadas con un colormap determinado a imágenes binarias, en blanco y negro<sup>10</sup>.

*X, I* son imágenes de entrada.

*Level* es el valor de umbral.

Además de las funciones presentadas anteriormente, también se puede tener en cuenta la función *rgb2ind* que permite convertir una imagen en RGB a una imagen indexada y la función *mat2gray*, que permite pasar de una matriz bidimensional a una imagen de intensidad.

**1.4.1 Herramientas para la conversión entre tipos de datos.** Los elementos que constituyen una imagen en Matlab tienen el formato entero *uint8*, que es un tipo de dato que puede variar de 0 a 255, sin poder soportar decimales y valores que salgan fuera de ese rango. Lo anterior resulta una desventaja principalmente en aquellos casos donde se implementan algoritmos que trabajan con este tipo de datos para realizar operaciones de división o multiplicación por tipo de dato flotante. En estos casos es necesario transformar la imagen de tipo de dato *uint8* a *double*. Es importante tener en cuenta que si se utiliza la

---

<sup>10</sup> HERVELLA AZOUZI, Op. Cit., p. 45.

función *imshow* para desplegar las imágenes; ésta no tiene la capacidad de poder desplegar imágenes del tipo *double* por lo que una vez realizado las operaciones de punto flotante es necesario después convertir al tipo de dato *uint8*. Supongamos que se debe tener una imagen de escala de grises representada en la variable *imagegray* y se quiere reducir sus intensidades a la mitad, entonces se escribe:

```
> >imagegrayD=double(imagegray) ;
> >imagegrayD=imagegrayD*0.5 ;
> >imagegray=uint8(imagegrayD) ;
> >imshow(imagegray) ;11
```

En la tabla 2, se pueden observar los comandos y sus funciones para la conversión del tipo de dato de imagen deseada.

**Tabla 2.** Comandos para conversión de datos

COMANDO:	CONVIERTE A:	TIPO VÁLIDO DE ENTRADA
im2uint8	uint8	logical, uint8, uint16 y double
im2uint16	uint16	logical, uint8, uint16 y double
mat2gray	double ([0,1])	double
im2double	double	logical, uint8, uint16 y double
im2bw	logical	uint8, uint16 y double

Fuente: Autor del proyecto

- **UINT8:** Convierte los datos de la imagen de entrada a enteros de 8-bit sin signo.

$I = \text{UINT8}(X)$ : Convierte los elementos de la matriz a enteros de 8-bits sin signo. La matriz (X) puede ser un objeto numérico de tipo Double. Los valores UIN8 van desde 0 a 255.

Donde X es la matriz de entrada.

- **UINT16:** Convierte los datos de la imagen de entrada a enteros de 16-bit sin signo.

$I = \text{UINT16}(X)$ : Convierte los elementos de la matriz a enteros de 16-bits sin signo. La matriz (X) puede ser un objeto numérico de tipo double. Los valores UIN16 van desde 0 a 65535.

<sup>11</sup> CUEVAS y ZALDIVAR, Op. Cit., p. 14.

Donde X es la matriz de entrada.

- *DOUBLE*: convierte los datos a double precisión, por defecto en Matlab los datos se crean en este tipo de precisión.

*I=DOUBLE(X)*: devuelve datos de tipo double para los elementos de la matriz X. Si X ya es una matriz de elementos double, la función no tiene efectos sobre ella.

Donde X es la matriz de entrada<sup>12</sup>.

## 1.5 ACCESO A PÍXELES Y PLANOS EN IMÁGENES

El acceso a píxel de una imagen es una de las operaciones más comunes en visión computacional y en matlab está sumamente simplificado; solo bastará con indexar el píxel de interés en la estructura de la imagen. Consideremos que se tiene una imagen *image1* en escala de grises y se desea obtener su valor de intensidad en el píxel especificado por  $m=100$  y  $n=100$ ; solo tendríamos que escribir:

```
> >image1(100,100)
ans =
84
```

De igual forma si se desea cambiar el valor de este píxel a negro, es decir asignarle el valor de 0 lo que tendría que escribirse en línea de comandos es:

```
> >image1(100,100)=0 ;
```

En el caso de imágenes a escala de grises éstas solo tienen un plano, constituido por la matriz  $m \times n$  que contiene los valores de intensidad para cada índice. Sin embargo las imágenes de color cuentan con más de un plano. En el caso de imágenes RGB, éstas cuentan con 3 planos uno para cada color que representa. Consideremos ahora que una imagen RGB está contenida en la variable *image2* y se desea obtener cada uno de los planos que la componen. Entonces se escribe

```
> >planeR=image2( :, :,1) ;
> >planeG=image2( :, :,2) ;
> >planeB=image2( :, :,3) ;
```

Si se deseará manipular un píxel de una imagen a color RGB este tendrá un valor para cada uno de sus planos correspondientes. Supongamos que se tiene la imagen RGB contenida en la variable *image2* y se desea obtener el valor del píxel  $m=100$  y  $n=100$  para cada uno de los diferentes planos R, G y B.

Se tendría que escribir:

```
> >valueR=image2(100,100,1) ;
> >valueG=image2(100,100,2) ;
> >valueB=image2(100,100,3) ;
```

---

<sup>12</sup> HERVELLA AZOUZI, Op. Cit., p. 45.

Lo cual dará como resultado una tripleta de valores. De igual forma que con el caso de escala de grises se puede modificar este píxel a otro color mediante el cambio de su valor en cada uno de sus respectivos planos; por ejemplo un cambio a color *blanco* mediante:

```
> >image2(100,100,1)=255;  
> >image2(100,100,2)=255;  
> >image2(100,100,3)=255;
```

En ocasiones resulta preferible saber el color o la intensidad de gris (el valor del píxel) de forma iterativa, es decir tener la posibilidad de seleccionar un píxel en una región y obtener el valor de éste. Esta posibilidad es ofrecida por la función *impixel*, la cual iterativamente entrega el valor (uno o tres) del píxel seleccionado que aparezca en la ventana desplegada por la función *imshow*. El formato de esta función es:

```
>> value=impixel;
```

Donde *value* representa un escalar, en el caso de que la imagen este en escala de grises o bien un vector de 1 x 3 con los valores correspondientes a cada uno de los planos RGB. Para utilizar esta función es necesario antes, desplegar la imagen con la función *imshow*. Una vez desplegada se llama a la función y cuando el cursor del ratón esté sobre la superficie de la imagen cambiara a una +. Cuando se presione el botón izquierdo del ratón se seleccionara el píxel, el cual se puede seleccionar otra vez en caso de que se haya cometido un error a la hora de posicionar el ratón, ya que la función seguirá activada hasta que se presione la tecla de enter.

Una operación importante en visión computacional es el determinar un perfil de la imagen; es decir convertir un segmento de la imagen a una señal unidimensional para analizar sus cambios. matlab dispone de la función *improfile* que permite trazar el segmento iterativamente con el ratón, desplegando después el perfil de la imagen en una grafica diferente. Esta función necesita que la imagen original sea previamente desplegada mediante la función *imshow*. Debe de considerarse que si la imagen es a escala de grises, el perfil mostrará solo una señal correspondiente a las fluctuaciones de las intensidades de la imagen, sin embargo si la señal es de color RGB ésta mostrará un segmento de señal para cada plano. Para la utilización de esta función solo es necesario escribir en línea de comandos:

```
> >improfile
```

Como es una función iterativa en cuanto el ratón se encuentra en la superficie de la imagen el puntero cambiara de símbolo a una +, de esta manera se puede modificar mediante el establecimiento de una línea en la imagen configurar el perfil deseado.<sup>13</sup>

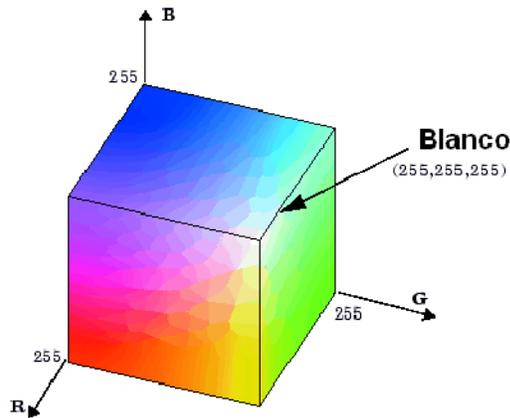
**1.5.1 Colores de una imagen.** Los colores de una imagen pueden variar dependiendo del dispositivo en que ésta fue capturada. De modo que, no todos los dispositivos en los que se realiza la adquisición de datos off-line generaran gamas de colores idénticas, por lo contrario cada una de ellas están sujetas a resoluciones, pixelajes y amplitudes diferentes. Esto podría llegar a ser un problema por la infinita cantidad de gamas de

---

<sup>13</sup> CUEVAS y ZALDIVAR, Op. Cit., p. 9.

colores posibles que se generan de la combinación del vector de componentes principales, R (rojo), G (verde) y B (Azul).

**Figura 4.** Representación binaria GRB



*Fuente:* Autor del proyecto

Las posibles representaciones de estos 3 principales componentes son:

RGB = [255, 0 ,0]	ROJO
RGB = [0, 0, 0]	NEGRO
RGB = [0, 0, 255]	AZUL
RGB = [255, 255, 255]	BLANCO

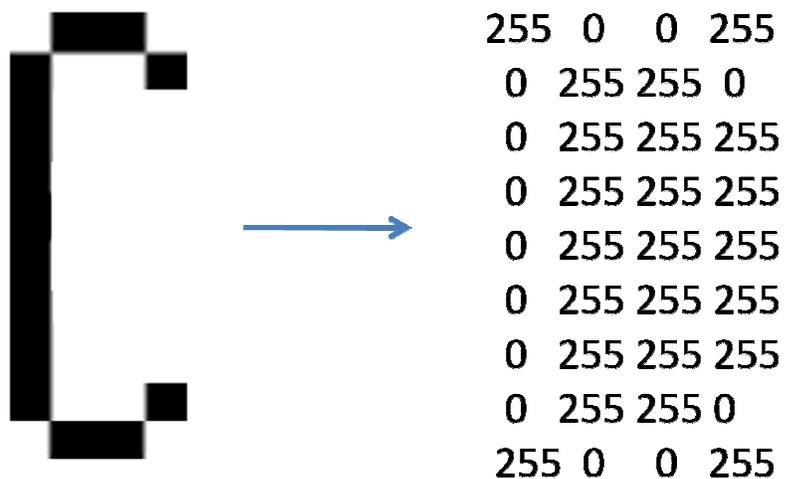
**1.5.2 Representación de colores.** Como se puede visualizar en la figura 4, cada color corresponde a un elemento del siguiente conjunto, de esta manera cada color se puede representar por medio de una representación matemática:

$$V = \{(X, Y, Z) / X, Y, Z \in \{0, 1, 2, \dots, 255\}\}$$

Debido a la inmensa cantidad de colores que puede ser procesada una imagen se realiza una conversión de cada uno de los elementos vectoriales de modo en que solo podrían pixelarse vectores [255, 255, 255] y vectores [0, 0, 0]. Cualquier vector diferente a estos deberá inclinarse a ser blanco o negro dependiente a cierto rango establecido en la conversión de colores RGB a escalas de grises.

Ahora, se estaría limitando los colores de la imagen únicamente a puntos negros o blancos lo que hace más fácil procesar y extraer los patrones de las estructuras contenidas en las imágenes o caracteres.

**Figura 5.** Representación binaria de una letra



*Fuente:* Autor del proyecto

## 1.6 FILTRAJE ESPACIAL

Se debe entender por ruido en imágenes digitales cualquier valor de un píxel de una imagen que no se corresponde exactamente con la realidad. Cuando se adquiere una imagen digital, está contaminada por ruido. El ruido se debe, la mayoría de las veces al equipo electrónico utilizado en la captación de las imágenes (ruido de cuantificación de la imagen, efecto de niebla en la imagen... etc.) y al ruido añadido en los tramos de transmisión (posibles interferencias o errores al transmitir los bits de información).

Se van a distinguir dos clases diferentes de ruido:

- Ruido gaussiano: Se caracteriza por tener un espectro de energía constante para todas las frecuencias. Cuando se presenta este problema, el valor exacto de cualquier píxel es diferente cada vez que se captura la misma imagen. Este efecto, suma o resta un determinado valor al nivel de gris real y es independiente de los valores que toma la imagen.
- Ruido impulsivo: Se caracteriza por la aparición de píxeles con valores arbitrarios normalmente detectables porque se diferencian mucho de sus vecinos más próximos. La distribución viene dada por:

Donde  $r(x, y)$  es un número aleatorio con distribución uniforme en  $[0,1)$  y  $p$  es la probabilidad de ocurrencia del ruido aleatorio, es decir, el porcentaje de puntos de la imagen que se verán afectados por el ruido impulsivo del total de puntos de la imagen<sup>14</sup>.

El filtrado lineal da un mejor resultado a la hora de eliminar ruido gaussiano, mientras que el filtrado no lineal es más adecuado a la hora de eliminar ruido impulsivo. Por ello, según las necesidades del proyecto se escogió un filtro no lineal para de esta forma eliminar los píxeles que puedan interponerse en la imagen por el ruido que adquieren cuando son tomadas o creadas desde algún programa de dibujo. Aquí se van a considerar tres filtros no lineales de orden:

- *Filtro de mediana*: Se sustituye cada valor de la imagen por la mediana de los píxeles del vecindario.
- *Filtro de mínimo*: Se elige el mínimo de los píxeles del vecindario.
- *Filtro de máximo*: Seleccionaremos el máximo del vecindario<sup>15</sup>

En el filtro de la mediana la máscara de convolución se emplea de forma distinta que en el resto de filtros:

- Se toman todos los valores de la imagen cubiertos por la máscara
- Se ordenan de menor a mayor
- Se toma el valor central<sup>16</sup>

**Figura 6.** Filtro de la mediana



<sup>14</sup> Disponible en: [http://www.des.udc.es/~adriana/TercerCiclo/CursolImagen/curso/web/Filtrado\\_Espacial\\_Ruido.html](http://www.des.udc.es/~adriana/TercerCiclo/CursolImagen/curso/web/Filtrado_Espacial_Ruido.html), consultado el 17 de mayo de 2009

<sup>15</sup> Disponible en internet: [decsai.ugr.es/mia/practicas/g6/g6.pdf](http://decsai.ugr.es/mia/practicas/g6/g6.pdf), consultado el 18 de mayo de 2009.

<sup>16</sup> Disponible en internet: <http://www.sia.eui.upm.es/master/VA05.pdf>, consultado el 18 de mayo de 2009.

**1.6.1 Función medfilt2.** La función medfilt2 se basa en el filtro de mediana aplicado para funciones o señales en 2D como las imágenes, recibe como parámetros de entrada la imagen distorsionada y la vecindad a elegir tiene como valor por defecto de vecindad el de [3,3].

$Y = \text{medfilt2}(x, [n1 \ n1])$

## 1.7 SUBMUESTREO DE IMÁGENES

En ocasiones es necesario hacer cálculos que requieren procesar por completo la imagen, en estos casos hacerlo sobre la resolución original de la imagen sería muy costoso. Una alternativa más eficiente, resulta el sub-muestreo de la imagen. Sub-muestreo significa generar una imagen a partir de tomar muestras periódicas de la imagen original, de tal forma que ésta quede más pequeña. Si se considera la imagen  $I(m,n)$  definida como:

$$I(m,n) = \begin{bmatrix} v_{11} & v_{12} \dots & v_{1n} \\ v_{21} & v_{22} \dots & v_{2n} \\ v_{m1} & v_{m2} \dots & v_{mn} \end{bmatrix}$$

y se desea sub-muestrear la imagen para obtener la mitad de su tamaño original, así la nueva imagen quedaría compuesta por los elementos tomando *uno sí y otro no* de la imagen original:<sup>17</sup>

$$I_s(m,n) = \begin{bmatrix} v_{11} \dots & v_{13} & \dots & v_{1(n-2)} \\ v_{31} & v_{33} & \dots & v_{3(n-2)} \\ v_{(m-2)1} & v_{(m-2)} \dots & v_{(m-2)(n-2)} \end{bmatrix}$$

## 1.8 CONECTIVIDAD DE COMPONENTES (PÍXELES) EN LA IMAGEN

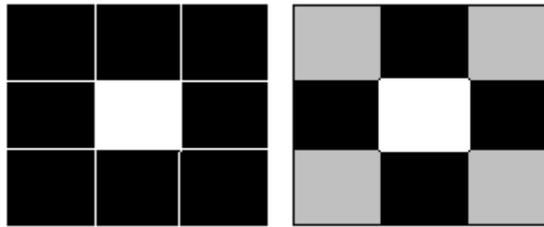
En una imagen binaria, puede definirse un objeto como un conjunto de píxeles conectados con valor 1. Para muchas operaciones la distinción de objetos depende la convención utilizada de conectividad, que es la forma en la que se considera si dos píxeles tienen relación como para considerar que forman parte del mismo objeto. La conectividad puede ser de dos tipos, de conexión-4 ó bien conexión-8. En la figura 7 se esquematiza ambas conectividades.

En la conectividad conexión-8 se dice que el píxel blanco pertenece al mismo objeto si existe un píxel de valor uno en las posiciones 1, 2, 3, 4, 5, 6, 7 y 8 (píxeles negros). Por su parte la conectividad conexión-4 relaciona solo a los píxeles 2, 4, 5 y 7 (píxeles negro).

---

<sup>17</sup> CUEVAS y ZALDIVAR, Op. Cit., p. 15.

**Figura 7.** Conectividad conexión de 8 y 4



*Fuente:* Autor del proyecto

Para las operaciones que consideran conectividad como un parámetro, es importante tomar en cuenta que ésta determina fuertemente el resultado final del procesamiento puesto que puede dar origen a objetos nuevos en donde si se hubiera elegido otra conectividad no existirán.

La función `bwlabel` realiza un etiquetado de los componentes existentes en la imagen binaria, la cual puede ser considerada como una forma de averiguar cuántos elementos (considerando como elementos agrupaciones de unos bajo alguno de los dos criterios de conectividad) están presentes en la imagen. La función tiene el siguiente formato:

`ImagenR=bwlabel (ImagenS, conectividad);`

Donde `ImagenR` es la imagen resultado que contiene los elementos etiquetados con el número correspondiente al objeto, `ImagenS` es la imagen binaria que se desea encontrar el número de objetos y conectividad puede ser 4 o 8 (correspondiendo al tipo de conectividad anteriormente explicado). El efecto de esta función puede explicarse fácilmente si se analiza la imagen original y el resultado de aplicar la función `bwlabel`, como por ejemplo una imagen de la cual se desea saber cuántos caracteres contiene, la variable `ImagenR` contendría los caracteres etiquetados con su número correspondiente.

La imagen resultado asigna a cada píxel perteneciente a un determinado objeto según su conectividad la etiqueta perteneciente al número de objeto mientras que los píxeles en *cero* no tienen efecto en la operación. La imagen resultado es del tipo `double`. Además debido a su contenido (los valores son muy pequeños) no puede ser desplegada por la función `imshow`.

De esta forma cada objeto de la imagen resultado al aplicar la función `bwlabel` puede ser identificado más rápidamente. Es importante recalcar que el número de índices necesarios para formar la imagen es igual al número de etiquetas encontradas por la función `bwlabel` más uno, ya que el fondo constituido de solo *ceros* también es un índice más<sup>18</sup>.

<sup>18</sup> CUEVAS y ZALDIVAR, Op. Cit., p. 22.

### RECONOCIMIENTO ÓPTICO DE CARACTERES

Para el ser humano es relativamente fácil poder reconocer las características de texto manuscrito o impreso cuando para la computadora se exigen una serie de etapas que a pesar de que se puedan procesar en un tiempo muy corto se necesitan que sean precisas, ya que la diferencia de los patrones de los caracteres para el ojo pueden ser distintivos mientras que la máquina se limita a clasificar patrones y a compararlos. Por eso es importante escoger un método de clasificación eficaz para el reconocimiento.

El objetivo del reconocimiento de caracteres óptico, es identificar patrones que puedan ser identificados como caracteres alfanuméricos dentro del contexto de una imagen digital. El proceso básico de un reconocedor de caracteres ópticos esta en las etapas de segmentación, extracción de patrones, y por último clasificación. Un sistema OCR tiene la capacidad de reconocer tanto caracteres impresos (maquinas) como caracteres manuscritos y son estos últimos en donde recae la complejidad del reconocimiento debido a que se puede encontrar alguna dificultad en la segmentación de los componentes de la imagen o por la variabilidad que se puede encontrar en la escritura por la caligrafía del escritor.

