

# University of Castilla-La Mancha

A publication of the  
Department of Computer Science

## Redes STN y CPN para el Reconocimiento de Rostros

by

A. Fernández Caballero, Francisco J. Gómez,  
Juan Moreno and Miguel A. Fernández

Technical Report

**#DIAB-01-03-18**

March 2001

DEPARTAMENTO DE INFORMÁTICA  
ESCUELA POLITÉCNICA SUPERIOR  
UNIVERSIDAD DE CASTILLA-LA MANCHA  
Campus Universitario s/n  
Albacete – 02071 – Spain  
Phone +34.967.599200, Fax +34.967599224



# Redes STN y CPN para el Reconocimiento de Rostros

Antonio Fernández-Caballero <sup>1</sup>, Francisco J. Gómez <sup>1</sup>,  
Juan Moreno <sup>2</sup> y Miguel A. Fernández <sup>1</sup>

<sup>1</sup> Grupo de Investigación en Computación Neuronal e Inteligencia Artificial (GICoNIA)  
Departamento de Informática, Escuela Politécnica Superior  
Universidad de Castilla-La Mancha, 02071 - Albacete, España  
{caballer, fgomez, miki}@info-ab.uclm.es

<sup>2</sup> Departamento de Informática, Escuela Universitaria de Ingeniería Técnica Industrial  
Universidad de Castilla-La Mancha, 45071 - Toledo, España  
jmgarcia@iti-to.uclm.es

## Resumen

*Este artículo describe, en primer lugar, un modelo de reconocimiento de rostros humanos, utilizando redes espacio-temporales (STN). Esta red neuronal incorpora una capa de entrada que se encarga de transformar la imagen de entrada al sistema en una secuencia de vectores normalizados que serán aplicados a una red espacio-temporal. Finalmente, la red tiene una capa de salida que utiliza unidades outstar de Grossberg (1982).*

*Por otra parte, se presenta el funcionamiento de las redes de contrapropagación (CPN) para el reconocimiento facial. Básicamente el procedimiento que se sigue es convertir la imagen en un vector de números y pasarlas a la capa de entrada de la red. Como vectores de entrada hemos probado vectores de niveles de gris de imágenes y vectores de histogramas por filas de las imágenes.*

*El artículo recoge los aspectos más relevantes de los trabajos de Moreno et al. "Reconocimiento de Rostros Utilizando Secuencias de Histogramas como Tramas Espacio-Temporales" presentado en el IV Simposio Iberoamericano de Reconocimiento de Patrones SIARP'99, "Reconocimiento Facial Utilizando Redes de Contrapropagación" presentado en el Segundo Encuentro de Computación ENC'99 y "Face Recognition by Counter-Propagation Networks" presentado en el V Simposio Iberoamericano de Reconocimiento de Patrones SIARP'2000.*

*La finalidad última de este trabajo es realizar un estudio comparativo entre ambos enfoques neuronales aplicados al reconocimiento de rostros.*

**Palabras clave:** Reconocimiento de rostros, tramas espacio-temporales, redes de contrapropagación, redes neuronales.

## 1 Introducción

El reconocimiento de rostros es un problema ampliamente abordado utilizando diferentes tipos de características (Chellapa, Wilson y Sirohey, 1995). El principal problema que aparece es que las características varían con pequeñas modificaciones de la posición o el gesto del rostro.

Son numerosos los enfoques que hasta la fecha se han realizado en cuanto al reconocimiento facial. Las numerosas técnicas empleadas en reconocimiento de rostros incluyen en primer lugar la correspondencia, y ello, por similitud respecto de un patrón (Brunelli y Poggio, 1993), por eigenvalores (Kirby y Sirovich, 1990; Turk y Pentland, 1991; Moghaddam y Pentland, 1994) o por métodos bayesianos (Moghaddam, Wahid y Pentland, 1998). Otro enfoque está relacionado con el modelado de los rostros por regiones (Duc, Fischer y Bigün, 1997; Kotropoulos y Pitas, 1997). De especial interés es el estudio de las curvaturas u otras características geométricas propias de los rostros (Gordon, 1991; Tanaka, Ikeda y Chiaki, 1998). Los modelos deformables también han sido aplicados al reconocimiento de rostros (Lanitis, Taylor y Cootes, 1995), así como el análisis de la componente principal (Etemad y Chellappa, 1997; Zhao, Chellappa y Krishnaswamy, 1998). La literatura ofrece igualmente la posibilidad de reconocimiento de rostros mediante la combinación de diferentes características (Nastar y Mitschke, 1998).