#### 2.1 RECONOCIMIENTO DE CARACTERES MANUSCRITOS

En el reconocimiento de texto manuscrito existen diferentes áreas en las que se pueden trabajar en un OCR. Algunas de estas áreas que son motivo de estudio se definirán a continuación<sup>19</sup>:

- *Reconocimiento del Carácter Manuscrito*: En esta área se podrá procesar el carácter digitalizado, extraer patrones relevantes y comparar muestras tomadas con muestras predeterminadas almacenadas en una base de datos para que por medio de la etapa de clasificación puedan ser reconocidos y convertidos en comandos alfanuméricos que para nuestro caso son los códigos ASCII de 8 bits utilizados en nuestros computadores.
- *Reconocimiento de Gráficos*: Los gráficos son una clase de símbolos (mapas, planos, diagramas, etc.) que también pueden ser reconocidos debidos a que cada uno de ellos presenta características propias. Estos símbolos a diferencia de los caracteres del abecedario son indefinidos y puedan llegar a tener formas definidas (geométricas) o simplemente amorfas.

---

<sup>19</sup> TOSELLI, Alejandro Héctor. Reconocimiento de Texto Manuscrito Continuo. Valencia, 20 de marzo de 2004, 230 p. Tesis Doctoral. Universidad Politécnica de Valencia. Departamento de Sistemas Informáticos y Computación.

- *Identificación de Caligrafía:* En este caso se tendrá en cuenta el estilo de escritura del escritor pero solo se expondrá los casos en donde al momento de escribir el escritor lo haga de manera natural y no condicionado a los múltiples estados emocionales por el entorno en que se puede encontrar la persona.
- *Reconocimiento de Firma:* En esta área se entiende que se puede realizar el reconocimiento de símbolos propios que pueden identificar la autoría o participación de un individuo en un documento.

Para esta tesis solo se hará énfasis en el reconocimiento de caracteres manuscritos e impresos.

## 2.2 IMPLEMENTACIÓN DE UN SISTEMAS OCR

Contemplar la posibilidad de poder convertir textos manuscritos digitales en texto editable, partiendo del reconocimiento de cada uno de los caracteres que conforman cada palabra que a su vez forman textos, implica reconocer e interpretar automáticamente la información contenida en las imágenes a analizar. Con base a la interpretación de caracteres escritos a mano, se pueden dar diversas aplicaciones a este tipo de sistemas tales como evitar convertir manualmente caracteres digital a escritos, ahorrando tiempo y permitiendo editar dichos caracteres con facilidad. Es en ese momento en donde empieza la creación de interfaces prácticas entre el usuario y la máquina que podrán ser de gran utilidad tanto en sectores comerciales como industriales.

**2.2.1 Medición de características.** En visión computacional es de particular interés encontrar mediciones de características tales como el área, centroide y otras de objetos previamente etiquetados o clasificados por la función `bwlabel` con el objetivo de identificar su posición en la imagen. Matlab dispone de la función `imfeature` para encontrar tales características. El formato de esta función es:

Stats=`imfeature` (L, medicion);

Donde Stats es un tipo de dato compuesto que contiene toda la medición indicada en la cadena de texto. Las propiedades de la imagen son una cadena de texto que indica a la función cuales mediciones realizar sobre los objetos contenidos en la imagen. El conjunto de propiedades posibles vienen resumidas en la tabla:

**Tabla 3.** Mediciones realizadas por la función `imfeature`

Area	Image	EulerNumber
Centroid	FilledImage	Extrema
BoundingBox	FilledArea	EquivDiameter
MajorAxisLength	ConvexHull	Solidity
MinorAxisLength	ConvexImage	Extend
Orientation	ConvexArea	PixelList

*Fuente:* CUEVAS y ZALDIVAR, Op. Cit., p. 27.

Es importante notar que stats es un tipo de dato que contiene la medición de todos y cada uno de los objetos contenidos en la imagen binaria, lo cual hace necesario que el tipo de

dato sea indexado; es decir se puede controlar mediante un índice la medición del objeto que se desea. Por ejemplo, si la imagen contiene 4 elementos, se puede acceder a la medición de cada uno de los diferentes objetos escribiendo:

```
stats(1).Medicion  
stats(2).Medicion  
stats(3).Medicion  
stats(4).Medicion
```

Por ejemplo, considerando que se tiene la imagen previamente binarizada (contenida en la variable *imagebinary*) y se intenta encontrar el centroide de 2 caracteres contenidos en tal imagen.

Como primer paso se utiliza la función *bwlabel* y se verifica el número de objetos:

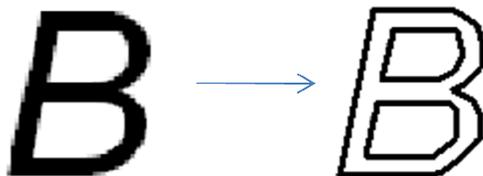
```
> >imageR=bwlabel(imagebinary,8);  
> >max(max(imageR))  
ans=  
2
```

Ahora se utiliza la función *imfeature* para encontrar el centroide de ambas figuras:

```
> >s=imfeature(imageR,Centroid);  
> >s(1).Centroid  
ans =  
81.9922 86.9922  
> >s(2).Centroid  
ans =  
192.5000 85.000020
```

**2.2.2 Funciones para la extracción de bordes.** En la visión por computador es de utilidad hacer reconocimiento de objetos o bien para segmentar regiones, extraer los bordes de objetos (que en teoría delimitan sus tamaños y regiones). La función *edge* da la posibilidad de obtener los bordes de la imagen. La función permite encontrar los bordes a partir de dos diferentes algoritmos que pueden ser elegidos, *canny* y *sobel*. El formato de esta función es:

**Figura 8.** Bordeado de una letra



Fuente: Autor del proyecto

<sup>20</sup> CUEVAS y ZALDIVAR, Op. Cit., p. 27 y 28.

ImageT=edge (ImageS, algoritmo);

Donde ImageT es la imagen obtenida con los bordes extraídos, ImageS es la variable que contiene la imagen en escala de grises a la cual se pretende recuperar sus bordes, mientras que el algoritmo implementado puede ser uno de los dos canny o sobel. Si a la imagen en escala de grises contenida en la variable imagegray se le quieren recuperar sus bordes utilizando en algoritmo canny se escribiría en línea de comandos:

> >ImageR=edge (imagegray, canny);<sup>21</sup>

**2.2.3 Taxonomía de un OCR.** Es la ciencia que permite clasificar y profundizar en el campo del reconocimiento de caracteres. La finalidad de la taxonomía de un OCR tiene como especialidad la determinación e identificación de caracteres diagnósticos que fueron previamente extraídos y sobre eso asociar dichos patrones en un taxón conocido del sistema de clasificación utilizado.

**2.2.4 Formas de adquisición de datos.** Existen a rasgos generales dos técnicas por las cuales se podrán adquirir los datos que serán procesados para su posterior reconocimiento.

- *On-Line:* Los datos se obtienen en tiempo real mientras se escribe. Necesariamente se debe trabajar con un hardware similar al del bolígrafo, que podría ser un lápiz óptico o tabla digitalizadora. El sistema OCR trata con una representación espacio-temporal de los datos de entrada.
- *Off-Line:* Los datos son registrados por medio de un escáner, cámara o lápiz óptico en forma de imagen. El sistema OCR trata con representaciones en un espacio de luminancia.

En el reconocimiento con el tipo de adquisición On-line, la escritura del carácter, está directamente relacionada en función del tiempo. Entonces el reconocimiento se realizaría de manera dinámica y dependerá necesariamente de la rapidez con la que los caracteres sean escritos.

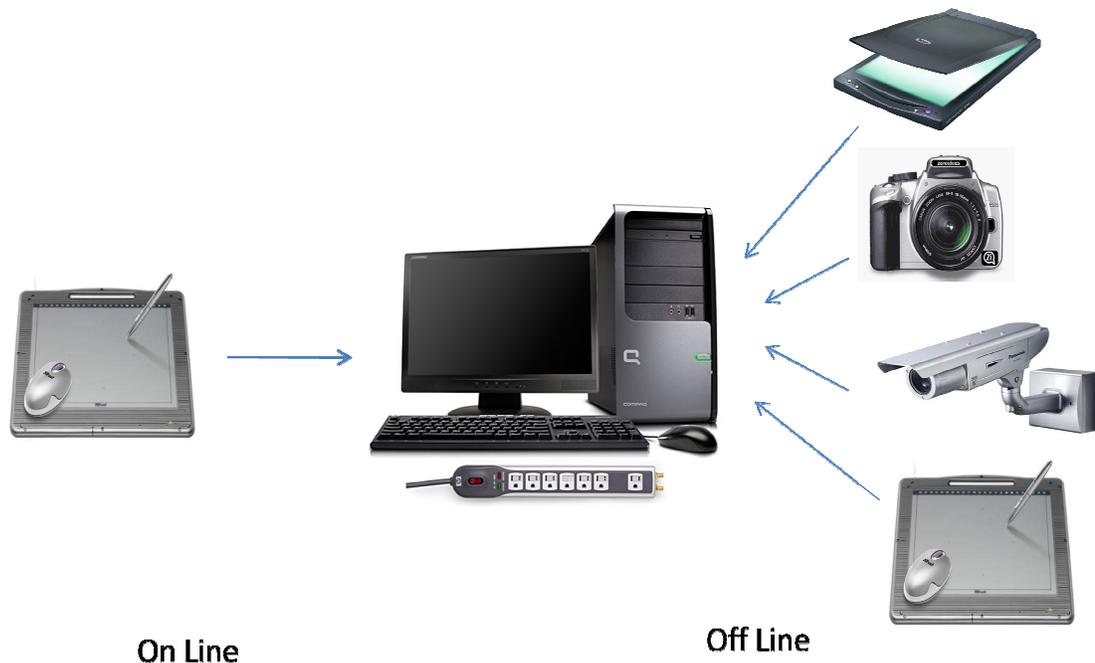
Por otro lado, la adquisición de datos de manera Off-line implica el procesamiento de datos binarios; ya que son imágenes adquiridas por medio de instrumentos electrónicos como cámaras, escáneres, lápiz óptico, etc., por lo que es de gran importancia en la etapa del pre-procesamiento encontrar el área de cada carácter en la imagen para que se pueda aislar y reconocer de manera más sencilla<sup>22</sup>.

---

<sup>21</sup> CUEVAS y ZALDIVAR, Op. Cit., p. 18.

<sup>22</sup> TOSELLI, Op. Cit., p. 7

**Figura 9.** Esquema ilustrativo de algunos dispositivos de adquisición “online” Y “off-line”

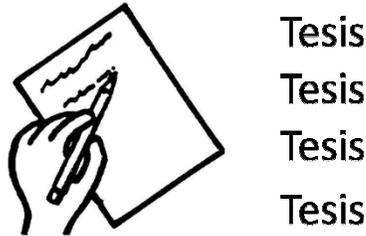


*Fuente:* Autor del proyecto

**2.2.5 Dependencia e independencia del escritor.** Como se mencionaba anteriormente en el reconocimiento de caracteres manuscritos, existe una parte importante del proceso que involucra el estilo de escritura o caligrafía del escritor. Estos datos son de suma importancia en el reconocimiento, ya que los caracteres de un individuo a otro varían significativamente y más aun cuando la caligrafía de ellos ha sido influenciada y desarrollada en diferentes ambientes o regiones. En el caso específico de esta tesis el sistema será dependiente del escritor, por lo cual se crearán varias bases de datos de cada individuo que almacenará los patrones característicos de cada uno de los caracteres del alfabeto.

- Dependencia del escritor: En nuestro sistema OCR se crearán bases de datos en donde se podrán recopilar características específicas de cada uno de los caracteres posibles que puedan escribir cada uno de los individuos. Por lo que al momento de reconocer el carácter capturado de una imagen digitalizada o a través del lápiz óptico y la tabla digitalizadora se debe cerciorar que se esté trabajando sobre la base de datos del correspondiente individuo que es donde existirá patrones muy similares a los extraídos del carácter capturado. Aunque es posible que se pueda reconocer la letra de un individuo en diferentes bases la precisión del reconocimiento no es la misma por lo que las demás serán más propensas a errores. Se decidió trabajar con la dependencia del escritor por la enorme variabilidad que hay en el estilo de escritura de cada autor.

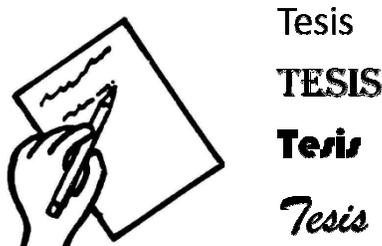
**Figura 10.** Texto de un solo escritor: La caligrafía de un solo escritor tiende a mantener formas, tamaños y textura de los caracteres.



Fuente: Autor del proyecto

- Independencia del escritor: El gran reto que conlleva un sistema de reconocimiento independiente del escritor es la gran variedad de la caligrafía y estilos de cada escritor, por lo que es común que se utilicen filtros que clasifique tipos de letras o técnicas que puedan normalizar los caracteres para que puedan ser comparados con unos únicos patrones representativos de cada carácter. Este tipo de reconocer es mucho más complejo y exige de mecanismos mucho más robustos debe tener estadísticas y predicciones matemáticas.

**Figura 11.** Texto de varios escritores



Fuente: Autor del proyecto

Debido a que cada escritor maneja de forma general una caligrafía única, se pueden distinguir características diferentes en la escritura de cada uno.

Como se puede observar en las figuras mientras que un único escritor tiende a escribir con mucha semejanza los mismos caracteres, en grupo de diferentes escritores se puede observar la diferencia en la que son escritos cada uno de esos mismos caracteres.

**2.2.6 Tamaño del léxico.** Como se pudo establecer en los objetivos de la tesis, el léxico a reconocer se limita a las letras del abecedario, por lo que se considera un léxico restringido y de una base de datos relativamente pequeña. Por otro lado, existen léxicos que no son restringidos y deja el reconocimiento de los caracteres a las interpretaciones que la máquina pueda hacer orientada a través de algoritmos o sistemas de aprendizajes manuales o automáticos como son el caso de la redes neuronales.

Esta limitación del léxico permite que la relación de los patrones en la etapa de clasificación y reconocimiento sea más simple, tanto por la velocidad de búsqueda como la delimitación de patrones. A mayor cantidad de patrones a clasificar, más extensa es la búsqueda, más es el tiempo de procesamiento y mayor la probabilidad de errores. Por lo que se puede concluir que a mayor sea el léxico a clasificar mayor deberá ser nuestro sistema de reconocimiento de caracteres ópticos<sup>23</sup>.

**2.2.7 Aislamiento de los caracteres.** Para reconocer caracteres es necesario, en primer lugar, su localización dentro del contexto de la imagen, teniendo en cuenta, en esta operación el orden en el que se disponen en el mismo y los espacios en blanco de línea.

Según las investigaciones realizadas se pudo escoger 2 métodos de aislamiento que son los más usuales cuando se trata de separar cada carácter de la imagen para su posterior análisis.

El método de aislamiento usado en la clasificación por correlación fue mediante una proyección horizontal y un análisis matricial de la imagen, en donde se detectará los caracteres contenidos en cada uno de los renglones. En el método de aislamiento usado en aplicación de redes neuronales fue mediante la selección manual (haciendo uso del mouse o del lápiz óptico) o automática del carácter (un valor de umbral para permitir el acceso a caracteres que en realidad abarcan un área y la restricción o eliminación de puntos o píxeles que representa área pero en realidad son ruido). A continuación se profundizará solo en los parámetros de aislamiento de caracteres para el método de correlación debido a las condiciones y restricciones que maneja. No se especificarán las condiciones utilizadas para el método de clasificación por redes neuronales porque éste sencillamente consta de una condición de área (que se establecerá por medio del umbral) que se acondicionará en las líneas de programación.

Las condiciones y requisitos para el aislamiento de caracteres por correlación fueron las siguientes:

Existen tres magnitudes que determinan el orden de los caracteres dentro del contexto de una imagen: los renglones de los que consta, las palabras de un renglón y las letras de una palabra.

Para nuestro sistema OCR, se han considerado las siguientes restricciones para facilitar la operación de aislamiento:

- Los renglones que forman el texto no podrán estar solapados ni inclinados. Deberán estar al menos separados por una línea en blanco.
- Los caracteres no pueden estar solapados o unidos.
- Las letras del texto pueden ser de cualquier tipo y tamaño.
- Dentro de un mismo renglón pueden existir palabras con distintos tipos de letra y tamaño.

---

<sup>23</sup> HILERA GONZALEZ, José R., ROMERO VILLAVERDE, Juan P., GUTIERREZ de MESA, José A. Sistema de Reconocimiento Óptico de caracteres (OCR) con Redes Neuronales. Universidad de Alcalá de Henares. Departamento de Ciencias de la Computación.

Teniendo en cuenta estas restricciones y considerando la imagen del texto como una matriz bidimensional en la que cada fila tiene un número de puntos igual al número de columnas (división que viene dada por la propia digitalización de la imagen), entonces las operaciones a realizar para localizar y separar renglones son las siguientes:

- Se hace una proyección horizontal consistente en contar los elementos de tinta que existentes en cada una de las filas, traspasando estos valores a otra matriz, unidimensional, resultado de la proyección, en la que existirán diferentes zonas de densidad de tinta separadas por otras vacías. Cada zona donde la proyección dé un valor no nulo será interpretado como un hipotético renglón.
- Se analiza la matriz unidimensional para detectar los posibles renglones de los que está compuesto el texto. Si se detecta una línea con densidad de proyección no nula y además la anterior estaba en blanco, en esa línea comienza un renglón. A continuación se realiza la misma operación pero a la inversa, se busca la línea posterior que sea blanca y que la anterior no lo fuera, ahí estará el final del renglón. Este método se aplica sucesivamente hasta el final de la matriz de proyección, consiguiendo así delimitar los renglones que forman el texto.

Una vez conocida la situación de los renglones y sus límites, se procede a aislar los caracteres. Como se asumía inicialmente que no existiría solapamiento, se puede realizar una proyección vertical dentro de cada renglón para detectar los posibles caracteres. Esta proyección vertical da un resultado nulo en las zonas donde no existe tinta, lo que representa la separación entre dos caracteres, y resultados no nulos que indican la presencia de caracteres. Los límites en altura, superior e inferior, se detectan analizando la vertical de cada carácter, desde la parte superior e inferior, respectivamente, del renglón que lo contiene.

Por otro lado, se trató de simplificar el método de aislamiento en el método de reconocimiento por redes neuronales, en donde se consigue aislar cada carácter en una ventana rectangular a través del mouse (elemento periférico) con las dimensiones correspondientes su ancho y alto.

Para ambos métodos, es válido el aislamiento según las condiciones, en el caso de existir caracteres de distinto tamaño dentro del mismo texto, dimensiones de la imagen que tendrán que ser normalizadas para una correcta comparación del carácter cuando se genere el fichero ASCII correspondiente a la imagen que se está reconociendo.

**2.2.8 Extracción de características de imágenes.** Después del aislamiento de los caracteres, se procede a la extracción de sus características. Aunque existen otros métodos, como la aplicación de transformada de Fourier, cálculo de momentos, obtención de finales de trazo, ángulos, cruces de trazos, etc., en este primer prototipo de sistema OCR se ha optado por dividir cada "ventana" resultante de aislar cada carácter, en ocho partes verticales por otras ocho horizontales, de tal forma que por cada carácter aislado se obtienen 64 valores representativos que constituyen el denominado vector (o matriz) de características de tal carácter. Mientras que en [3] se proponen 25 valores, en este caso, al trabajar con imágenes reales procedentes de un escáner, tal magnitud resulta

insuficiente, por lo que, después de probar con diferentes tamaños, se ha adoptado el número de 64 por ser suficiente para diferenciar los caracteres.

Las imágenes (ventanas) de los caracteres aislados están compuestas de un número de puntos de imagen o píxeles. Normalmente, este número no coincide con el número de parámetros (64) de su matriz de características. En tal situación pueden darse dos casos:

- Que haya más píxeles en la ventana que valores en la matriz de características. En tal caso, cada valor de la matriz se obtiene como un nivel de gris, dividiendo el número de píxeles que pertenecen al carácter por el número total de píxeles en la ventana que limita la imagen del carácter.
- Que el número de píxeles sea menor. En este caso se aplica un algoritmo de proyección, que repite píxeles hasta completar todos los valores de la matriz de características<sup>24</sup>.

## 2.3 PROCESAMIENTO DE IMÁGENES

Al realizar el pre-procesamiento de una imagen se piensa en tratarla y acondicionarla de tal forma que al entrar en la etapa de extracción de patrones, ésta se pueda ejecutar de la forma más eficiente ya que la imagen debe presentar un mínimo margen de ruido (casi nulo), lo que permite que factores externos no relevantes o ajenos a los caracteres influyan en la clasificación de los mismos. En esta etapa existen tres pasos generales ejecutados y son los siguientes:

**2.3.1 Binarización.** Debido a la cantidad casi innumerable de colores y de sus posibles combinaciones, se decidió trabajar únicamente con imágenes binarias para permitir que el proceso de extracción de caracteres sea más sencillo de encontrar y distinguir características estructurales, ya que se estaría reduciendo información de las mismas. Además, no es objetivo del proyecto establecer los colores de los caracteres sino sus etiquetas.

Además que la binarización de imágenes es una etapa básica en muchas de las aplicaciones que implican el proceso de digitalización de imágenes. El objetivo como tal, es obtener la imagen final representada en dos colores, que por lo general son blancos y negros. Para determinar el color de cada píxel que se encuentra en escala de grises, se establece un umbral (valor numérico de la escala de grises) para determinar si dicho píxel es blanco o negro. De modo que, el algoritmo de binarización hace que la información de la imagen se divida en dos: negro absoluto con un valor de "0" en la matriz o blanco absoluto con un valor de "1". La binarización estaría cumpliendo su objetivo en esta etapa del pre-procesamiento que estaría separando las regiones u objetos de interés.

Supongamos que se tiene la imagen a escala de grises y se binariza aplicando un umbral de 85 en la escala de grises se obtiene así una nueva imagen.

```
> >imagebinary = imagegray<85;
```

---

<sup>24</sup> HILERA, ROMERO, GUTIERREZ, Op. Cit., p. 4.

Se aplica la función `bwlabel` para obtener los objetos contenidos en la imagen binarizada considerando la conectividad conexión-8

```
> >Mat=bwlabel (imagebinary, 8);
```

Se encuentra el número de objetos contenidos en la imagen

```
> >max (max (Mat))  
ans=  
    22
```

Se genera la imagen indexada con 154 elementos<sup>25</sup>.

**2.3.2 Cierre de contornos.** Se pueden considerar dos técnicas básicas: "técnicas binarias" y "técnicas utilizando el gradiente".

- ✓ Técnicas binarias: Una vez que se tiene la imagen binarizada, con 1's (o 255) para los puntos de contornos y 0's para el resto, una estrategia muy sencilla para cerrar contornos es aplicar un cierre a la imagen.
- ✓ Técnicas utilizando el gradiente: Una de las técnicas más simples consiste en analizar las características de los píxeles en un pequeño entorno alrededor de cada punto de la imagen que ha sido etiquetado como punto de contorno. Todos los puntos que son similares de acuerdo con un conjunto de criterios predefinidos son unidos, formando un contorno de píxeles que comparten este criterio. Las dos principales propiedades usadas para establecer estas similitudes de puntos de contornos son:
  1. La magnitud de la respuesta del operador de gradiente utilizado para producir el píxel del contorno.
  2. La dirección del vector gradiente.

Entonces para cada punto de coordenadas  $(i, j)$  se considera una ventana centrada en el punto. Si en la ventana ya hay un punto de contorno  $(i', j')$  tal que ambos tienen magnitudes y direcciones similares de gradiente, entonces  $(i, j)$  es etiquetado también como punto de contorno (y la dirección del borde es perpendicular a la dirección del gradiente).

Este proceso se repite para cada píxel de la imagen. Hay que llevar registro de los nuevos puntos de contorno a medida que se va moviendo el centro de la ventana de píxel a píxel (se les puede asignar un nivel de gris -diferente al de los puntos de contornos iniciales a cada uno de ellos)<sup>26</sup>.

**2.3.3 Operadores morfológicos.** En el proceso de digitalización de las imágenes es muy importante conocer la forma y estructura de los caracteres de modo que sea más

---

<sup>25</sup> CUEVAS y ZALDIVAR, Op. Cit., p. 25.

<sup>26</sup> HERVELLA AZOUZI, Op. Cit., p. 20

fácil el reconocimiento de sus figuras, tamaños, etc., de tal modo que la modificación de los puntos o píxeles de la imagen se pueda realizar para mejorar su contenido.

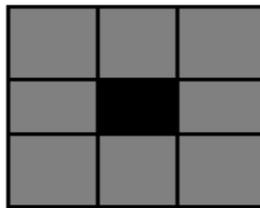
El objetivo de las transformaciones morfológicas es la extracción de estructuras geométricas en los conjuntos sobre los que opera, mediante la utilización de otro conjunto de forma conocida denominado elemento estructurante. El tamaño y la forma de este elemento se escogen, a priori, de acuerdo a la morfología del conjunto sobre el que va interaccionar y de acuerdo a la extracción de formas que se desea obtener<sup>27</sup>. Algunos ejemplos básicos de elementos estructurantes que generalmente se utilizan son:

**Figura 12.** Ejemplos de formas básicas de elementos estructurantes básicos



*Fuente:* Autor del proyecto

**Figura 13.** Elemento estructurante plano de tamaño 3x3



*Fuente:* Autor del proyecto

El origen del elemento se sitúa en el centro. De modo que la morfología ha llegado a ser una herramienta importante para la extracción de información, sus técnicas también han ido avanzado con el desarrollo de la tecnología con lo que se hace más fácil interpretar y entender imágenes tanto en blanco y negro, escala de grises y las de colores. Existen operadores morfológicos básicos que permiten las imágenes sean modificadas de la forma necesaria para hacer más fácil su interpretación y clasificación. Los operadores básicos para este proceso son la dilatación, erosión y sus combinaciones como apertura y cierre.

En Matlab las funciones utilizadas para realizar estas dos operaciones morfológicas son *erode* y *dilate*. El formato de ambas funciones es:

```
ImageR=erode(ImageS,w);  
ImageR=dilate(ImageS,w);
```

---

<sup>27</sup> TOSINA, María José. Diseño de un filtro morfológico para la reducción de fluorescencia en espectros Raman. Barcelona, 2008, 109 p. Tesis de grado. Universidad Politécnica de Catalunya. Departamento de Teoría de la Señal y Comunicaciones. Disponible en internet: [http://upcommons.upc.edu/pfc/bitstream/2099.1/6588/1/PFC\\_MARI\\_COMPLETO\\_definitivo.pdf](http://upcommons.upc.edu/pfc/bitstream/2099.1/6588/1/PFC_MARI_COMPLETO_definitivo.pdf).

TOSELLI, Alejandro Héctor. Reconocimiento de Texto Manuscrito Continuo. Valencia, 20 de marzo de 2004, 230 p. Tesis Doctoral. Universidad Politécnica de Valencia. Departamento de Sistemas Informáticos y Computación.

Donde ImageR es la variable que recibe a la imagen resultado, ImageS es la imagen binaria origen a la que se desea aplicar la operación morfológica y w es una matriz de unos y ceros que determina el formato y estructura de la rejilla.

La dilatación y la erosión son operadores morfológicos, también conocidos como operadores locales, ya que el resultado de aplicar un operador a un punto, depende del propio punto de la imagen y de los demás puntos que se encuentran alrededor estos, los determina el elemento estructurante con el que realizamos la operación. A continuación se explicarán cada uno de ellos para entender el funcionamiento y conceptos de esta etapa:

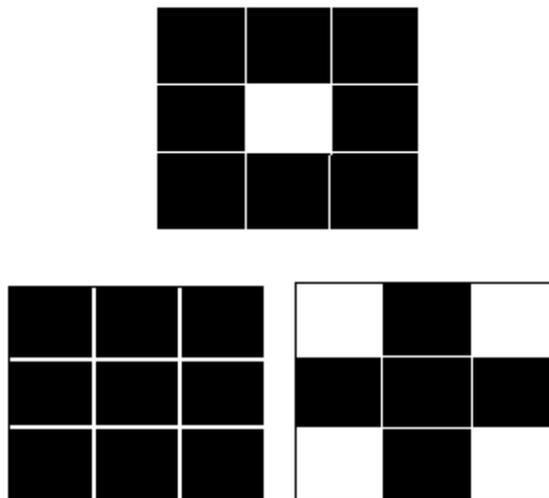
➤ **DILATACIÓN**

Cada uno de los operadores morfológicos presentan modelos matemáticos que permiten analizar la modificación de cada punto por medio de la iteración de ecuaciones. Para la dilatación se puede considerar el siguiente modelo básico:

$$G(j,k)=\text{máx}[F(j,k),F(j,k+1),F(j,k-1),\dots]$$

La operación matemática de la dilatación, se tiene en cuenta dos operandos, uno de ellos la propia imagen y por otro una submatriz de puntos de la imagen, llamada elemento estructurante. La ubicación de un punto en la imagen modificada quedaría definida por las coordenadas j y k. cada píxel reestructura será modificado por un nuevo nivel en la escala de grises o en el caso progresivo después de la etapa de binarización un píxel puede ser modificado por un nivel de negro o blanco absoluto.

**Figura 14.** Matriz original y submatrices



$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Fuente: Autor del proyecto

Al realizar la operación de dilatación,  $G(j,k)$ , asigna al antiguo  $F(j,k)$  el valor numérico de gris mayor existente en dicho grupo. Debido a esto el píxel o punto de la imagen original, que tenga una valor de gris bajo la imagen modificada estará reemplazada en esa misma coordenada por un nivel numérico blanco (255) y viceversa, el píxel de la imagen original que tenga un valor numérico de gris alto dará como resultada una imagen que reemplazara ese píxel con un nivel numérico de negro (0). Esto daría como resultado en la imagen de salida que las áreas oscuras de pequeño tamaño tiendan a desaparecer y las áreas claras las suavizaría aun más.

Esta operación se podría interpretar de cierta forma como la convolución de dos funciones que en este caso serían la imagen original y la submatriz (los elementos estructuradores). La submatriz trabajaría como función positiva que al convolucionarse con cada punto de la imagen original, aumentaría su nivel y suavizará las zonas de alto nivel de variación de pendiente.

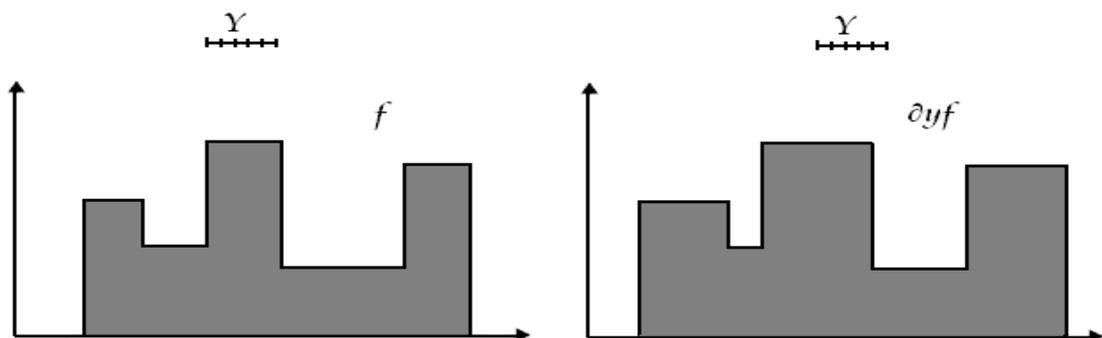
En la aplicación a una imagen lo que se consigue es que sea más clara y en la que las zonas oscuras de pequeño tamaño desaparezcan. Ejemplo de imagen binaria:

**Figura 15.** Dilatación de una imagen binaria mediante un elemento estructurante de tamaño 3x3



Fuente: Autor del proyecto

**Figura 16.** Dilatación de una señal unidimensional definida por la función  $f$ . Elemento estructurante de tamaño 3. El resultado es el máximo de todos los valores presentes bajo la definición del elemento estructurante.



Fuente: Fundamentos de Morfología Matemática. Biblioteca virtual Miguel de Cervantes. Disponible en internet:[http://descargas.cervantesvirtual.com/servlet/SirveObras/57915842105571617400080/008591\\_3.pdf](http://descargas.cervantesvirtual.com/servlet/SirveObras/57915842105571617400080/008591_3.pdf)

El resultado de la dilatación en señales bidimensionales de escala grises (imágenes) es una señal de mayor valor, es decir, una imagen más clara, puesto que la dilatación maximiza el valor de la señal. El efecto visual es tal que los objetos claros aumentan su definición frente a los oscuros.

➤ **EROSIÓN**

La erosión es el operador que causa el efecto contrario al de la dilatación y está representada en términos generales por la siguiente ecuación:

$$G(j,k)=\min[F(j,k),F(j,k+1),F(j,k-1),\dots]$$

Todo lo explicado en la dilatación ahora cobra el sentido contrario. De la submatriz que contiene a  $F(j,k)$ , al aplicar la erosión, se coge el valor del mínimo valor de gris existente en un punto validado, y se le asigna. Como ha quedado claro, es la operación contraria.

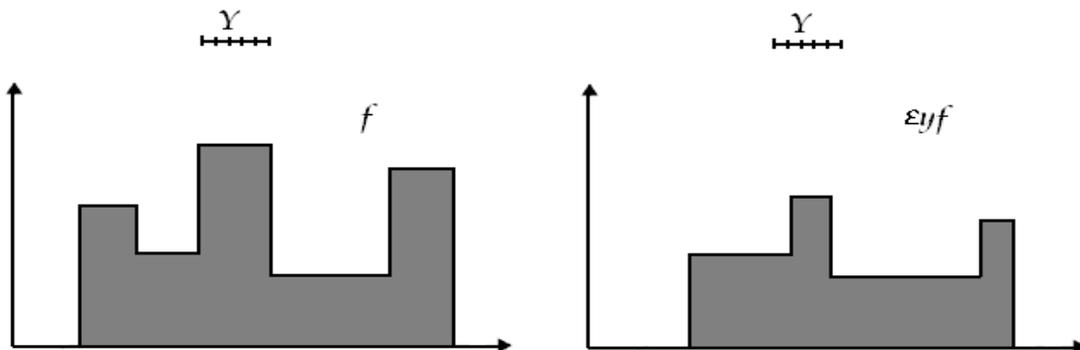
El resultado en una imagen es un resultado más oscuro, donde los detalles claros han desaparecido y se han unido zonas oscuras antes separadas por una pequeña zona más clara.

**Figura 17.** Erosión de una imagen binaria mediante un elemento estructurante de tamaño 3x3.



*Fuente:* Autor del proyecto

**Figura 18.** Erosión de una señal unidimensional definida por la función  $f$ . Elemento estructurante de tamaño 3. El resultado es el mínimo de todos los valores presentes bajo la definición del elemento estructurante.



*Fuente:* Fundamentos de Morfología Matemática. Biblioteca virtual Miguel de Cervantes. Disponible en internet:[http://descargas.cervantesvirtual.com/servlet/SirveObras/57915842105571617400080/008591\\_3.pdf](http://descargas.cervantesvirtual.com/servlet/SirveObras/57915842105571617400080/008591_3.pdf)

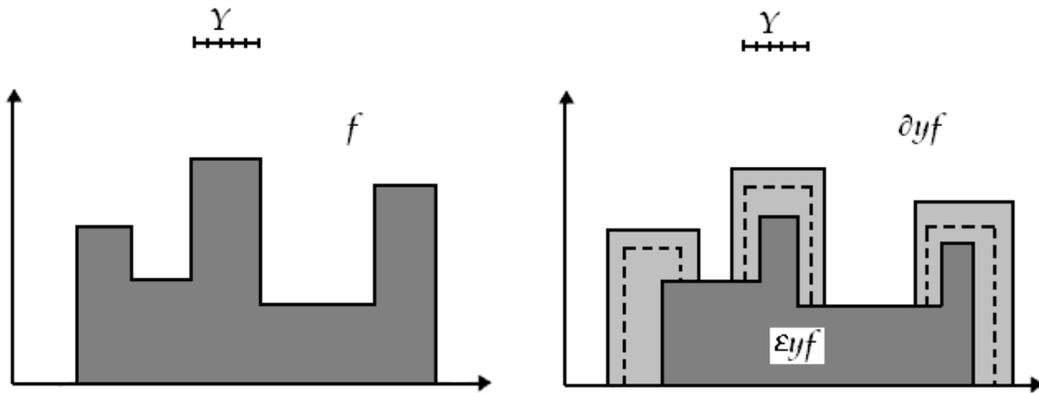
### ➤ APERTURA Y CIERRE

Son las combinaciones de erosión/dilatación, dilatación/erosión respectivamente. El objetivo de la apertura y del cierre es simplificar la imagen, es decir, suavizan la señal.

La apertura sirve para eliminar pequeños detalles claros, pues lo que hace es erosionar para eliminar detalles pequeños oscuros, y más tarde dilatar para recuperar, ya sin esos detalles, el nivel de gris del principio.

El cierre es la operación o conjunto de operaciones, contraria. La aplicación de la dilatación al principio, elimina los detalles de poca importancia de un nivel más claro, y la aplicación posterior de la erosión restablece el nivel del principio de gris general<sup>28</sup>.

**Figura 19.** Dilatación y erosión de una señal unidimensional definida por la función  $f$ . elemento estructurante de tamaño 3.



*Fuente:* Fundamentos de Morfología Matemática. Biblioteca virtual Miguel de Cervantes. Disponible en internet:[http://descargas.cervantesvirtual.com/servlet/SirveObras/57915842105571617400080/008591\\_3.pdf](http://descargas.cervantesvirtual.com/servlet/SirveObras/57915842105571617400080/008591_3.pdf)

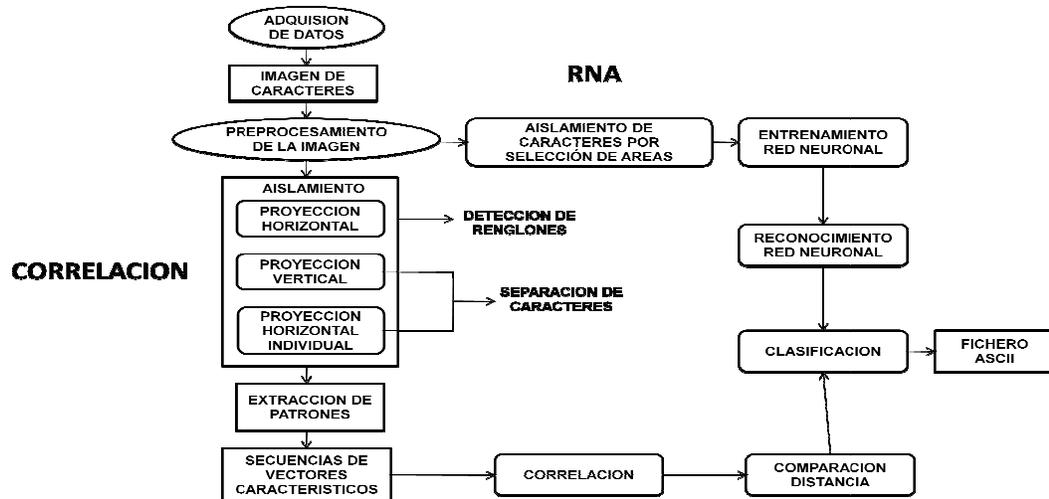
<sup>28</sup>Disponible en internet: [http://www.ieev.uma.es/tdi/www\\_netscape/TEMAS/Tdi\\_10/index.php#1.-¿Qué es la morfología y adaptado por el autor de proyecto. \(Consultado: Mayo 27 de 2009\)](http://www.ieev.uma.es/tdi/www_netscape/TEMAS/Tdi_10/index.php#1.-¿Qué es la morfología y adaptado por el autor de proyecto. (Consultado: Mayo 27 de 2009))

CONSTRUCCIÓN DE UN SISTEMA OCR

En la figura 20 se puede observar las etapas del desarrollo implementando en los dos métodos de clasificación propuestos en el proyecto para realizar la construcción del sistema OCR. Se puede observar el proceso de reconocimiento desde la adquisición de datos, que es cuando se adquiere la imagen mediante algún dispositivo electrónico como una cámara, un escáner o mediante el uso del lápiz óptico con la ayuda de un software en donde se puedan crear o editar imágenes. Una vez adquirida la imagen se continúa con el pre procesamiento de la misma haciendo uso de las técnicas y herramientas descritas en capítulos anteriores de tal forma que la información a procesar sea lo más limpia posible de ruido para que no hallan elementos que alteren la estructura básica del carácter o del área en donde se desea extraer aquellos patrones en la imagen.

Para el caso del método de correlación se crea un algoritmo capaz de aislar los caracteres mediante la detección de renglones y la separación o enumeración de los mismos para su posterior análisis en la extracción de patrones, que permitan determinar por medio de una relación de distancia o grado de relación entre cada uno de los caracteres, cual es el dato de salida más semejante al dato de entrada. Por otro lado, se puede también visualizar las etapas en la clasificación de caracteres por medio del método de redes neuronales, en donde se pudo determinar la red más indicada a trabajar era el tipo de red neuronal de base radial (se profundizará este tipo de red en el último capítulo de este documento) para que mediante un entrenamiento adecuado de la red ésta sea capaz de generar como salida, el código ASCII del carácter cuya imagen se presenta como entrada.

Figura 20. Diagrama del reconocedor de caracteres ópticos (algoritmo del OCR)



Fuente: Autor del proyecto

La construcción de un sistema de reconocimiento de caracteres ópticos no es posible a menos de que se realice el proceso de segmentación de la manera adecuada, siendo este proceso el encargado de eliminar información que no interesa, permitir la extracción de caracteres según las cualidades de los mismos y permitir en última instancia, la toma de decisión en la etapa en que se debe etiquetar el carácter según el reconocimiento dado por el sistema.