Desde nuestro punto de vista, el problema debe ser abordado con herramientas de amplia tolerancia a fallos, de las que son un buen ejemplo las redes neuronales. Dentro de este campo, aparecen diferentes soluciones como las redes de retropropagación, perceptrón multicapa, etc. (Kosugi, 1993; Howell y Buxton, 1995; Poggio y Beymer, 1995; Gutta et al., 1996; Sato, Shah y Aggarwal, 1998). Nosotros hemos preferido abordar el problema desde otras dos perspectivas: las tramas espacio-temporales y las redes de contrapropagación.

## 2 STN para el reconocimiento de rostros

En primer lugar se propone en este trabajo una solución al reconocimiento de rostros, que convierte la imagen bidimensional en una secuencia espacio-temporal de vectores. Se trata de un modelo que, a partir de la imagen, calcula el histograma de cada una de las filas que la componen. Una imagen de  $NF$  filas y  $NC$  columnas con  $NG$  niveles de gris, dará lugar a  $NF$  histogramas, cada uno correspondiendo a una fila de  $NG$  componentes cada una. La secuencia ordenada de estos  $NF$  histogramas puede utilizarse para identificar una imagen. Aunque en teoría es posible que dos imágenes diferentes tengan la misma secuencia ordenada de histogramas, la posibilidad de que esta situación se produzca en un proceso de identificación es prácticamente nula.

En una secuencia de histogramas normalizados, cada histograma corresponde a una fila de la imagen. En ella, se ofrece una información robusta acerca de la luminosidad de la misma. Esta solución supone analizar el rostro como una secuencia del nivel de luminosidad desde la parte superior a la parte inferior del mismo. Por sus características, las redes espacio-temporales (Freeman y Skapura, 1993) son una buena herramienta para reconocer estas secuencias ordenadas con una gran tolerancia a fallos, por lo que la consideramos adecuada para la naturaleza de la señal de entrada (secuencia de vectores de histogramas normalizados).

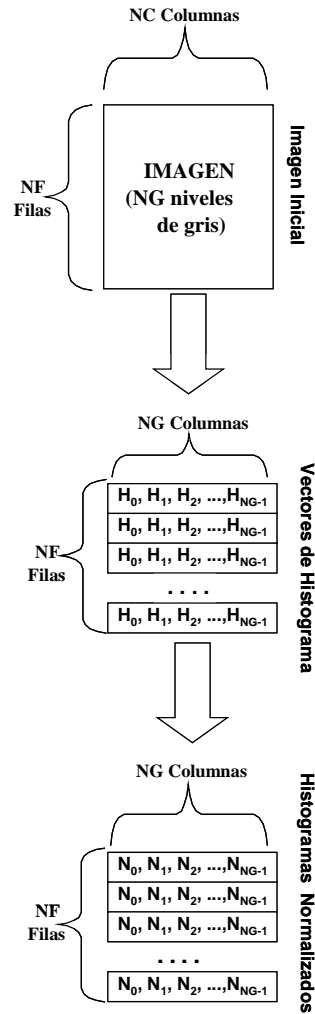


Figura 1. Función de la capa de entrada

## 2.1 Introducción

La solución que presentamos se describe como sigue. La imagen es la entrada a la primera capa (*capa de entrada*), que calcula los histogramas correspondientes a cada una de las filas de la imagen, y procede a continuación a normalizar dichos histogramas. Una capa interna recibe la salida de la capa de entrada. Dicha capa interna es una red espacio-temporal. Finalmente, una *capa de salida (outstar)*, tiene como función la de elegir el rostro ganador de entre todas los rostros aprendidos. Este proceso se muestra en la figura 1.