### 3.1 SEGMENTACIÓN

En la etapa de segmentación se verifica la conectividad de las formas, se etiqueta y se separan. La segmentación permite al reconocedor extraer los patrones de cada carácter. Al utilizar matlab para la segmentación se utilizarían funciones como `bwlabel`.

En un sistema “off-line”, dependiendo de si el proceso de reconocimiento/interpretación de la imagen requiere o no de una identificación previa de sus elementos (caracteres), encontramos técnicas:

- Con la segmentación se pueden realizar procesos que reconocen palabras/frases por sus letras/palabras constituyentes. Es decir, se busca en primer lugar localizar y reconocer cada letra/palabra en orden a reconocer luego la palabra/frase entera.
- Con procesos libres de segmentación se hace el reconocimiento global de la palabra/frase sin analizar explícitamente sus letras/palabras componentes.

La aproximación basada en la segmentación tiene la ventaja de no requerir de diccionarios de palabras, ni tampoco de estructuras gramaticales. Sin embargo aparecen algunos inconvenientes, como por ejemplo combinaciones no válidas de letras o palabras, confusión en la distinción de espacios entre letras con respecto del entre palabras, etc.

En el sistema OCR desarrollado se utilizó un tipo de procesamiento de la imagen con segmentación que facilitará el análisis de cada carácter contenido en la imagen de tal forma que se pudieran aislar, clasificar y etiquetar uno a uno los objetos considerados caracteres en el contexto general de la imagen.

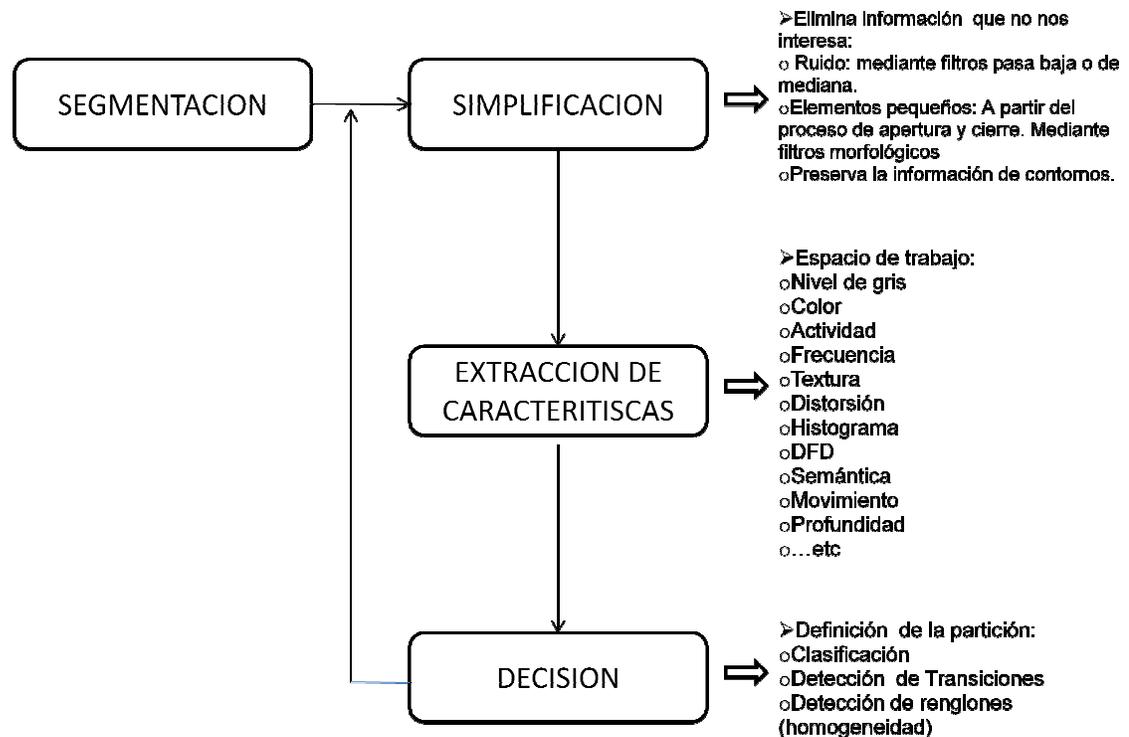
Considerando el otro tipo de segmentación, se podría denotar que los sistemas con reconocimiento global normalmente llevan a sistemas de mayor complejidad, que radica generalmente en el manejo de un léxico de tamaño mayor que implica la necesidad de disponer y manejar más modelos para las palabras del vocabulario. Normalmente el desarrollo de estos sistemas se halla supeditado a aplicaciones que conlleven la utilización de léxicos restringidos. Es conveniente aclarar que, si bien se puede hablar de entrenamiento y reconocimiento libres de segmentación. Esta propiedad puede estar implícitamente considerada en los modelos probabilísticos utilizados (entre ellos los HMMs). Estos modelos asumen cualquier posible segmentación de la imagen de texto manuscrito con una determinada probabilidad asociada. En otras palabras, la segmentación se encuentra integrada dentro de los procesos de entrenamiento y de reconocimiento<sup>29</sup>.

---

<sup>29</sup> TOSELLI, Op. Cit., p. 32

**3.1.1 Estrategias de segmentación.** La palabra 'segmentación' tiene un significado que depende del uso y del contexto en el cual se utilice. El objetivo básico de los algoritmos de segmentación es definir una partición del espacio. A continuación se revisa los pasos principales implicados en un algoritmo de segmentación y las principales opciones a realizar. Un esquema general para la segmentación se puede considerar como la concatenación de tres pasos principales, representados en la siguiente figura<sup>30</sup>:

**Figura 21.** Proceso global de la segmentación de imágenes



*Fuente:* HERVELLA AZOUZI, Samira. Editor de imágenes basado en regiones. Aplicación en entorno Matlab [online]. I.T.Telecomunicaciones,esp. Sonido e Imagen EUETIT.-Terrassa 2005-2006. [Consultado el 15 de mayo de 2009]. Disponible en internet: <http://upcommons.upc.edu/pfc/bitstream/2099.1/3860/1/54960-1.pdf>, y modificado por el autor del proyecto.

## 3.2 SIMPLIFICACIÓN

En algunas ocasiones, los datos originales en una imagen contienen información que es irrelevante para una aplicación dada. En tales casos, los datos se deben simplificar, removiendo toda aquella información irrelevante. La simplificación controla la cantidad y naturaleza de la información que es preservada. Además los datos simplificados deben contener áreas fáciles de segmentar. Por ejemplo, la simplificación puede reducir la complejidad de texturas, eliminando ruido o eliminar objetos muy pequeños para un

<sup>30</sup> HERVELLA AZOUZI, Op. Cit., p. 11

tamaño dado sin llegar a modificar la información de los contornos que es relevante para la aplicación<sup>31</sup>.

### 3.3 EXTRACCIÓN DE CARACTERÍSTICAS

La segmentación se realiza sobre características específicas de los datos. La selección del espacio de características conduce al tipo de homogeneidad que se espera en la partición final. Algunas características típicas en la etapa de extracción de características es la de discriminar patrones propias en el espacio de trabajo, tales como actividad, frecuencia, textura, distorsión, histograma, DFD, espacios que caracterizan algunas nociones semánticas, movimiento o profundidad. En algunos casos, la estimación de las características tiene que ser hecha en una región de soporte homogénea en términos del mismo criterio que la característica dada<sup>32</sup>.

**3.3.1 Extracción de caracteres y métodos de clasificación.** Dado un carácter de segmento aislado se pueden enumerar una serie de patrones útiles para el reconocimiento:

- Patrones basados en el momento:

Los momentos 2-D de los caracteres están dados por:

$$m_{pq} = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} X^p Y^q f(x, y)$$

De los momentos con los cuales se pueden computar los patrones de la imagen se tienen:

1. Masa total (número de píxeles en un carácter binario)
  2. Centroide: centro de masa
  3. Parámetros elípticos:
    - Excentricidad (radio de mayor a menor eje)
    - Orientación (ángulo de mayor eje)
  4. Asimetría
  5. Curtosis
  6. Momentos de orden superior
- Transformada de código de cadena y transformada de Hough
  - Transformadas y series de fourier
  - Redes neuronales y Correlación

---

<sup>31</sup> HERVELLA AZOUZI, Op. Cit., p. 12

<sup>32</sup> Ibíd., p.12

En este proyecto se realizaron las investigaciones pertinentes para los métodos de correlación y redes neuronales, los cuales fueron los temas a investigar y profundizar para la aplicación en el reconocimiento de caracteres ópticos. Más adelante, se plasmarán conceptos teóricos, matemáticos, metodologías y aplicaciones en estos temas de investigación.

### **3.4 DECISIÓN**

Finalmente para obtener una partición de los datos, el espacio de características debe ser analizado. El paso de decisión se tiene en cuenta sobre la posición de los contornos que forman la partición en el espacio de la imagen y los contornos separan áreas de datos que contienen elementos con las mismas características en el espacio de características seleccionado.

En la práctica, tres estrategias se pueden utilizar para definir la partición: la Clasificación, la Detección de Transiciones y la Detección de regiones (Homogeneidad)<sup>33</sup>. En este caso para los sistemas de reconocimiento realizados se utilizaron estrategias de clasificación que contiene algoritmos capaces de distinguir mediante patrones similares cada uno de los caracteres.

---

<sup>33</sup> *Ibíd.*, p.12

#### 4.1 CORRELACIÓN

La correlación es uno de los métodos utilizados en el reconocimiento de imágenes y es usado en este proyecto. Como tal, esta metodología está definida en términos de probabilidad y estadística, como el indicador de la fuerza y la dirección de una relación lineal entre dos variables aleatorias. Se considera que dos variables cuantitativas están correlacionadas cuando los valores de una de ellas varían sistemáticamente con respecto a los valores homónimos de la otra: si tenemos dos variables (A y B) existe correlación si al aumentar los valores de A lo hacen también los de B y viceversa.

La relación entre estas dos variables cuantitativas (A y B) queda representada mediante la línea de mejor ajuste, trazada a partir de la nube de puntos. Los principales componentes elementales de una línea de ajuste y, por lo tanto, de una correlación, son la fuerza, el sentido y la forma:

- La fuerza extrema según el caso, mide el grado en que la línea representa a la nube de puntos: si la nube es estrecha y alargada, se representa por una línea recta, lo que indica que la relación es fuerte; si la nube de puntos tiene una tendencia elíptica o circular, la relación es débil.
- El sentido mide la variación de los valores de B con respecto a A: si al crecer los valores de A lo hacen los de B, la relación es positiva; si al crecer los valores de A disminuyen los de B, la relación es negativa.
- La forma establece el tipo de línea que define el mejor ajuste: la línea recta, la curva monotónica o la curva no monotónica.

La relación de las variables puede calcularse mediante diversos coeficientes que miden el grado de correlación. El más conocido es el coeficiente de correlación de Pearson, que se obtiene dividiendo la covarianza de dos variables por el producto de sus desviaciones estándar<sup>34</sup>:

Dependiendo de las características de los datos a correlacionar, el coeficiente de correlación que debe aplicarse difiere. Las características principales de cada uno de ellos son las siguientes:

- Coeficiente de correlación de Pearson: Aplica cuando la métrica de las variables correlacionadas es como mínimo de intervalo.

---

<sup>34</sup> Disponible en internet: <http://es.wikipedia.org/wiki/Correlación>. (Consultado: Junio 10 de 2009)

- Coeficiente de correlación de Spearman y coeficiente de correlación t de Kendall: Son aplicables cuando la métrica de las variables es ordinal. El coeficiente de Spearman es especialmente indicado en aquellos casos en que se dé una violación del supuesto de normalidad y, en aquellos casos, en que aunque la métrica de las variables no sea de intervalo o de razón, podemos suponer que la variable con la que trabajamos presenta dicha métrica
- Coeficiente de correlación parcial: Es aplicable cuando se pretende estudiar la relación entre dos variables eliminado el efecto de una tercera variable.

Los procedimientos de regresión, por su parte nos permitirán modelizar la relación existente entre uno (regresión lineal simple) o más predictores (regresión lineal múltiple) con una variable criterio.

Cuando se realiza la comparación entre dos imágenes en Matlab, se ejecutará la acción mediante el comando “corr2” que establecerá un grado de relación entre los datos mediante el análisis de los coeficientes de Pearson. Para su mayor entendimiento, los coeficientes de Pearson se explicarán de forma geométrica a continuación:

El objetivo de esta metodología es cuantificar la intensidad de la relación lineal entre dos variables. El parámetro que da tal cuantificación es el coeficiente de correlación lineal de Pearson  $r$ , cuyo valor oscila entre  $-1$  y  $+1$ :

$$-1 \leq r = \frac{cov(X * Y)}{S_X * S_Y} = \frac{\sum_{i=1}^N (x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} * \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}} \leq +1$$

Los valores para este coeficiente  $r$  están comprendidos entre  $-1$  y  $1$ . Se tiene los siguientes criterios para  $r$ :

- $r = 1$  la correlación n lineal es perfecta, directa o correlación n lineal positiva
- $r = 0$  no existe correlación n lineal o correlación n lineal nula
- $r = -1$  la correlación n lineal es perfecta, inversa o correlación n negativa

Entre más se aproxima a los valores  $1$  y  $-1$  la aproximación a una correlación se considera buena. Cuando más se aleja de  $1$  o de  $-1$  y se acerca a cero se tiene menos confianza en la dependencia lineal por lo que una aproximación lineal será lo menos apropiado, sin embargo no significa que no existe dependencia, lo único que se puede decir es que la dependencia no es lineal. Un valor positivo para  $r$  indica que a medida que una variable crece la otra también lo hace, por el contrario si su valor es negativo, lo que se puede decir es que a medida que una variable crece la otra decrece.

En general, este tipo de clasificación resulta ser de mucha utilidad aplicada al tema de la comparación de caracteres o imágenes, ya que se podría determinar si existe o no alguna relación entre el carácter a reconocer y los caracteres predeterminados, que son aquellos con los que la imagen de entrada va a realizar la comparación. Esta comparación se realiza pixel a pixel para etiquetar mediante el máximo grado de relación el carácter que corresponda con los datos procesados. Esta relación plasmada en un plano generaría rectas lineales, parabólicas o exponenciales que partirían el espacio de ubicación de cada

uno de los caracteres creando nubes de separación que distinguirían los caracteres uno del otro y que a su vez se distinguirán según el tipo de relación que tengan. Debido a la naturaleza de la forma en cómo son separados los patrones de los caracteres, se estaría hablando de las ubicaciones de los mismos en un plano bidimensional, diferente al que se plasma en la separación de patrones en la clasificación mediante el uso de redes neuronales de base radial, ya que el conjunto de patrones son separados mediante funciones que serán analizadas en un plano euclidiano de n dimensiones.

**4.1.1 Correlación canónica.** Para corroborar la explicación teórica de correlación, también se hará su respectiva interpretación geométrica:

Ambas series de valores  $X(x_1, \dots, x_n)$  e  $Y(y_1, \dots, y_n)$  pueden estar consideradas como vectores en un espacio a n dimensiones. Si se reemplazan los vectores centrados:

$$X(x_1 - \bar{x}, \dots, x_n - \bar{x}) \text{ e } Y(y_1 - \bar{y}, \dots, y_n - \bar{y})$$

El coseno del ángulo alfa entre estos vectores es dado por la fórmula siguiente:

$$\cos = \frac{\sum_{i=1}^N (x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} * \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}}$$

Pues Cos ( $\alpha$ ) es el coeficiente de correlación de Pearson.

El coeficiente de correlación es el coseno entre ambos vectores centrados

Si  $r = 1$ , el ángulo  $\alpha = 0^\circ$ , ambos vectores son colineales (paralelos).

Si  $r = 0$ , el ángulo  $\alpha = 90^\circ$ , ambos vectores son ortogonales.

Si  $r = -1$ , el ángulo  $\alpha = 180^\circ$ , ambos vectores son colineales de dirección opuesto.

Más generalmente:  $\alpha = \arccos(r)$ .

Por supuesto, del punto vista geométrica, no se habla de correlación lineal: el coeficiente de correlación tiene siempre un sentido, cualquiera que sea su valor entre -1 y 1. Informa de modo preciso, no tanto sobre el grado de dependencia entre las variables, que sobre su distancia angular en la hiperesfera a n dimensiones<sup>35</sup>.

**4.1.2 Correlación entre variables.** La correlación es una técnica usada para determinar la relación entre dos o más variables. Por medio de la correlación entre variables se puede determinar el grado similitud o relación entre variables. La correlación puede ser de al menos dos variables o de una variable dependiente y dos o más variables independientes, denominada correlación múltiple.

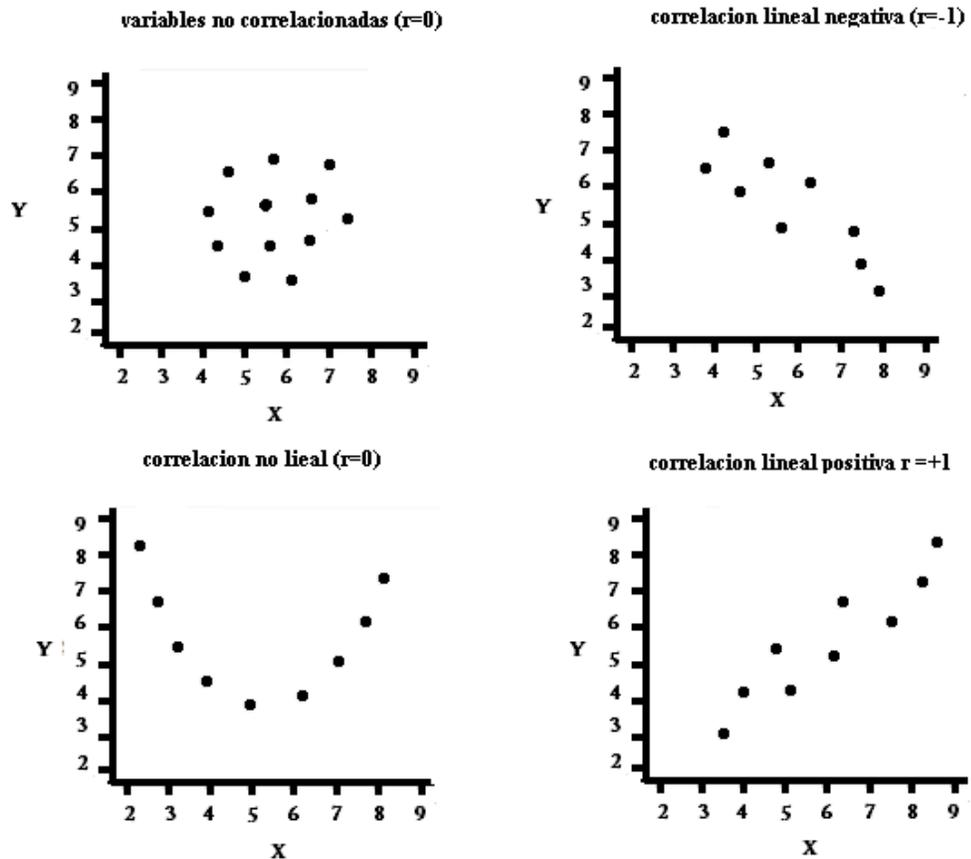
#### ➤ Coeficientes de correlación

El coeficiente de correlación es un valor cuantitativo de la relación entre dos o más variables. El coeficiente de correlación puede variar desde -1.00 hasta 1.00. La correlación de proporcionalidad directa o positiva se establece con los valores +1.00 y de

<sup>35</sup> Disponible en internet: <http://es.wikipedia.org/wiki/Correlaci3n>. (Consultado: Junio 10 de 2009)

proporcionalidad inversa o negativa, con  $-1.00$ . No existe relación entre las variables cuando el coeficiente es de  $0.00$ <sup>36</sup>.

**Figura 22.** Diagrama de dispersión de los coeficientes de correlación



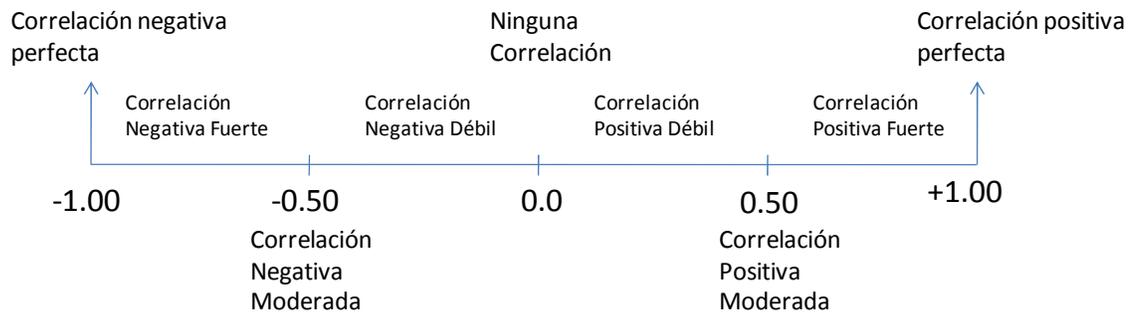
Fuente: Disponible en internet: <[www.uoc.edu/in3/emath/docs/RegresionLineal.pdf](http://www.uoc.edu/in3/emath/docs/RegresionLineal.pdf)>.

Como se puede observar en el diagrama de dispersiones anteriores, el valor de  $r$  se aproxima a  $+1$  cuando la correlación tiende a ser lineal directa (relación proporcional directa entre las variables  $X$  y  $Y$ ) y se aproxima a  $-1$  cuando la correlación tiende a ser lineal inversa. A continuación se muestra un diagrama que permite visualizar el tipo de correlación presente entre 2 variables según con el coeficiente  $r$  de que comparten<sup>37</sup>:

<sup>36</sup> Disponible en internet: [http://viref.udea.edu.co/contenido/menu\\_alterno/apuntes/gusramon/seminario\\_invest\\_vi/05-correlacion-variables.pdf](http://viref.udea.edu.co/contenido/menu_alterno/apuntes/gusramon/seminario_invest_vi/05-correlacion-variables.pdf), consultado el 8 de junio de 2009.

<sup>37</sup> VILA, Alicia., SEDANO, Maximo., LOPEZ, Ana, JUAN, Ángel A. Proyecto e-math financiado por la secretaria de estado de educación y universidades (MECD). Correlacion lineal y análisis de regresion [online]. [Consultado el 08 de abril de 2009]. Disponible en internet: <<http://www.uoc.edu/in3/emath/docs/RegresionLineal.pdf>>.

**Figura 23.** Coeficiente de correlación entre dos variables



Fuente: Disponible en internet: <[www.uoc.edu/in3/emath/docs/RegresionLineal.pdf](http://www.uoc.edu/in3/emath/docs/RegresionLineal.pdf)>.

**4.1.3 Correlación en Matlab.** En el lenguaje de Matlab se encuentra que la correlación es:

- corr2: Coeficiente de correlación 2-D
- Sintaxis:  $r = \text{corr2}(A,B)$
- Descripción:  $r = \text{corr2}(A, B)$  computa el coeficiente de correlación entre A y B, donde A y B son matrices o vectores de el mismo tamaño.
- Soporte de Clase: A y B pueden ser numéricos o lógicos. El valor devuelto en r es un doble escalar.
- Algoritmo: corr2 computa el coeficiente de correlación usando:

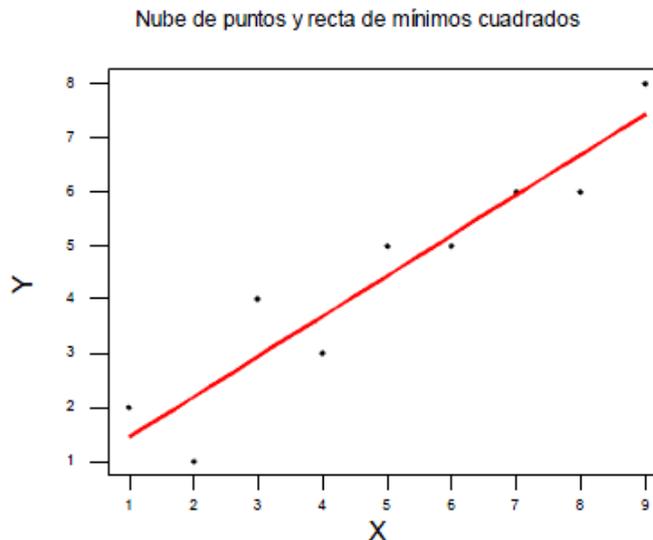
$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}}$$

Donde  $\bar{A} = \text{mean2}(A)$  y  $\bar{B} = \text{mean2}(B)$

**4.1.4 Regresión lineal.** En aquellos casos en que el coeficiente de regresión lineal sea “cercano” a +1 o a -1, tiene sentido considerar la ecuación de la recta que “mejor se ajuste” a la nube de puntos (recta de mínimos cuadrados). Uno de los principales usos de dicha recta será el de predecir o estimar los valores de Y que se obtendrían para distintos valores de X.

Es importante tener en cuenta que como la relación entre dos variables no suele ser perfecta, la recta solo dará una predicción aproximada, y por tanto, llevará siempre asociado un término que represente el error de predicción que se comete con ella.

**Figura 24.** Nubes de puntos y recta de mínimos cuadrados



Fuente: Disponible en internet: <[www.uoc.edu/in3/emath/docs/RegresionLineal.pdf](http://www.uoc.edu/in3/emath/docs/RegresionLineal.pdf)>.

La ecuación de la recta de mínimos cuadrados (en forma punto-pendiente) es la siguiente<sup>38</sup>:

$$y - \bar{y} = \frac{Cov(X, Y)}{S_x^2} (x - \bar{x})$$

## 4.2 DESARROLLO

Para realizar el proceso de segmentación se redimensiona la imagen adquirida para disminuir el tiempo de procesamiento sin afectar el resultado de clasificación del programa OCR, lo cual se realiza con la función “*imresize*”. Posteriormente se realiza el tratamiento de la imagen adquirida realizando el proceso de binarización y eliminando elementos que provocan ruido, no relevantes para la adecuada extracción de patrones. Para el proceso de binarización se utiliza la función *rgb2gray* a la variable donde está almacenada la imagen redimensionada.

```
imagen=imresize(imagen,[1400 1400]);  
Imagen=rgb2gray(imagen)
```

Una vez redimensionada y tratada la imagen, se empieza con el proceso de separar líneas y etiquetar objetos. El algoritmo empieza a realizar un barrido de cada una de las filas de la imagen determinando por medio de un contador el número de filas que existen y el número de objetos presente en cada una de esas filas.

<sup>38</sup> VILA, Alicia., SEDANO, Máximo., LOPEZ, Ana, JUAN, Ángel A. Correlación lineal y Análisis de Regresión [online]. [Consultado el 08 de abril de 2009]. Disponible en internet: <[www.uoc.edu/in3/emath/docs/RegresionLineal.pdf](http://www.uoc.edu/in3/emath/docs/RegresionLineal.pdf)>.

Con el siguiente algoritmo se logra realizar el barrido de cada una de las filas y el recorte de cada uno de los objetos o letras.

Subrutina para la separación de líneas, “línea”

```
function [pl,re]=linea(aa);
fila=recorte(aa);
r=size(fila,1);
for s=1:r;
    if sum(fila(s,:))==0
        nm=fila(1:s-1,1:end);
        rm=fila(s:end,1:end);
        pl=~recorte(~nm);
        re=~recorte(~rm);
        break
    else
        pl=~fila;
        re=[];
    end
end
```

La función “islogical” devuelve una respuesta verdadera para una matriz lógica, por lo que si la respuesta es falsa, se debe convertir aquellos elementos cercanos al valor de 255 en RGG deban aproximarse al 0 lógico o en la situación contraria elementos que sean considerados más cercanos al valor 0 en RGB sean considerados como 1 lógico. Para ello se determina un umbral que se escogió a prueba y error, dando 0.5 una constante efectiva para realizar el proceso.

La función “im2bw” se utiliza para establecer el umbral en proceso de binarización de la imagen, en este caso como se puede observar es de 0.5. Este umbral se escogió de tal forma para que algunos colores RGB sean eliminados y considerados como valores lógicos 0 en la imagen binarizada.

Posteriormente a la imagen recortada se le realiza un filtro “medfilt2” para eliminar ruidos y elementos u objetos pequeños mediante la disposición de los operadores morfológicos “bwareaopen”. Para eliminar los objetos pequeños mediante los operadores morfológicos se establece un umbral que determina el valor del píxel del elemento. De esta forma si el píxel del elemento es menor que el umbral será eliminado convirtiéndose en un 0 lógico o si supera dicho umbral será convertido en 1 lógico.

Subrutina para el recorte de la imagen, “recorte”

En este algoritmo se recorta la imagen y se separan fila por fila. Para recortar la imagen se busca cada una de las coordenadas de los elementos, en este caso de las letras que conforman la imagen haciendo un barrido fila por fila encontrando las letras y permitiendo la extracción de patrones de cada imagen previamente tratada y recortada.

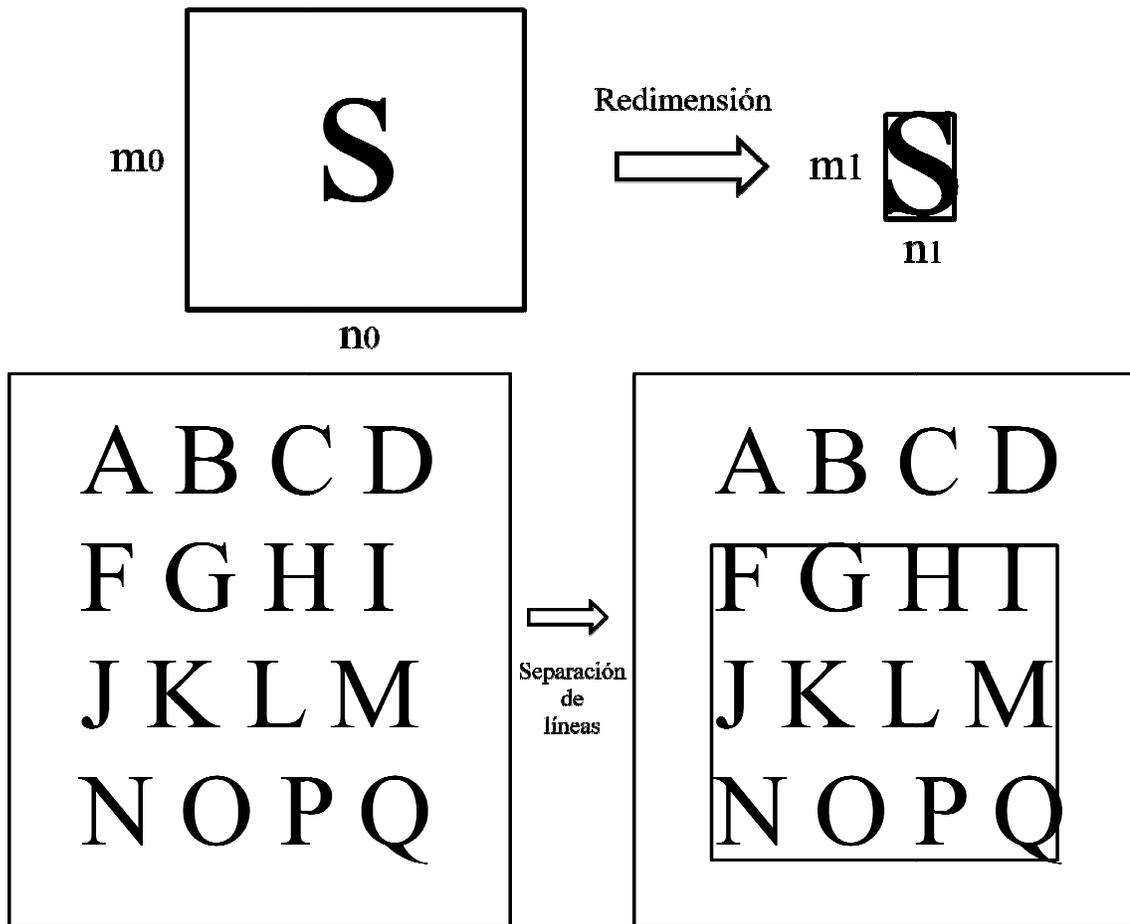
```
function [im,y_max,y_min,x_max,x_min]=recorte(imagen);
x=islogical(imagen);
```

```

if x==0;
    imagen=im2bw(imagen,0.5);
end
imagen=~imagen;
imagen=medfilt2(imagen,[8 8]);
imagen=bwareaopen(imagen,300);
[x y]= find(imagen);
y_max=max(y);
y_min=min(y);
x_max=max(x);
x_min=min(x);
im=imagen(x_min:x_max,y_min:y_max);

```

**Figura 25.** Redimensionamiento de la imagen y separación de líneas



Fuente: Autor del proyecto

Después de separar cada una de las filas que conforman la imagen, se procede a la extracción de caracteres o patrones propios de cada una de las letras. Para ello se etiquetó cada una de las letras mediante la función “bwlabel” y se separó cada una de las letras de modo que al detectarse espacios nulos o una matriz de ceros se volviese a etiquetar el siguiente elemento en donde se detectan trazos continuos. Para ello se utilizó la siguiente subrutina:

```
for c=1:objetos;
[fila,columna] = find(letra==c);
cord = [fila columna];
[sx sy]=size(cord);
n=zeros(x,y);
for i=1:sx;
x1=cord(i,1);
y1=cord(i,2);
n(x1,y1)=1;
end
```

Se realiza un ciclo que consta de un contador de columnas que termina hasta el número de letras u objetos detectados entregados por la función “bwlabel” y los introduce en una matriz mxn. Posteriormente se hallan los máximos y son almacenados en otra matriz. Finalmente se encuentran las posiciones en las que estaban los máximos para así determinar la letra al que corresponden y mediante otro ciclo de comparación se pueda mostrar la imagen clasificada y asignar la posición en la matriz del abecedario.

Ahora, se normaliza la imagen recortada para poder ser comparada mediante el método de correlación con las plantillas predeterminadas que poseen varios ejemplares de cada una de las letras a detectar. Las imágenes en las plantillas tienen un tamaño de 150x100 píxeles. De modo que todas las imágenes capturas en OCR van a ser posteriormente normalizadas a este tamaño.

```
N=recorte(n);
n=imresize(n,[150 100]);
imwrite(n,'eyytey.jpg');
```

Por último, se realiza el proceso de clasificación por medio del método de correlación entre dos matrices bidimensionales, el cual realiza el siguiente proceso matemático para así, de esta forma escoger aquella imagen que presente el mínimo error cuadrático.

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}}$$

Entonces, se realiza la correlación realizando este proceso matemático entre cada una de las imágenes adquiridas y cada una de las plantillas. Para eso se realiza la subrutina de comparación que se muestra a continuación:

```

for a=1:26;
    if a==1;
        im=imread('a0.jpg');
        comp(1,1)=corr2(im,n);
        im=imread('a1.jpg');
        comp(1,2)=corr2(im,n);
        im=imread('a2.jpg');
        comp(1,3)=corr2(im,n);
        im=imread('a3.jpg');
        comp=max(comp);
        comp;
    elseif a==2
        im=imread('b0.jpg');
        comp(1,1)=corr2(im,n);
        im=imread('b1.jpg');
        comp(1,2)=corr2(im,n);
        im=imread('b2.jpg');
        comp(1,3)=corr2(im,n);
        im=imread('b3.jpg');
        comp(1,4)=corr2(im,n);

```

Y así sucesivamente hasta la última letra. Después de terminada la rutina, se realiza la comparación con la plantilla de letras predeterminada:

```

word(1,a)=comp;
end
vd=find(word==max(word));
if vd==1
    letter='A';
elseif vd==2
    letter='B';

```

Se sigue haciendo una a una la comparación con cada una de las letras de las plantilla y por último escoger la letra que más se aproxime a la adquirida por el OCR.

### 4.3 REDES NEURONALES

Existe en el hombre un deseo profundo de poder reproducir la habilidad cognoscitiva por medios artificiales. La fascinación que la inteligencia como materia de estudio ha suscitado al género humano, puede verse reflejada en la aparición de una rama íntegra del estudio científico llamada " Inteligencia Artificial " a secas o también estudio de la inteligencia.

En el campo de la inteligencia artificial se refiere habitualmente de forma más sencilla como redes de neuronas o redes neuronales, las redes de neuronas artificiales (denominadas habitualmente como RNA o en inglés) son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida.

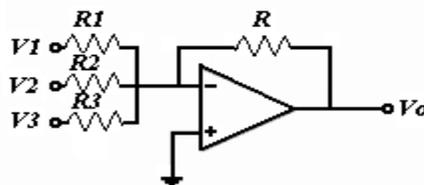
Una de las múltiples ramas por las cuales se ha desarrollado la investigación es el desarrollo de las llamadas "redes neuronales". Una red neuronal es el intento de poder realizar una simulación computacional del comportamiento de partes del cerebro humano mediante la réplica en pequeña escala de los patrones que éste desempeña para la formación de resultados a partir de los sucesos percibidos.

Concretamente, se trata de poder analizar y reproducir el mecanismo de aprendizaje y reconociendo de sucesos que poseen los animales más evolucionados. Una de las preguntas más interesantes que se plantean al hablar de este tema es la habilidad del cerebro para poder reconocer patrones. Se denomina reconocimiento de patrones a la capacidad de poder interpretar una imagen compleja (una foto, lo que ve el ojo) y actuar en consecuencia. Las computadoras digitales fueron diseñadas a partir de una lógica binaria (de 2 valores 0 - 1 o Verdadero - Falso), lo cual si bien facilitó su construcción, ha tenido como efecto una gran dificultad para procesar y reconocer imágenes, fotos, planos y dibujos<sup>39</sup>.

Las redes neuronales son estructuras pensadas para simular las estructura y el comportamiento del sistema nervioso, por esta razón una neurona artificial posee entradas, salidas y un estado además de estar conectadas a otras neuronas siendo las salidas de estas entradas de las demás<sup>40</sup>.

**4.3.1 Características de una red neuronal artificial.** Una neurona es una unidad procesadora de información que es fundamental para la operación de las redes neuronales. El modelo de una neurona artificial es una imitación del proceso de una neurona biológica, puede también asemejarse a un sumador hecho con un amplificador operacional tal como se ve en la figura<sup>41</sup>.

**Figura 26.** Red neuronal artificial



$$V_0 = \sum_{i=1}^n \frac{R}{R_i} V_i$$

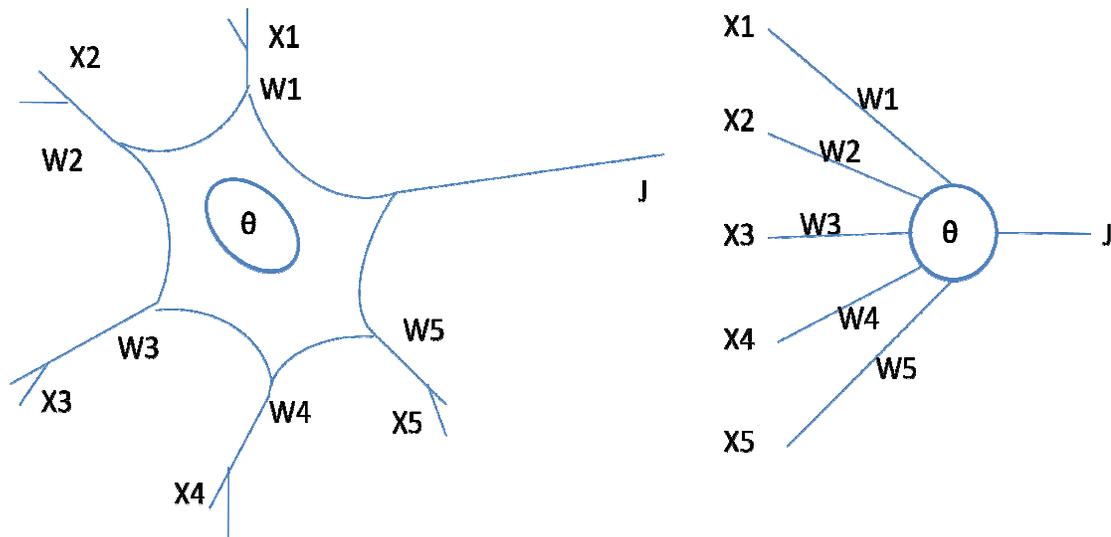
Fuente: <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf>

<sup>39</sup> RICOTE GIL, Fernando. Teoría de las Redes Neuronales para la Gerencia de Riesgos. Madrid. Universidad Complutense. Facultad de Ciencias Económicas y Empresariales. Disponible en la página de internet: <http://mfa2004.uclm.es/papers/ricote-fernando%20.pdf>, consultado el 8 de mayo de 2009.

<sup>40</sup> Disponible en internet: <http://www.udabol.uni.cc/jornada2007/conferencias/documentos/redesneuronales.pdf>, consultado el 8 de mayo de 2009.

<sup>41</sup> Disponible en internet: <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf>, consultado el 8 de mayo de 2009.

**Figura 27.** Comparación entre una red neuronal biológica y una red neuronal artificial



Fuente: <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf>

De la observación detallada del proceso biológico se han hallado los siguientes análogos con el sistema artificial

- Las entradas  $X_i$  representan las señales que provienen de otras neuronas y que son capturadas por las dendritas.
- Los pesos  $W_i$  son la intensidad de la  $W_i$  son valores reales.
- $\theta$  es la función umbral que la neurona debe sobrepasar para activarse; este proceso ocurre biológicamente en el cuerpo.