Cada imagen de entrada se corresponde con una secuencia de histogramas (vectores) normalizados; la secuencia de histogramas en el tiempo identificará la imagen. Sea  $\{Q_{11}, Q_{12}, \dots, Q_{1n}\}$  la secuencia de histogramas normalizados que entra en nuestra red; la avalancha (Grossberg, 1982) para el reconocimiento de un rostro viene representada en la figura 2.

Cada uno de los  $Q_{li}$  es un vector normalizado (función que realiza la *capa de entrada*), y se aplica a las entradas de todas y cada una de las unidades de la *STN*; se les permite permanecer allí durante un tiempo  $t$ ; después se aplica el vector siguiente  $Q_{li+1}$ , y así

sucesivamente. Durante ese tiempo  $t$ , cada unidad ajusta dinámicamente su activación y sus valores de salida de acuerdo con la regla que veremos más adelante. Obsérvese que la salida de cada unidad está conectada a cada una de las unidades sucesivas, de izquierda a derecha, con una intensidad de conexión  $d$ , donde  $0 < d < 1$ . Después de procesar todos los dígitos por las capas de *entrada* y *STN*, la *outstar* elige un rostro entre los aprendidos.

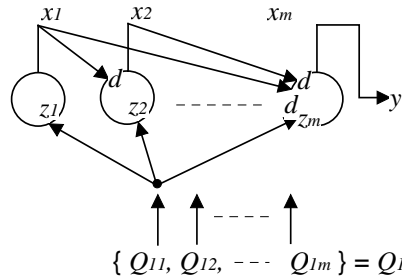


Figura 2. Estructura de la red

## 2.2 Descripción

Hemos diseñado una arquitectura de tres capas, como se muestra en la figura 3. Una *capa de entrada* con un único elemento de proceso transforma la imagen de entrada. Una *capa interna*, que responde a una red espacio-temporal, aprende una secuencia de vectores normalizados. Y la *capa de salida* asocia a la subred ganadora con un rostro de entre los aprendidos.

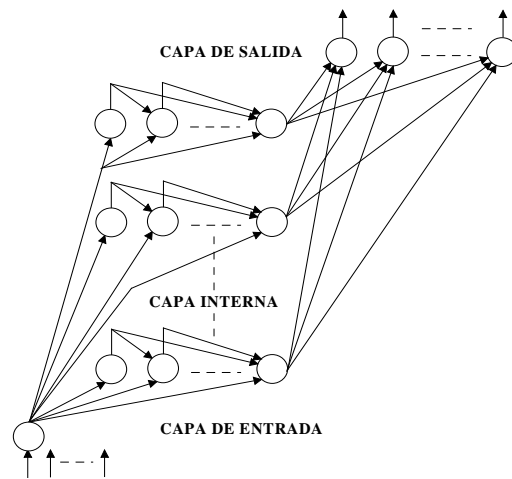


Figura 3. Arquitectura de nuestra red

La imagen se pasa a la única unidad de la *capa de entrada*. Entonces dicha capa propaga su salida (secuencia de histogramas normalizados) simultáneamente a todas las unidades de la *capa interna*. Una vez procesada toda la imagen por las capas de *entrada* e *interna*, la *capa de salida* asocia el grupo ganador de la *capa interna* con un rostro que muestra como salida.

### 2.2.1 La capa de entrada

La *capa de entrada* realiza la misión de transformar una imagen de entrada en una secuencia de vectores de histogramas normalizados. La imagen consta de NF filas y NC columnas, con NG niveles de gris. Esta capa calcula el histograma de cada una de las NF filas, dando como resultado una secuencia de NF histogramas de NG componentes cada uno. Por último, y dado que las STN (Hecht-Nielsen, 1986; Hecht-Nielsen, 1990; Freeman y Skapura, 1993) requieren que su entrada esté normalizada, se normaliza cada uno de estos NF histogramas según la ecuación:

$$X = \left( \frac{x_1}{\sqrt[2]{x_1^2 + \dots + x_{NG}^2}}, \frac{x_2}{\sqrt[2]{x_1^2 + \dots + x_{NG}^2}}, \dots, \frac{x_{NG}}{\sqrt[2]{x_1^2 + \dots + x_{NG}^2}} \right) \quad (1)$$

### 2.2.2 La capa interna (STN)

En nuestra arquitectura, esta capa corresponde a una red espacio-temporal, compuesta de NI\*NF unidades, donde NI es el número de imágenes (rostros) a aprender, y NF es el número de elementos (histogramas) de que consta una secuencia. Estas redes se rigen por una serie de ecuaciones que pueden consultarse en Freeman y Skapura (1993) y Moreno et al. (1998).