Las señales de entrada a una neurona artificial lugar de pulsos discretos, como se presentan en una neurona biológica. La entrada neta a cada unidad puede escribirse de la siguiente manera<sup>42</sup>:

**4.3.2 Funcionamiento de una RNA.** Una de las misiones en una red neuronal consiste en simular las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos recreados mediante mecanismos artificiales (como un circuito integrado, un ordenador o un conjunto de válvulas). El objetivo es conseguir que las

<sup>42</sup> Disponible en internet:

<<http://docs.google.com/gview?a=v&q=cache:aAtUNRYKyn8J:bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf.pdf+http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf.pdf&hl=es&gl=co>>. Consultado el día 18 de mayo de 2009

máquinas den respuestas similares a las que es capaz de dar el cerebro que se caracterizan por su generalización y su robustez. Una red neuronal se compone de unidades llamadas neuronas. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. Esta salida viene dada por tres funciones:

- Una función de propagación (también conocida como función de excitación), que por lo general consiste en el sumatorio de cada entrada multiplicada por el peso de su interconexión (valor neto). Si el peso es positivo, la conexión se denomina *excitatoria*; si es negativo, se denomina *inhibitoria*.
- Una función de activación, que modifica a la anterior. Puede no existir, siendo en este caso la salida la misma función de propagación.
- Una función de transferencia, que se aplica al valor devuelto por la función de activación. Se utiliza para acotar la salida de la neurona y generalmente viene dada por la interpretación que queramos darle a dichas salidas. Algunas de las más utilizadas son la función sigmoide (para obtener valores en el intervalo  $[0,1]$ ) y la tangente hiperbólica (para obtener valores en el intervalo  $[-1,1]$ )<sup>43</sup>.

**4.3.3 Diseño y programación de una RNA.** Con un paradigma convencional de programación en ingeniería del software, el objetivo del programador es modelar matemáticamente el problema en cuestión y posteriormente formular una solución mediante un algoritmo codificado que tenga una serie de propiedades que permitan resolver dicho problema. En contraposición, la aproximación basada en las RNA parte de un conjunto de datos de entrada suficientemente significativo y el objetivo es conseguir que la red aprenda automáticamente las propiedades deseadas. En este sentido, el diseño de la red tiene menos que ver con cuestiones como los flujos de datos y la detección de condiciones, y más que ver con cuestiones tales como la selección del modelo de red, la de las variables a incorporar y el pre procesamiento de la información que formará el conjunto de entrenamiento. Asimismo, el proceso por el que los parámetros de la red se adecuan a la resolución de cada problema no se denomina genéricamente programación sino que se suele denominar entrenamiento. Por ejemplo en una red que se va a aplicar al reconocimiento de caracteres ópticos, durante la fase de entrenamiento el sistema recibe imágenes de caracteres que conforman el abecedario. Si el entrenamiento es el adecuado, una vez concluido, el sistema podrá recibir caracteres impresos o manuscritos y obtener su clasificación con un buen grado de seguridad<sup>44</sup>.

**4.3.4 Estructura.** La mayoría de los científicos coinciden en que una RNA es muy diferente en términos de estructura de un cerebro animal. Al igual que el cerebro, una RNA se compone de un conjunto masivamente paralelo de unidades de proceso muy simples y es en las conexiones entre estas unidades donde reside la inteligencia de la red. Sin embargo, en términos de escala, un cerebro es muchísimo mayor que cualquier RNA creada hasta la actualidad, y las neuronas artificiales también son más simples que su contrapartida animal.

---

<sup>43</sup> FLORES ARMENTA, Luis Alberto. Red neuronal artificial. Junio de 2008. Área de Ingeniería en sistemas computacionales. Disponible en Internet: <http://www.itnogales.edu.mx/formatos/Red%20neuronal%20artificial.pdf>, consultado el día 18 de mayo de 2009.

<sup>44</sup> *Ibíd.*, p. 2.

Biológicamente, un cerebro aprende mediante la reorganización de las conexiones sinápticas entre las neuronas que lo componen. De la misma manera, las RNA tienen un gran número de *procesadores* virtuales interconectados que de forma simplificada simulan la funcionalidad de las neuronas biológicas. En esta simulación, la reorganización de las conexiones sinápticas biológicas se modela mediante un mecanismo de *pesos*, que son ajustados durante la fase de aprendizaje. En una RNA entrenada, el conjunto de los pesos determina el *conocimiento* de esa RNA y tiene la propiedad de resolver el problema para el que la RNA ha sido entrenada. Por otra parte, en una RNA, además de los pesos y las conexiones, cada neurona tiene asociada una función matemática denominada función de transferencia. Dicha función genera la señal de salida de la neurona a partir de las señales de entrada. La entrada de la función es la suma de todas las señales de entrada por el peso asociado a la conexión de entrada de la señal. Algunos ejemplos de entradas son la función escalón o Heaviside, la lineal o mixta, la sigmoide y la función gaussiana, recordando que la función de transferencia es la relación entre la señal de salida y la entrada.

**4.3.5 Ventajas.** Las redes neuronales artificiales (RNA) tienen muchas ventajas debido a que está basado en la estructura del sistema nervioso, principalmente el cerebro.

- *Aprendizaje:* Las RNA tienen la habilidad de aprender mediante una etapa que se llama *etapa de aprendizaje*. Ésta consiste en proporcionar a la RNA datos como entrada a su vez que se le indica cuál es la salida (respuesta) esperada.
- *Auto organización:* Una RNA crea su propia representación de la información en su interior, descargando al usuario de esto.
- *Tolerancia a fallos:* Debido a que una RNA almacena la información de forma redundante, ésta puede seguir respondiendo aceptablemente aún si se daña parcialmente.
- *Flexibilidad:* Una RNA puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada (ej. si la información de entrada es la imagen de un objeto, la respuesta correspondiente no sufre cambios si la imagen cambia un poco su brillo o el objeto cambia ligeramente)
- *Tiempo real:* La estructura de una RNA es paralela, por lo cual si esto es implementado con computadoras o en dispositivos electrónicos especiales, se pueden obtener respuestas en tiempo real.

**4.3.6 Tipología de las RNA.** Existe una serie de modelos que aparecen en la mayoría de estudios académicos y en bibliografía especializada:

- Perceptrón
- Adaline
- Perceptrón multicapa
- Memorias asociativas
- Máquina de Boltzmann

- Máquina de Cauchy
- Propagación hacia atrás (Backpropagation)
- Redes de Elman
- Redes de Hopfield
- Red de contrapropagación
- Redes de neuronas de base radial
- Redes de neuronas de aprendizaje competitivo
- Mapas Autoorganizados (RNA)
- Crecimiento dinámico de células
- Gas Neuronal Creciente
- Redes ART (*Adaptative Resonance Theory*).

**4.3.7 Topología.** Una primera clasificación de las redes de neuronas artificiales que se suele hacer es en función del patrón de conexiones que presenta. Así se definen tres tipos básicos de redes:

Dos tipos de redes de propagación hacia delante o acíclicas en las que todas las señales van desde la capa de entrada hacia la salida sin existir ciclos, ni conexiones entre neuronas de la misma capa.

1. *Monocapa*: Perceptrón, Adaline y Madaline.
2. *Multicapa*: Perceptrón multicapa.

Y el segundo tipo que son las redes recurrentes que presentan al menos un ciclo cerrado de activación neuronal. Ejemplos: Elman, Hopfield, máquina de Boltzmann<sup>45</sup>.

- **Redes Monocapa (1 Capa).** Esta red está conformada por una capa de señales de entrada y una capa de señales de salida. Es llamada red de una sola capa, ya que se toma en cuenta aquella donde se realiza alguna operación, la cual corresponde de la capa de salida. Se utiliza típicamente en tareas relacionadas con auto asociación.

- ***El Perceptrón***

Fue uno de los primeros modelos de redes neuronales con conexión hacia adelante, consiste de una sola neurona con pesos variables y un umbral, como función de transferencia se usa a la función limitador duro. Un Perceptrón imita una neurona tomando la suma ponderada de sus entradas y enviando a la salida un **1** si la suma es más grande que algún valor umbral ajustable o una salida **0** si ocurre lo contrario.

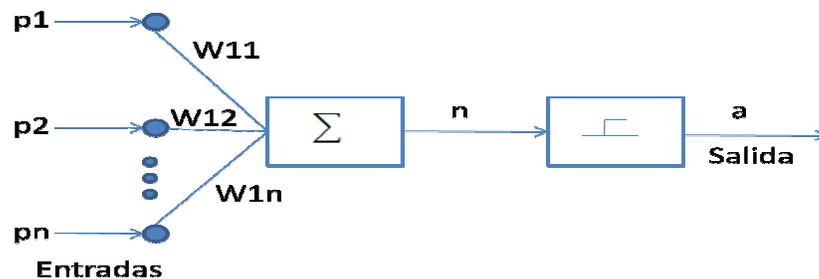
Las entradas ( $p_1, p_2, \dots, p_n$ ) y los pesos de conexión ( $w_1, w_2, \dots, w_n$ ) que se ilustrará en la figura son valores normalmente reales que pueden ser positivos o negativos. De alguna característica  $X_i$  tiende a causar la activación del Perceptrón, el

---

<sup>45</sup> FLORES ARMENTA, Luis Alberto. Red neuronal artificial. Junio de 2008. Área de Ingeniería en sistemas computacionales. Disponible en Internet: <http://www.itnogales.edu.mx/formatos/Red%20neuronal%20artificial.pdf>, consultado el día 18 de mayo de 2009.

peso  $W_i$  será positivo; si la característica  $X_i$  inhibe al Perceptrón, entonces el peso  $W_i$  será negativo<sup>46</sup>.

**Figura 28.** RNA (Perceptrón)

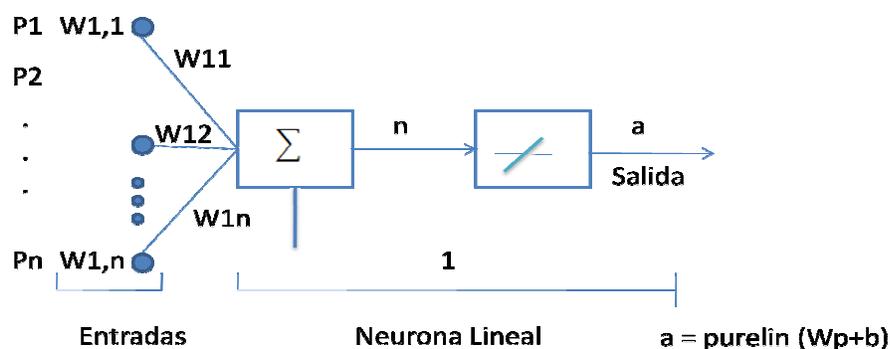


Fuente: <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf>

- **Redes Adaline y Madaline**

Las redes ADALINE (Adaptive Linear Neuron) ò (Adaptive Linear Element) y MADALINE (MULTIPLE ADALINE), Las arquitecturas de las redes Adaline y Madalin son similares al Perceptrón, excepto en su función de transferencia. La red Adenaline está limitada a una única neurona de salida, mientras que la Madaline puede tener varias capas. La diferencia fundamental respecto al Perceptrón se refiere al mecanismo de aprendizaje. Adaline y Madaline utilizan la denominada regla de Widrow Hoff o regla del mínimo error cuadrático medio (LMS)<sup>47</sup>.

**Figura 29.** Red Adaline



Fuente: <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf>

<sup>46</sup> Disponible en internet: <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf>, consultado el 8 de mayo de 2009.

<sup>47</sup> Disponible en internet: <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf>, consultado el 8 de mayo de 2009.

- **Redes Multicapa (n Capas).** Es aquella que dispone de conjuntos de neuronas agrupadas en varios niveles o capas como (2, 3,....., n). En este caso, una forma para distinguir la capa a la que pertenece una neurona, consistiría en fijarse en el origen de las señales que recibe la entrada y el destino de la señal de salida. Generalmente todas las neuronas de una capa reciben señales de entrada de otra capa anterior, más cercana a las entradas de la red y envían las señales de salida a una capa posterior más cercana a la salida de la red. A estas conexiones se les denomina conexiones hacia delante (feedforward).

Sin embargo en un gran número de estas redes existe también la posibilidad de conectar las salidas de neuronas posteriores a las entradas de las capas anteriores, a estas conexiones se les denomina conexiones hacia atrás (feedback).

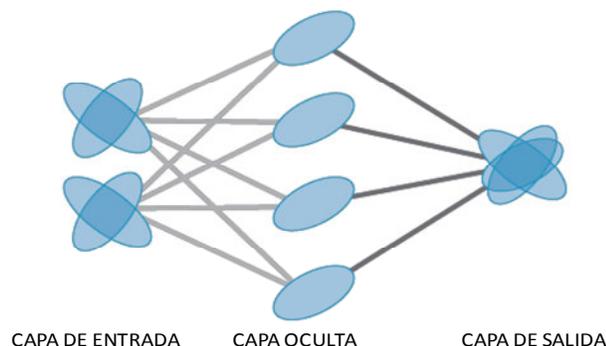
Un Perceptrón Multicapa tiene tres capas:

1. El modelo de cada neurona incluye una función de transferencia no lineal. La cual debe ser derivable en todo el rango. La función de transferencia más usada es la Sigmoidal, su uso tiene una motivación biológica por representar el estado refractario de una neurona real. Como funciones de transferencia pueden ser usadas también las funciones log-sigmoide (logsig), Tan-sigmoide (tansig) y la función lineal (purelin).
2. La red contiene varias capas ocultas que no corresponden a la capa de entrada ni a la capa de salida, esto posibilita el aprender tareas complejas que son extraídas regresivamente de las entradas.
3. La red presenta un alto grado de conectividad, la cual está determinada por la cantidad de pesos existentes.

- **Red de Retropropagación (Backpropagation)**

La red de retro propagación, conocida como Perceptrón multicapa (Multilayer Perceptrón), está formada por varias capas de neuronas. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, éste se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas<sup>48</sup>.

**Figura 30.** Red Backpropagation



Fuente: Autor del proyecto

<sup>48</sup> Disponible en internet:  
<http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf.pdf>, consultado el 8 de mayo de 2009.

En este tipo de redes, el problema a la hora de entrenarlas estriba en que sólo se conoce la salida de la red y la entrada, de forma que no se pueden ajustar los pesos sinápticos asociados a las neuronas de las capas ocultas, ya que no se puede inferir a partir del estado de la capa de salida como tiene que ser el estado de las capas ocultas.

El sistema de entrenamiento mediante Backpropagation consiste en:

- Empezar con unos pesos sinápticos cualquiera (generalmente elegidos al azar).
- Introducir unos datos de entrada (en la capa de entradas) elegidos al azar entre los datos de entrada que se van a usar para el entrenamiento.
- Dejar que la red genere un vector de datos de salida (propagación hacia delante).
- Comparar la salida generada por la red con la salida deseada.
- La diferencia obtenida entre la salida generada y la deseada (denominada error) se usa para ajustar los pesos sinápticos de las neuronas de la capa de salidas.
- El error se propaga hacia atrás (Backpropagation), hacia la capa de neuronas anterior, y se usa para ajustar los pesos sinápticos en esta capa.
- Se continua propagando el error hacia atrás y ajustando los pesos hasta que se alcance la capa de entradas.

Este proceso se repetirá con los diferentes datos de entrenamiento.

Es difícil encontrar el mejor conjunto de pesos sinápticos para la red de manera que la eficiencia sea máxima. Esto se debe a que la función que mide la eficiencia de las redes con capas ocultas, en la mayoría de los casos, es una función compleja que presenta muchos mínimos locales. Y el mejor conjunto de pesos sinápticos es aquel que hace que la función alcance un mínimo global.

**4.3.8 Tipos de entrada de una RNA.** Finalmente también se pueden clasificar las RNA según sean capaces de procesar información de distinto tipo en:

- Redes analógicas: procesan datos de entrada con valores continuos y, habitualmente, acotados. Ejemplos de este tipo de redes son: Hopfield, Kohonen y las redes de aprendizaje competitivo.
- Redes discretas: procesan datos de entrada de naturaleza discreta; habitualmente valores lógicos booleanos. Ejemplos de este segundo tipo de redes son: las máquinas de Bolzman y Cauchy, y la red discreta de Hopfield<sup>49</sup>.

**4.3.9 Funciones de transferencia.** Estas funciones permiten la transformación de una entrada procesada en un valor de salida. A continuación se especificarán las más utilizadas<sup>50</sup>:

---

<sup>49</sup> FLORES ARMENTA, Luis Alberto. Red neuronal artificial. Junio de 2008. Área de Ingeniería en sistemas computacionales. Disponible en Internet: <http://www.itnogales.edu.mx/formatos/Red%20neuronal%20artificial.pdf>, consultado el día 18 de mayo de 2009.

<sup>50</sup> Disponible en internet: <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf.pdf>, consultado el 8 de mayo de 2009.

➤ **Función escalón ò limitador fuerte (Hardlim)**

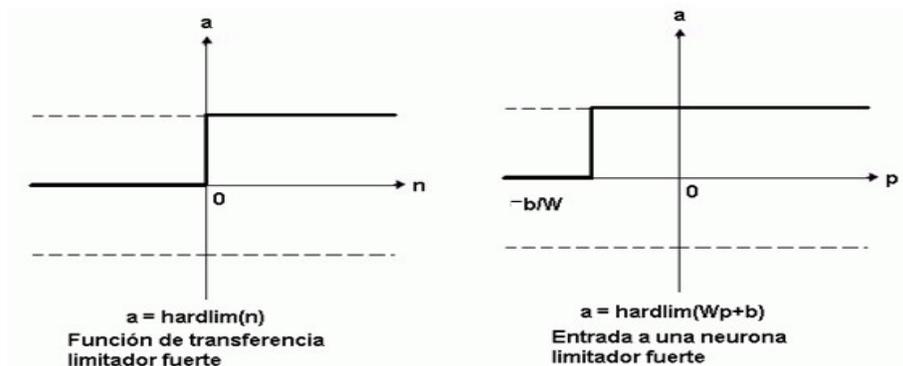
Para este tipo de función de transferencia se tiene:

$$a = f(n) = \begin{cases} 1 & \text{sin } n \geq 0 \\ 0 & \text{sin } n \leq 0 \end{cases}$$

Donde n es el campo local inducido de la neurona que es:

$$ak = \sum w_k n_j + b_k$$

**Figura 31.** Función escalón



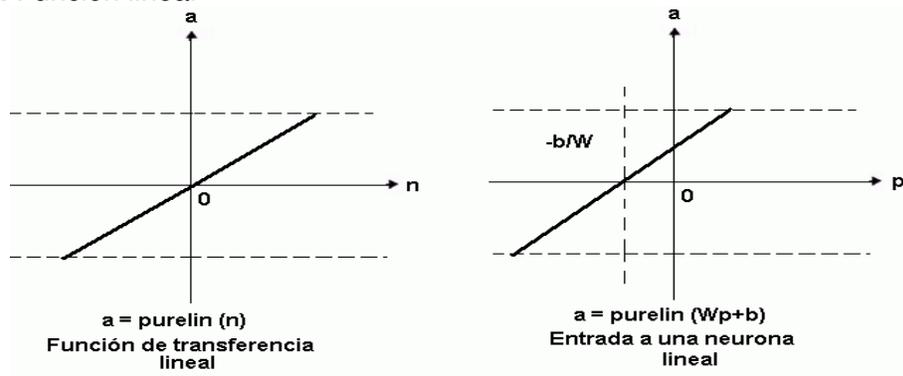
Fuente: <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf.pdf>

➤ **Función lineal (purelin)**

Para esta función se tiene:

$$a = f(n) = K \cdot n$$

**Figura 32.** Función lineal



Fuente: <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf.pdf>

Donde k es el factor de amplificación en la región lineal de operación.