La *capa interna* está dividida en grupos de NF unidades, cada una de las cuales se encarga de aprender y reconocer los NF histogramas correspondientes a una imagen. La salida de la última unidad de cada grupo será el valor de salida del grupo. La unidad de salida de cada grupo que tenga mayor respuesta será la única que tenga salida no nula. Es decir, convertimos a 1 la mayor de las salidas de cada grupo, y a 0 el resto. Dichas salidas convertidas son las que la *capa de salida* recibirá como entrada.

Con respecto a los parámetros, a  $\delta t$  le hemos asignado el valor 1, con el fin de dejar tal cual la salida del valor de actualización. Al parámetro  $c$ , que representa el valor de descarga de la intensidad de salida de cada unidad, le hemos asignado el valor 1/NF con el fin de que una vez cargada una unidad, ésta no se descargue rápidamente. La *entrada total* se filtra con un *valor umbral* ( $\Gamma$ ). Para nuestra aplicación hemos asignado a  $\Gamma$  el valor 0, puesto que un valor umbral mayor no permitiría que las primeras unidades de la red se activasen, ya que el vector normalizado tiene componentes con valor muy bajo, debido a que estamos trabajando con 256 componentes (niveles de gris). A los parámetros  $a$  y  $b$ , que indican la importancia de la salida anterior y la *entrada total* respectivamente, les hemos asignado a ambos el valor 1, puesto que queremos que pese de igual forma la activación anterior de la unidad y la entrada a la unidad en un instante dado. El parámetro  $d$  representa la influencia de las conexiones de las unidades anteriores a la *entrada total* de la unidad. Hemos decidido que tome el valor 1/NF para no perder la carga de una secuencia de histogramas.

### 2.2.3 La capa de salida

Finalmente, la *capa de salida* consta de NF elementos de proceso llamados *outstar* (Grossberg, 1982; Freeman y Skapura, 1993). La *outstar* alcanza rápidamente un valor de equilibrio igual al valor del peso que haya en la conexión procedente de la unidad ganadora de entre las salidas de los grupos. Una forma sencilla de visualizar este procesamiento consiste en darse cuenta de que la salida de equilibrio de la *outstar* es igual a la entrada de la *outstar*,

$$y_k^{eq} = \sum_j w_{kj} z_j$$

donde  $z_j$  es la entrada que se recibe del correspondiente grupo de la *STN*. Dado que  $z_j = 0$  a no ser que  $j=i$ , entonces:

$$y_k^{eq} = w_{ki} z_i = w_{ki} \quad (2)$$

Este sencillo algoritmo utiliza valores de equilibrio de las actividades y salidas de los nodos. De esta manera, se evita tener que resolver numéricamente todas las ecuaciones diferenciales correspondientes.

### 2.3 Aprendizaje

En este apartado vamos a describir el aprendizaje de las distintas capas de nuestra arquitectura.

La *capa de entrada* no necesita una fase de aprendizaje, pues para la obtención de los NF histogramas normalizados no es necesario ningún aprendizaje. El aprendizaje de cada unidad de la *STN* se realiza a partir del vector de pesos inicial  $w$ , que va evolucionando según la ecuación diferencial  $\dot{w} = -cw + dIy$ , en donde  $y$  es la salida, y  $c, d > 0$ . La misión de una unidad de la *STN* es aprenderse un vector de entrada  $y$  normalizado, proporcionando una mayor intensidad de salida cuanto más se parezca el vector de entrada al vector aprendido. Por otra parte, los pesos de cada unidad *outstar* de la *capa de salida* evolucionan durante su proceso de aprendizaje, según la ecuación diferencial

$$\dot{y}'_i = -ay'_i + by_i + c * net_i$$

donde  $a, b, c > 0$  y  $net_i = I * W$ .

### 2.4 Implementación

Para probar la solución propuesta se ha implementado un simulador utilizando el lenguaje de programación *Visual C++ 6.0*. Dicho simulador acepta como entrada una imagen y genera como resultado otra imagen. La imagen de entrada corresponde al rostro que queremos reconocer y la de salida al rostro seleccionado de entre los aprendidos.

### 2.5 Resultados

Para la verificación de resultados hemos preparado una batería de pruebas, formada por 16 series de 6 imágenes cada una. Las imágenes tienen una dimensión de 112 filas por 92 columnas, con 256 niveles de gris. Dichas series se muestran en la figura 5.

El aprendizaje se ha realizado mostrando al sistema únicamente la primera imagen de cada serie. Los resultados obtenidos se indican en la tabla 1. Un símbolo **X** indica que la imagen de la serie correspondiente no ha sido reconocida.



		Imagen					
		1	2	3	4	5	6
Series	1						
	2						
	3			×	×	×	×
	4				×	×	×
	5						
	6						
	7					×	
	8				×	×	×
	9				×	×	×
	10						
	11						
	12			×	×	×	
	13			×	×		
	14			×		×	
	15			×	×	×	×
	16			×	×	×	×

Tabla 1. Resultados

Como se puede observar, 6 de las 16 series son reconocidas totalmente. Todas ellas tienen en común que la luminosidad se mantiene constante en toda la serie. Algunas veces, en series en las que se aprecia un cambio importante de luminosidad, se producen fallos en la identificación; véase como ejemplo la serie 16 en las imágenes 4 y 5. Otra situación que provoca fallos es que en una misma serie el individuo pueda aparecer con o sin gafas. Véanse como ejemplo, las series 4 y 13, aunque hay casos en que no representa un problema (véase serie 7, imagen 4).

Este método presenta una gran tolerancia a fallos, obteniéndose muy buenos resultados cuando se produce un giro o movimiento de la cabeza que no afecta a la luminosidad de la imagen. Véanse como ejemplo las series 1, 5 y 11. Las series elegidas son representativas de un gran abanico de rostros posibles (mujeres, hombres, con o sin barba, calvos o con pelo, con gafas o sin ellas, rubios o morenos, etc.), obteniendo un resultado global más que aceptable. Así, de 96 imágenes, sólo se han presentado 29 fallos en la identificación, lo cual nos da un 70% de aciertos.

Hay que tener en cuenta que no se ha realizado ningún tipo de preprocesado a las imágenes para intentar corregir o mejorar ciertas características, como por ejemplo la luminosidad. Además, casi todas las imágenes contienen una zona de fondo que no es significativa y que por tanto se puede eliminar, lo cual reduciría con seguridad la tasa de fallos.

### 3 CPN para el reconocimiento de rostros

Esta segunda parte del artículo pretende comprobar la eficiencia de las redes de contrapropagación (CPN) para el reconocimiento facial. Para comprobar la red se ha implementado un simulador que tiene un proceso de aprendizaje por medio del cual aprende los mismos 16 rostros, y posteriormente recibe como entrada variaciones de cada uno de los rostros aprendidos, proporcionando como salida un código que representa a uno de los rostros aprendidos.

### 3.1 Introducción

Hecht-Nielsen (Hecht-Nielsen, 1990) sintetizó la CPN combinando una estructura conocida como red competitiva con la estructura outstar de Grossberg (Grossberg, 1982; Hecht-Nielsen, 1987a; Hecht-Nielsen, 1987b), obteniendo así lo que se llama red de contrapropagación. El funcionamiento de la red se muestra en la figura 4. Dado un conjunto de vectores  $(x_1, y_1), (x_2, y_2), \dots, (x_L, y_L)$ , la red puede aprender a asociar a un vector X en la capa de entrada con un vector Y en la capa de salida. Si la relación entre X e Y se puede definir mediante una función continua,  $\Omega$ , tal que  $Y = \Omega(X)$ , entonces la red aprenderá a aproximar esa correspondencia para todo valor de X en el intervalo especificado por el conjunto de vectores de entrenamiento.