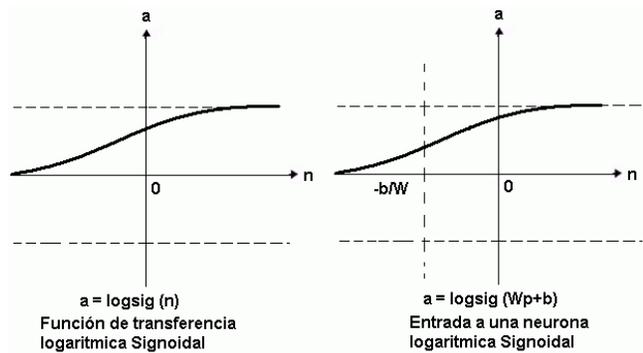
➤ **Funciones sigmoiales (logsig)**

Las funciones de transferencia sigmoiales son las funciones de construcción de redes neuronales artificiales, definida por:

$$a = f(n) = \frac{1}{(1+e^{-n})}$$

$$a = f(n) = \text{tansig}(n)$$

**Figura 33.** Funciones sigmoiales



Fuente: <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf>

**Tabla 4.** Funciones de transferencia

NOMBRE	RELACION ENTRADA/SALIDA	FUNCION DE TRANSFERENCIA
Limitador Fuerte	$a = 0 \ n < 0$ $a = 1 \ n \geq 0$	hardlim
Limitador Fuerte Simétrico	$a = -1 \ n < 0$ $a = +1 \ n \geq 0$	harlims
Lineal Positivo	$a = -1 \ n < 0$ $a = n \ 0 \leq n$	poslin
Lineal	$a = n$	Purelin
Lineal Saturado	$a = 0 \ n < 0$ $a = n \ 0 \leq n \leq 1$ $a = 1 \ n > 1$	satlin
Lineal Saturado Simétrico	$a = -1 \ n < -1$ $a = n \ -1 \leq n \leq +1$ $a = +1 \ n > 1$	Satlims
Sigmoial Logarítmico	$a = \frac{1}{1 + e^{-n}}$	Logsig

Tangente Sigmoidal Hiperbólica	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	tansig
Competitiva	a=1 Neurona con N Max. a=0 El resto de neuronas	compet

Fuente: <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf>

**4.3.10 Aprendizaje.** Una segunda clasificación que se suele hacer es en función del tipo de aprendizaje de que es capaz (si necesita o no un conjunto de entrenamiento supervisado). Para cada tipo de aprendizaje encontramos varios modelos propuestos por diferentes autores:

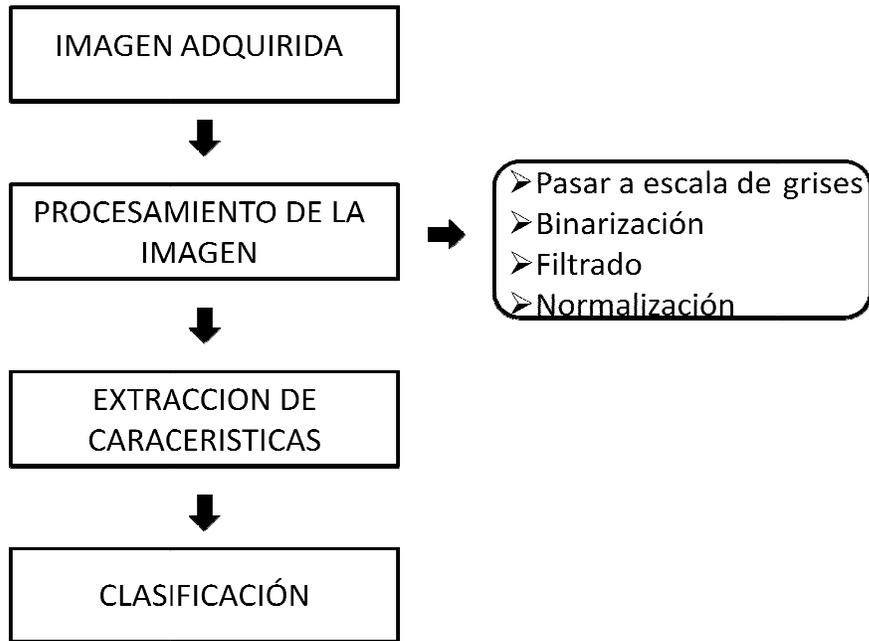
- Aprendizaje supervisado: necesitan un conjunto de datos de entrada previamente clasificado o cuya respuesta objetivo se conoce. Ejemplos de este tipo de redes son: el Perceptrón simple, la red Adaline, el Perceptrón multicapa y la memoria asociativa bidireccional.
- Aprendizaje no supervisado o auto organizado: no necesitan de tal conjunto previo. Ejemplos de este tipo de redes son: las memorias asociativas, las redes de Hopfield, la máquina de Bolzman y la máquina de Cauchy, las redes de aprendizaje competitivo, las redes de Kohonen o mapas auto organizados y las redes de resonancia adaptativa (ART)
- Redes híbridas: son un enfoque mixto en el que se utiliza una función de mejora para facilitar la convergencia. Un ejemplo de este último tipo son las redes de base radial.
- Aprendizaje reforzado: se sitúa a medio camino entre el supervisado y el auto organizado<sup>51</sup>.

#### 4.4 DESARROLLO

Para el desarrollo del programa mediante el método de clasificación mediante redes neuronales se tuvieron en cuenta las etapas del proceso que se muestran en la figura 34. El éxito de este método de clasificación está en escoger el tipo de red neuronal a utilizar y realizar un buen entrenamiento de la red escogido. Para este caso se escogió el tipo de redes neuronales de base radial, del cual se realizará un énfasis más adelante en este espacio de desarrollo del algoritmo.

<sup>51</sup> FLORES ARMENTA, Luis Alberto. Red neuronal artificial. Junio de 2008. Área de Ingeniería en sistemas computacionales. Disponible en Internet: <http://www.itnogales.edu.mx/formatos/Red%20neuronal%20artificial.pdf>, consultado el día 18 de mayo de 2009

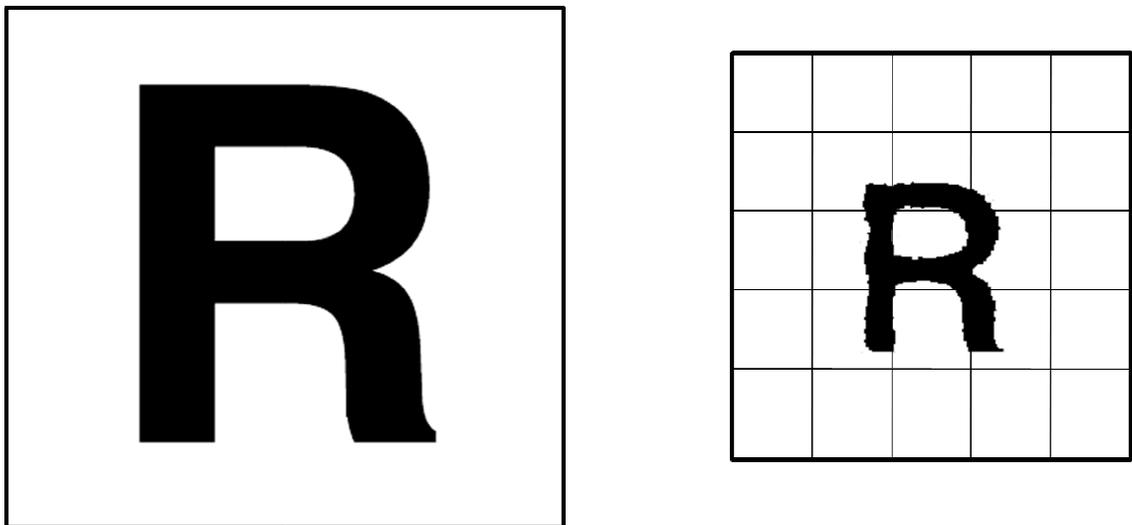
**Figura 34.** Etapas del proceso de desarrollo con el método de RNA



*Fuente:* Autor del proyecto

El proceso de adquisición de la imagen y en el procesamiento de la misma, no difiere en lo absoluto de la forma como se realizó para el método de clasificación por correlación, por lo que se omitirán especificaciones de estas etapas al respecto.

**Figura 35.** Imagen original de tamaño 500x379 píxeles e imagen redimensionada a un tamaño de 70x50 píxeles.



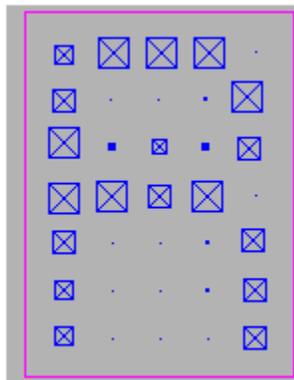
*Fuente:* Autor del proyecto

La extracción de las características de la imagen se basa en el siguiente algoritmo. Sea  $\mathbf{R}$  La matriz de dimensión 70x50 que representa a una imagen binaria.

$$\begin{aligned}
 & \text{Para } i=1:7 \\
 & \text{Para } j=1:5 \\
 & \vec{s} = \sum_{K=10*i-9}^{K=10*i} \sum_{W=10*j-9}^{W=10*j} A_{K,W} \\
 \xrightarrow{\text{Suma}} ((i-1)*5+j) &= \sum_{n=1}^{n=10} \vec{s}_n \\
 & \text{FIN} \\
 & \text{FIN} \\
 \xrightarrow{\text{Carac}} &= \left( \left( 100 - \xrightarrow{\text{Suma}} \right) / 100 \right) \\
 & \text{FIN}
 \end{aligned}$$

De este algoritmo se obtiene un vector de características  $\xrightarrow{\text{Carac}}$  de tamaño 26x1 el cual contiene la información de imagen y permite reducir la dimensionalidad de la misma sin perder mucha información, y reduce el tamaño de la imagen de 70x50 a una de 7x5, también este vector de características es usado como patrones de entrenamiento en la red neuronal.

**Figura 36.** Reducción de la imagen a un tamaño de 7x5

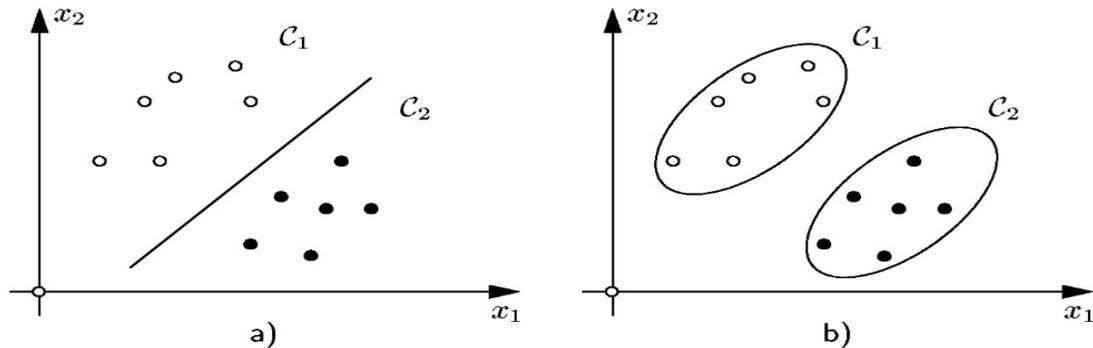


Fuente: Autor del proyecto

**4.4.1 Implementación de un OCR con redes neuronales con funciones de base radial.** Características generales y diferencias resaltantes con los modelos neuronales multicapas (Backpropagation): Broomhead y Lowe, 1988, introducen un método alternativo al Perceptrón multicapa (MLP) (ej.: Backpropagation) para hacer ajuste a funciones no lineales. Esto es clasificación no lineal. Ver figura 37: las redes RBF. A diferencia de la disposición que se tiene en la funciones de activación que permite

construir modelos de entrenamiento mediante Backpropagation, estas nuevas redes basadas en RBF construyen sus modelos con funciones de activación que son diferente tanto en la capa oculta como la de salida. Esto es, una red RBF está diseñada con neuronas en la capa oculta activadas mediante funciones radiales de carácter no lineal con sus centros gravitacionales propios y en la capa de salida mediante funciones lineales.

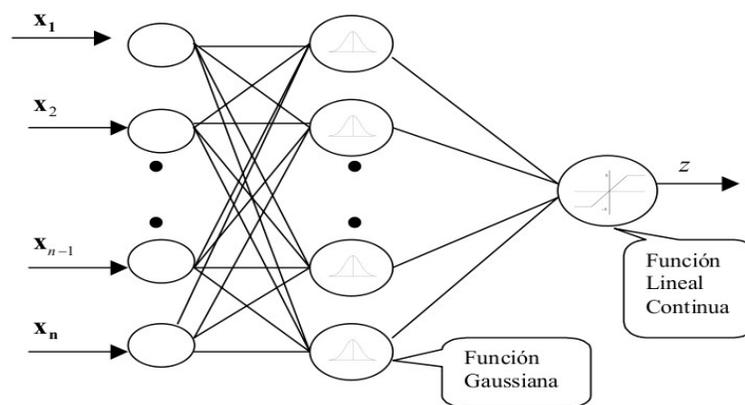
**Figura 37.** a) El Perceptrón representa límites de decisión (lineales). b) Las RBF-NN representan clases (a través de las funciones base).



Fuente: Autor del proyecto

El modelo clásico de las redes RBF está construido con una arquitectura rígida de tres capas: la de entrada, la oculta y la de salida. En general, una red RBF tiene un mejor desempeño con un mayor volumen de datos de entrenamiento. La construcción de una red RBF requiere de una mayor cantidad de neuronas en los nodos ocultos que en las redes que usan Backpropagation. Estas redes son método alternativo para aproximar funciones y clasificar patrones<sup>52</sup>.

**Figura 38.** Arquitectura de una RBF-NN



Fuente: [http://webdelprofesor.ula.ve/economia/gcolmen/programa/redes\\_neuronales/capitulo4\\_funciones\\_bases\\_radiales.pdf](http://webdelprofesor.ula.ve/economia/gcolmen/programa/redes_neuronales/capitulo4_funciones_bases_radiales.pdf)

<sup>52</sup> Disponible en internet:

[http://webdelprofesor.ula.ve/economia/gcolmen/programa/redes\\_neuronales/capitulo4\\_funciones\\_bases\\_radiales.pdf](http://webdelprofesor.ula.ve/economia/gcolmen/programa/redes_neuronales/capitulo4_funciones_bases_radiales.pdf), consultado el 8 de mayo de 2009.

**4.4.2 Como funciona una RBF-NN.** Tal como ya se dijo anteriormente, una red RBF está conformada de tres capas:

1. La capa de entrada que sirve para los ejemplos o patrones de entrenamiento y prueba.
2. La capa oculta completamente interconectada entre todos sus nodos con la capa de entrada y activada a través de la función radial (gaussiana).
3. La capa de salida, también completamente interconectada a la capa oculta y activada a través de una función lineal continua.

El entrenamiento, a diferencia de la red usando Backpropagation, es solamente hacia delante. De este modo, la salida  $z$  de una red RBF, en general, está influenciada por una transformación no lineal originada en la capa oculta a través de la función radial y una lineal en la capa de salida a través de la función lineal continua<sup>53</sup>.

**4.4.3 Arquitectura.** Se tiene un espacio de entrada de  $p$  dimensiones, para el OCR es de 35 dimensiones, y un espacio de salida de  $r$  dimensiones, para el OCR es de 26 dimensiones para el caso del alfabeto o 35 dimensiones si se incluyen los caracteres para los números excluyendo al cero.

Se cuenta con  $m$  neuronas ocultas. El patrón de entrada es el siguiente.

$$X(n) = (x_1(n), x_2(n), \dots, x_p(n))$$

Las activaciones de las neuronas de salida  $Y_k(n)$  para el patrón de entrada  $n$  serán:

$$Y_k(n) = \sum_{i=1}^m W_{i,K} \phi(n) + \mu_K \text{ para } K = 1, 2, \dots, r$$

$W_{i,K}$  : Peso de la conexión de la neurona oculta  $i$  a la de salida  $K$   
 $\phi(n)$  : Activación de la neurona oculta  $i$   
 $\mu_K$  : Umbral de la neurona de salida  $K$

$$\phi_i(n) = \phi\left(\frac{\|X(n) - C_i\|}{d_i}\right) \text{ para } i = 1, 2, \dots, m$$

$\phi$ : es una función de base radial

$C_i$ :  $(C_{i,1}, C_{i,2}, C_{i,3}, \dots, C_{i,p})$

$d_i$ : Son números reales, desviaciones de las funciones.

$\| \cdot \|$ : La distancia euclidiana desde el vector de entrada hasta el centro de la función.

$$\|X(n) - C_i\| = \sqrt{\sum_{j=1}^{j=p} (x_j(n) - C_{i,j})^2}$$

<sup>53</sup> Disponible en internet:

[http://webdelprofesor.ula.ve/economia/gcolmen/programa/redes\\_neuronales/capitulo4\\_funciones\\_bases\\_radiales.pdf](http://webdelprofesor.ula.ve/economia/gcolmen/programa/redes_neuronales/capitulo4_funciones_bases_radiales.pdf), consultado el 8 de mayo de 2009.

Donde la función de base radial más usada es la función Gaussiana:

$$\phi(r) = e^{-\frac{r^2}{2}}$$

Para el entrenamiento de la red: Se tomaron 11 ejemplos de entrenamiento por cada letra del alfabeto y se le realizó tanto el procesado de la imagen como la extracción de las características obteniendo de este proceso un conjunto de vectores de dimensión 26 con la información de cada imagen, asociando a cada uno de estos vectores un vector de longitud 26 con las siguientes características. A continuación se muestran las posiciones y valores respectivos de las posiciones para el vector de salida.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>	<b>J</b>	<b>K</b>	<b>L</b>	<b>M</b>	<b>N</b>	<b>O</b>	<b>P</b>	<b>Q</b>	<b>R</b>	<b>S</b>	<b>T</b>	<b>U</b>	<b>V</b>	<b>W</b>	<b>X</b>	<b>Y</b>	<b>Z</b>

En el vector de salida se marca con un uno la posición que corresponda al carácter que representa la imagen que se esté utilizando para entrenar la red, las demás posiciones permanecerán en cero.

Por ejemplo para la letra A el vector que representa esta letra es el siguiente:

A= [ 1 0 ]

De los ejemplos de entrenamiento se deben usar el 70% de las muestras para entrenar y el 40% para probar el desempeño de la red. En Matlab para entrenar una red neuronal de base radial se usan tres parámetros:

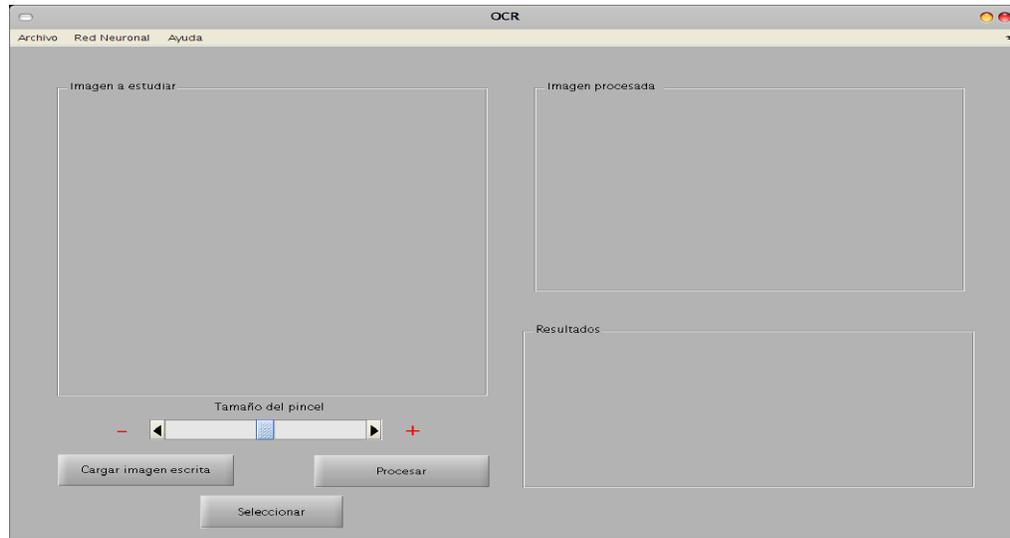
- *Meta (goal)*: que es la precisión con que se quiere se realice el proceso de entrenamiento no se debe colocar un valor de cero pues la red tiende a memorizar los ejemplos de entrenamiento y la generalización que realice es muy mala. Para la red que se entreno se uso un goal= 0.00001 y mostró un buen desempeño.
- *Spread*: es el radio con el que se quiere tengan las funciones de base radial en este caso se uso un radio de 1.
- *mr* = el número máximo de neuronas en la capa oculta, para este tipo de redes neuronales se deben usar mas neuronas que en los modelos de propagación hacia atrás. Pero al igual que con la meta no se deben colocar números de neuronas muy grande pues la red tiende a memorizar para el caso particular se uso un número máximo de neuronas de 100.

Este programa permite reconocer uno o varios caracteres impresos o manuscritos a partir de una imagen que se le monte o se le escriba. El reconocimiento de caracteres está basado en el uso de redes neuronales tipo Backpropagation o redes con funciones de base radial.

El funcionamiento del programa es el siguiente: En la pantalla de inicio, Ver figura 39, se muestran todas las posibles alternativas que se tienen con el programa. Si es la primera vez que se usa se puede probar con algunas redes neuronales que vienen pre entrenadas y listas para su uso. Para ello en el menú Archivo se escoge la opción Cargar red, una vez

cargada la red, en ese mismo menú, se puede o cargar una imagen o escribir nuestros propios caracteres. Mediante las opciones cargar imagen o escribir.

**Figura 39.** Pantalla principal



*Fuente:* Autor del proyecto

En el caso de montar la imagen, Ver Figura 40, una vez montada si esta tiene la característica de componerse de una sola palabra y en forma horizontal puede reconocer todos los caracteres que la componen.