Como se ve en la figura 4, la arquitectura consta de tres capas. Las unidades de la capa 1 reciben como entrada un rostro, que es procesado como un vector de niveles de gris, y normalizan dicho vector. Todas las unidades de la capa 2 reciben el mismo vector normalizado de entrada, y calculan su intensidad de salida gracias a un vector de pesos aprendido y al vector que reciben como entrada. La unidad de mayor intensidad de salida envía un 1 a la capa 3, el resto envía un 0. Finalmente, la capa 3 selecciona la imagen de salida de entre las aprendidas.

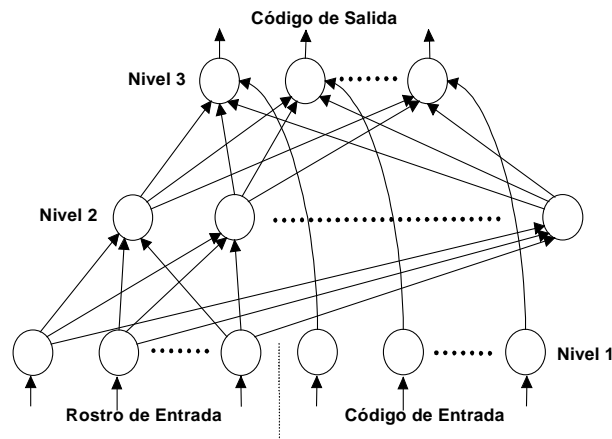


Figura 4. Estructura de la CPN

### 3.2 Descripción

#### 3.2.1 Capa 1

Consideremos la capa 1 de la figura 4. La intensidad total del vector de entrada está gobernada por  $I = \sum_i I_i$ . En correspondencia con cada  $I_i$ , definiremos una magnitud mostrada en la ecuación (3). El vector  $(\vartheta_1, \vartheta_2, \dots, \vartheta_n)^t$  es lo que se denomina trama de reflectancia (Freeman y Skapura, 1993). Obsérvese que esta trama está normalizada, es decir que  $\sum \vartheta_i = 1$ .

$$\vartheta = I_i \left( \sum_i I_i \right)^{-1} \quad (3)$$

La trama de reflectancia es independiente de la intensidad total de la trama de entrada correspondiente. Por ejemplo, la trama de reflectancia correspondiente a la

imagen de la cara de una persona es independiente de que la imagen de la persona tenga más o menos luminosidad. La ecuación (4) representa la salida de la capa 1, donde  $0 < x_i(0) < B$ ,  $A, B > 0$ .

$$\dot{x}_i = -Ax_i + (B - x_i)I_i - x_i \sum_{k \neq i} I_k \quad (4)$$

Los elementos de proceso alcanzan rápidamente un estado de equilibrio una vez que se aplica el vector de entrada  $X$ , ( $\dot{x} = 0$ ) (Freeman y Skapura, 1993). A la hora de la simulación de la CPN, se puede simplificar el programa normalizando por software los vectores de entrada. La simulación normaliza de acuerdo a la ecuación (5).

$$x_i = I_i / \left( \sqrt{\frac{\sum I_i^2}{n}} \right) \quad (5)$$

### 3.2.2 Capa 2

La capa 2 se conoce con el nombre de red competitiva. Consta de una serie de elementos de proceso llamados instar (Grossberg, 1982; Freeman y Skapura, 1993). El vector de entrada  $I$ , y el vector de pesos  $w$  se han normalizado. La salida de la instar está gobernada por la ecuación (6), donde  $neta = I * w$  y  $a, b > 0$ .

$$\dot{y} = -ay + b * neta \quad (6)$$

Las unidades instar alcanzan el valor de equilibrio cuando  $y^{eq} = (b/a) * neta$ . A la hora de la simulación los valores  $a$  y  $b$  han tomado el mismo valor, luego se eliminan de la formula. La red competitiva clasifica cualquier vector de entrada. La instar de mayor salida es la ganadora de la competición, y será la única que tendrá salida no nula.