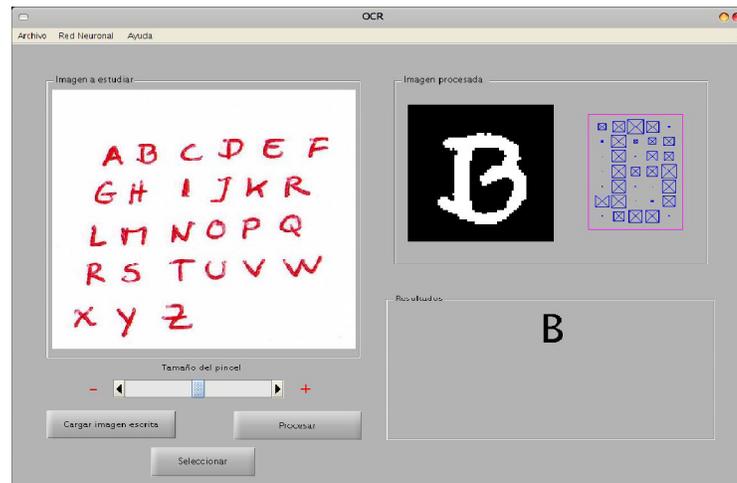
**Figura 40.** Reconocimiento de caracteres a partir de una imagen (a).



*Fuente:* Autor del proyecto

En el caso de que no se cumpla con la anterior restricción, Ver Figura 41, se puede presionar sobre el botón seleccionar y escoger con el ratón y escoger el carácter que se desee reconocer.

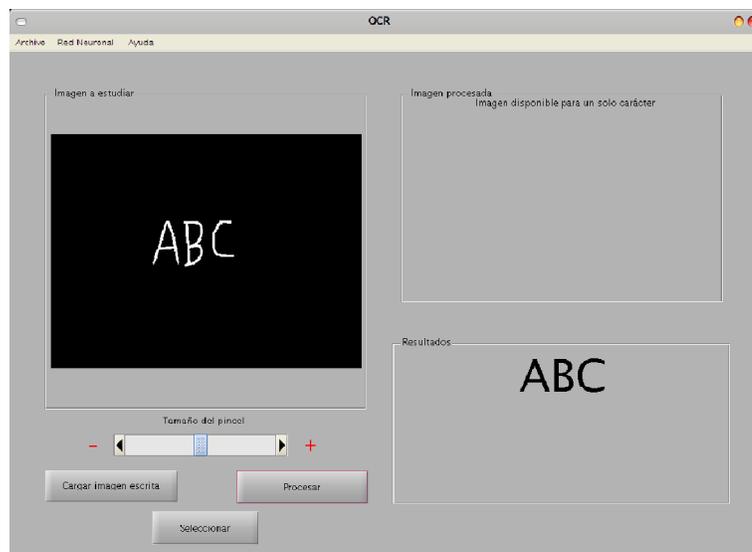
**Figura 41.** Reconocimiento de caracteres a partir de una imagen (b)



*Fuente:* Autor del proyecto

En el caso de las imágenes escritas, ver Figura 42, se puede reconocer uno o varios caracteres, después de haber ido al menú archivo y seleccionar la opción escribir y haber escrito lo que se deseaba. En la pantalla principal se presiona el botón cargar imagen escrita y luego procesar.

**Figura 42.** Reconocimiento de texto manuscrito



*Fuente:* Autor del proyecto

Si lo que se quiere es entrenar una nueva red neuronal, se debe seleccionar el menú *Red Neuronal* y escoger la opción *Entrenar Red*, Ver Figura 43. Con lo cual se abre la ventana para entrenamiento de redes. Para entrenar una nueva red se puede partir de unos datos de entrenamiento previamente almacenados, para esto se escoge el menú *Archivo* y se escoge la opción *Cargar entradas y salidas de la red*.

**Figura 43.** Entrenamiento de redes neuronales



Fuente: Autor del proyecto

Luego se debe presionar alguno de los botones para escoger el tipo de red que se quiere entrenar, *Backpropagation* o de *Base Radial*. Por defecto estas redes ya tienen sus parámetros configurados, pero si es necesario modificar estos parámetros se escoge el menú *Herramientas* y se presiona sobre *parámetros Backpropagation* o *parámetros Base Radial*, Ver Figura 44 y 45.

**Figura 44.** Parámetros para redes Backpropagation

The screenshot shows a window titled "RedParame" with the following parameters:

- Número de capas ocultas: 1
- Algoritmo de Entrenamiento: traingdx
- Neuronas por capa (sub-panel):
  - capa: [dropdown]
  - Número de neuronas: [input field]
  - Función de activación: logsig
  - Ingresar button
- Parámetros libres (sub-panel):
  - Número de Epocas: 5000
  - Goal: 0.0001
  - mc: 0.0095
- Guardar button

*Fuente:* Autor del proyecto

**Figura 45.** Parámetros para red de base radial

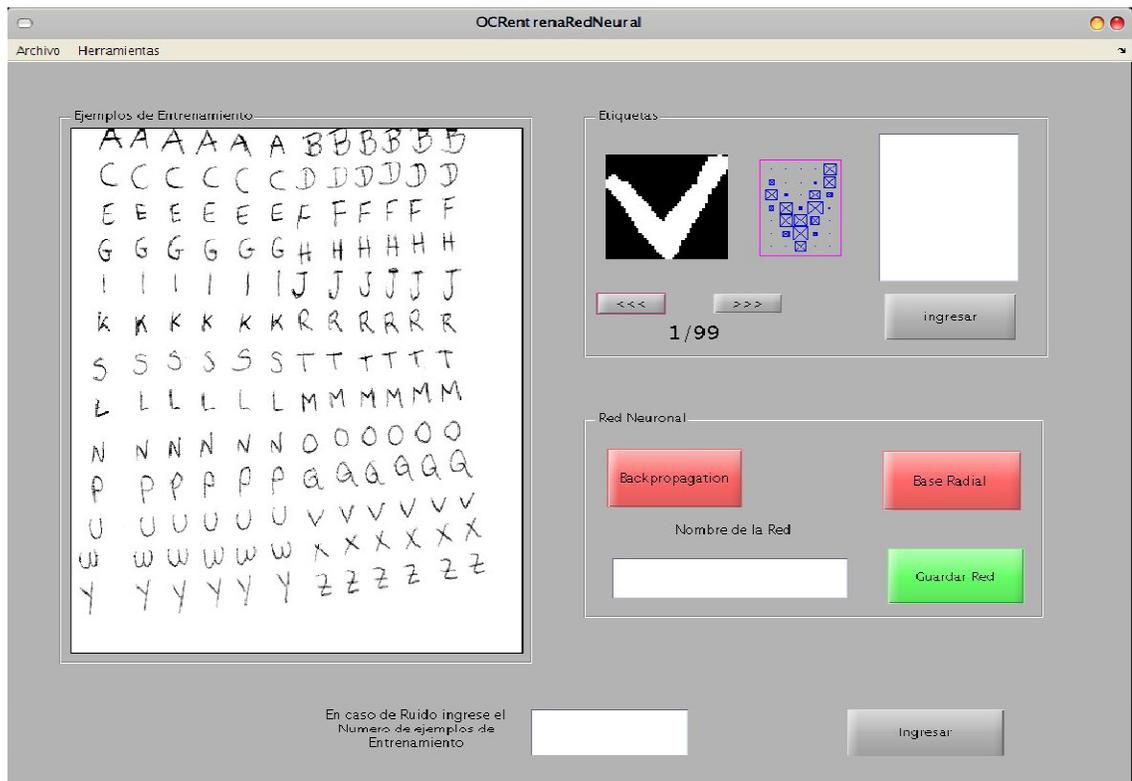
The screenshot shows a window titled "parambaseradial" with the following parameters:

- Goal: [input field]
- Spread: [input field]
- MN: [input field]
- Guardar button

*Fuente:* Autor del proyecto

Para entrenar una red con nuevos ejemplos de entrenamiento se escoge el menú *Archivo* y se escoge la opción *Cargar ejemplos de entrenamiento*. Ver Figura 46.

**Figura 46.** Ejemplos de entrenamiento



*Fuente:* Autor del proyecto

En la parte superior derecha de la ventana se encuentra el botón ingresar y dos botones más que permiten desplazarse por los diferentes ejemplos de entrenamiento y ser etiquetados con sus respectivos valores. Al terminar el etiquetado se puede entrenar la red neuronal y guardar la red asignándole un nombre y presionando el botón Guardar Red. También se pueden guardar los datos correspondientes a los ejemplos de entrenamiento escociendo el menú *Archivo* y seleccionando la opción *Guardar ejemplos de entrenamiento*.

En el caso que las imágenes de entrenamiento tengan ruido en la parte inferior de la ventana se puede ingresar el numero de ejemplos de entrenamiento que se van a usar, se presiona el botón *Ingresar* y acudiendo al menú *Herramientas* se selecciona la opción *Imágenes con ruido*. Una vez terminado el proceso de entrenamiento de la red se puede cerrar esta ventana y regresar a la ventana inicial del programa.

## CONCLUSIONES

Para el reconocimiento de caracteres ópticos mediante el método de correlación se pudieron concluir las siguientes afirmaciones:

- La utilización de los elementos estructurantes, definidos en Matlab, es una herramienta poderosa para simplificar los procedimientos a realizar cuando se trabajamos con imágenes y debemos eliminar estructuras y objetos irrelevantes.
- Las imágenes se pueden recortar si se desea analizar un área específica mediante la función `imcrop`.
- El tamaño más adecuado de la letra para la normalización es de 150 x 100 píxeles, de esta forma se reduce las dimensiones de la imagen para su análisis conservando sus formas y características fundamentales.
- La imagen de entrada puede ser con letras de color o no.
- El grosor de la letra debe ser semejante al del grosor de la plantilla de caracteres (muestras). Si el tamaño de la letra es delgado, se debe usar la función `imdilate` para aumentar su grosor antes de pasar al programa o de manera práctica mediante la herramienta de Matlab aumentar el tamaño del pincel con el que se desea escribir el carácter a comparar, de esta forma se puede identificar con que grosor se puede escribir.
- En un PC con 2 GB de RAM el tiempo de procesamiento de un carácter de prueba fue de 1.75236 segundos.
- Al ser mayor el vocabulario a reconocer, mayor es el tiempo de procesamiento de la imagen y mayor es la posibilidad de error en el reconocimiento.
- El método de correlación permite de manera eficiente el reconocimiento de imágenes.
- La efectividad del método de correlación es muy buena para sistemas que manejan vocabulario restringido como es el caso del presente aplicativo, generando un porcentaje de efectividad alrededor del 87%.

En el caso de las redes neuronales, se puede mencionar múltiples aplicaciones dirigidas a la solución de problemas en la vida cotidiana del hombre y que son de gran complejidad. En general, debido a que su funcionamiento está basado en el comportamiento del cerebro humano, las redes neuronales son muy bien referenciadas ya que no se ven afectadas por las emociones y sentimientos que si son frecuentes en el hombre. Las conclusiones obtenidas en el desarrollo del algoritmo con RNA son:

- El grosor de la letra debe ser semejante al del grosor de los ejemplos de entrenamiento.
- Es fundamental realizar un entrenamiento adecuado con los ejemplos de entrenamientos determinados para el reconocimiento de los caracteres ópticos.
- Para la aplicación de las RNA's se construyeron 2 tipos de redes neuronales: La Red Backpropagation y la Red de Base Radial.
- En un PC con 2 GB de RAM el tiempo de procesamiento de un carácter de prueba fue de 3.14571 segundos.
- Para el entrenamiento de la Red Backpropagation se debe tener en cuenta que los datos de entrada de esta red deben ser mínimos, por ejemplo el vocabulario o los caracteres numéricos y asegurarse que al realizar el entrenamiento del conjunto de entrada hayan una buena cantidad de ejemplos por carácter a identificar, para que la red realice un adecuado aprendizaje. En los parámetros de la red se encontraron: número de capas, número de neuronas por capa, función de activación, número de épocas, goal y mc.
- La Red de Base Radial permite un mejor reconocimiento de caracteres cuando se introduce una cantidad de ejemplos amplia para cada carácter a identificar, de esta forma al entrenar la Red el sistema pueda separar teóricamente en un hiperplano euclidiano regiones en donde se puedan localizar cada uno de ellos, y a su vez pueda agruparlos mediante el uso de funciones exponenciales. De tal forma, al intentar reconocer un carácter se puede localizar el centro de masa más apropiado en el conjunto de masas localizadas en el hiperplano en donde se encuentran el conjunto de cada uno de los caracteres.
- La efectividad mostrada por la Red Backpropagation estuvo en un 78% de reconocimiento para conjuntos de datos de entrada pequeño y sigue disminuyendo a medida que aumenta el vocabulario a reconocer. Esto la hace una red inapropiada para el reconocimiento de caracteres ópticos.
- Por otro lado, se logró un porcentaje más amplio y esperanzador en el uso de la Red de base radial, obteniendo un porcentaje en el 91%, gracias a que su método es más robusto y requiere de conjuntos de entrada amplios para su correcto entrenamiento.

## RECOMENDACIONES

Para investigaciones y logros posteriores a los realizados en el presente proyecto se recomienda el reconocimiento no solo de caracteres aislados sino concatenados que forman palabras o hasta frases, de tal forma que la complejidad de la tarea sería mucho más compleja siendo necesario el uso de métodos globales y probabilísticos que permitan un mayor reconocimiento de los caracteres cuidando la secuencia gramática de las palabras o frases.

Debido a que una de las mayores restricciones del desarrollo del programa fue la de utilizar un vocabulario bastante limitado y la de aislar cada uno de los caracteres, se recomienda el uso de sistemas de redes con arquitecturas más robustas, como la construcción de subredes neuronales especializadas para cada carácter o la del uso de modelos probabilísticos que realizan procesos estocásticos como lo son los modelos ocultos de Markov, que dada una secuencia de salida o un conjunto de tales secuencias, encuentra el conjunto de estados de transición y salidas más probables.

Con los objetivos alcanzados en esta tesis se da un paso hacia la tecnología del reconocimiento de caracteres y se empiezan a proponer alternativas y campos de investigación como los mencionados anteriormente para lograr resultados reduciendo al máximo el error y aumentar la efectividad en el proceso de reconocimiento.

## BIBLIOGRAFÍA

- BALLESTEROS CÁRDENAS, Anderson E.; CABALLERO MORENO, Zaira L. Reconocimiento de Placas de Automóviles usando Redes Neuronales. Bucaramanga, 2006. Trabajo de grado. Universidad Pontificia Bolivariana (UPB). Facultad de Ingeniería Electrónica.
- CUEVAS JIMENEZ, Erik Valdemar y ZALDIVAR NAVARRO, Daniel. Visión por Computador utilizando Matlab y el Toolbox de Procesamiento Digital de Imágenes [online]. [Consultado el 15 de mayo de 2009]. Disponible en internet: <[proton.ucting.udg.mx/tutorial/vision/cursovision.pdf](http://proton.ucting.udg.mx/tutorial/vision/cursovision.pdf)>.
- DE ARMAS DOMÍNGUEZ, Roberto Carlos; BAUTISTA RODRÍGUEZ, Lener Samuel. Reconocimiento Automático de Caracteres Manuscritos y Marcas, orientado al procesado de Encuestas. Bucaramanga, 2003. Tesis (pregrado). Universidad Industrial de Santander (UIS). Escuela Ingeniería Eléctrica, Electrónica y Telecomunicaciones. Disponible en internet: <http://unired.uis.edu.co/bibresultados.do?instituciones=1&nos=2=10&q=gualdrón&tidct>, consultado el 18 de enero de 2009.
- Disponible en internet: [decsai.ugr.es/mia/practicas/g6/g6.pdf](http://decsai.ugr.es/mia/practicas/g6/g6.pdf), consultado el 18 de mayo de 2009.
- Disponible en internet: <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%2011076%20CAPITULO%202.pdf>, consultado el 8 de mayo de 2009.
- Disponible en internet: [http://www.des.udc.es/~adriana/TercerCiclo/CursoImagen/curso/web/Filtrado\\_Espacial\\_Ruido.html](http://www.des.udc.es/~adriana/TercerCiclo/CursoImagen/curso/web/Filtrado_Espacial_Ruido.html)>. Consultado el 17 de Mayo de 2009
- Disponible en internet: <<http://docs.google.com/gview?a=v&q=cache:aAtUNRYKyn8J:bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%252011076%2520CAPITULO%25202.pdf.pdf+http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1212/8/T%252011076%2520CAPITULO%25202.pdf.pdf&hl=es&gl=co>>. Consultado el día 18 de mayo de 2009
- Disponible en internet: [http://www.ieev.uma.es/tdi/www\\_netscape/TEMAS/Tdi\\_10/index.php#1.-¿Qué es la morfología y adaptado por el autor de proyecto.](http://www.ieev.uma.es/tdi/www_netscape/TEMAS/Tdi_10/index.php#1.-¿Qué es la morfología y adaptado por el autor de proyecto.) (Consultado: Mayo 27 de 2009)
- Disponible en internet: <http://www.monografias.com/trabajos60/deteccion-spam/Image26215.gif>, consultado el 18 de mayo de 2009.
- Disponible en internet: <http://saevius.iespana.es/descargas/Investigacion/Redes%20Neuronales/Redes%20Neuronales.pdf>, consultado el 8 de mayo de 2009.

- Disponible en internet: <http://www.sia.eui.upm.es/master/VA05.pdf>, consultado el 18 de mayo de 2009.
- Disponible en internet: <http://www.udabol.uni.cc/jornada2007/conferencias/documentos/redesneuronales.pdf>, consultado el 8 de mayo de 2009.
- Disponible en internet: <http://www.udb.edu.sv/Academia/Laboratorios/biomedica/Imagenes%20Medicas/guia1IM.pdf>, consultado el 9 de mayo de 2009.
- Disponible en internet: [http://viref.udea.edu.co/contenido/menu\\_alterno/apuntes/gusramon/seminario\\_invest\\_vi/05-correlacion-variables.pdf](http://viref.udea.edu.co/contenido/menu_alterno/apuntes/gusramon/seminario_invest_vi/05-correlacion-variables.pdf), consultado el 8 de junio de 2009.
- Disponible en internet: [http://webdelprofesor.ula.ve/economia/gcolmen/programa/redes\\_neuronales/capitulo4\\_funciones\\_bases\\_radiales.pdf](http://webdelprofesor.ula.ve/economia/gcolmen/programa/redes_neuronales/capitulo4_funciones_bases_radiales.pdf), consultado el 8 de mayo de 2009.
- FLORES ARMENTA, Luis Alberto. Red neuronal artificial. Junio de 2008. Área de Ingeniería en sistemas computacionales. Disponible en Internet: <http://www.itnogales.edu.mx/formatos/Red%20neuronal%20artificial.pdf>, consultado el día 18 de mayo de 2009.
- GARCÍA PÉREZ, David. Grupo de Visión Artificial. Manejo Básico de Imágenes con Matlab [online]. [Consultado el 15 de mayo de 2009]. Disponible en internet: <http://gva1.dec.usc.es/~antonio/docencia/20042005tci/practicas/apuntes-practica1.pdf>
- HERVELLA AZOUZI, Samira. Editor de Imágenes basado en Regiones. Aplicación en entorno Matlab [online]. I.T.Telecomunicaciones,esp. Sonido e Imagen EUETIT.-Terrassa 2005-2006. [Consultado el 15 de mayo de 2009]. Disponible en internet: <http://upcommons.upc.edu/pfc/bitstream/2099.1/3860/1/54960-1.pdf>.
- HILERA GONZALEZ, José R., ROMERO VILLAVERDE, Juan P., GUTIERREZ de MESA, José A. Sistema de Reconocimiento Óptico de caracteres (OCR) con Redes Neuronales. Universidad de Alcalá de Henares. Departamento de Ciencias de la Computación.
- TOSELLI, Alejandro Héctor. Reconocimiento de Texto Manuscrito Continuo. Valencia, 20 de marzo de 2004, 230 p. Tesis Doctoral. Universidad Politécnica de Valencia. Departamento de Sistemas Informáticos y Computación.
- TOSINA, María José. Diseño de un filtro morfológico para la reducción de fluorescencia en espectros Raman. Barcelona, 2008, 109 p. Tesis de grado. Universidad Politécnica de Catalunya. Departamento de Teoría de la Señal y Comunicaciones. Disponible en internet: [http://upcommons.upc.edu/pfc/bitstream/2099.1/6588/1/PFC\\_MARI\\_COMPLETO\\_definitivo.pdf](http://upcommons.upc.edu/pfc/bitstream/2099.1/6588/1/PFC_MARI_COMPLETO_definitivo.pdf).

- VILA, Alicia., SEDANO, Maximo., LOPEZ, Ana. Proyecto e-math financiado por la secretaria de estado de educación y universidades (MECD). Correlación lineal y análisis de regresión [online]. [Consultado el 08 de abril de 2009]. Disponible en internet: [edu/in3/emath/docs/RegresionLineal.pdf](http://edu/in3/emath/docs/RegresionLineal.pdf)>.