### 3.2.3 Capa 3

La ganadora enviará a las outstar el valor 1, y el resto de instar de la red competitiva enviará el valor 0.

$$\dot{y}'_i = -ay'_i + c w_i^{eq}$$

La outstar alcanza rápidamente un valor de equilibrio igual al valor del peso que haya en la conexión precedente de la unidad ganadora de la red competitiva. Una forma sencilla de visualizar este procesamiento consiste en darse cuenta de que la salida de equilibrio de la outstar es igual a la ecuación (7),

$$y_k^{eq} = \sum_j w_{kj} z_j \quad (7)$$

donde  $z_j$  es la entrada que se recibe de la correspondiente instar de la red competitiva.

Además, dado que  $z_j = 0$  a no ser que  $j=i$ , la ecuación (8) es la salida de la outstar.

$$y_k^{eq} = w_{ki} z_i = w_{ki} \quad (8)$$

### 3.3 Aprendizaje

El aprendizaje de la CPN se realiza por capas. La capa 1 no tiene aprendizaje, ya que tan solo normaliza los valores de entrada. Como ya se dijo la capa 2 esta formada por unos elementos llamados instar. La instar se aprende el vector de pesos  $w$ . El aprendizaje (Moreno et al, 1998) se realiza a partir del vector de pesos inicial  $w$ , que va evolucionando según la ecuación diferencial (9), en donde  $y$  es la salida,  $y$ ,  $c$ ,  $d > 0$ .

$$\dot{w} = -cw + dly \quad (9)$$

La misión de una instar es aprenderse un vector de entrada  $y$ , proporcionando una mayor intensidad de salida cuando más se parezca el vector de entrada al vector aprendido. A la hora de la simulación es posible simplificar el aprendizaje de la instar, asignando directamente los valores de los pesos del vector  $w$  a partir de vector  $y$ , ya normalizados.

Finalmente, el aprendizaje de cada outstar de la capa 3 evoluciona según la ecuación (10), donde los parámetros  $a$ ,  $b$ ,  $c > 0$  y el valor de la  $neta_i$  se calcula de la forma ya descrita.

$$\dot{y}'_i = -ay'_i + by_i + c * neta_i \quad (10)$$

Al efectos de la simulación digital, se puede aproximar este aprendizaje asignando directamente los pesos de la outstar a partir de los vectores de entrada  $y$ .

### 3.4 Implementación

En la simulación la salida de la outstar se obtiene mediante la ecuación (8).

Como ya se ha comentado antes, nuestro sistema realiza una fase de aprendizaje, en la cual la CPN se aprende la serie 1 de los 16 rostros de la figura 5. A continuación se le pasan todos los rostros de dicha figura. Posteriormente el sistema devuelve como resultado un código que representa cada una de las imágenes aprendidas. En nuestra aplicación hemos utilizado imágenes de tamaño 112 filas y 92 columnas, con 256 niveles de gris. Finalmente el esquema de nuestra CPN es el siguiente:

- La capa 1 consta de  $112 \times 92$  nodos en el vector de entrada  $x$ , que corresponde a la longitud del vector de entrada. Con respecto a las entradas  $y$ , tenemos 4 nodos correspondientes a los 4 valores binarios de que se compone el código de cada uno de los rostros a aprender.
- La capa 2 (red competitiva) está formada por 16 nodos instar, donde cada instar aprende un rostro.
- La capa 3 o capa de salida, consta de 4 nodos, cada una de las cuales se aprende un número binario que permite identificar uno de los componentes del código de salida. La salida de esta capa será el rostro seleccionado.

Consideramos que una agenda con 16 rostros es una muestra significativa para comprobar la validez de la red de contrapropagación para este sistema, aunque nos consta que con más rostros la CPN tendría un comportamiento peor.

Para terminar este apartado comentar que para cada valor de entrada  $i$  se ha elegido una intensidad de entrada al sistema de  $i+1$ , para que cuando  $i=0$  se proporcione algún valor de entrada distinto de 0.

### 3.5 Resultados

Para la verificación de resultados se ha implementado un simulador de la CPN en *Visual C++ 6.0*. El simulador realiza previamente la fase de aprendizaje de las diversas capas. Y a continuación envían a la capa de entrada de ambas redes, uno a uno, todos los rostros a comprobar (figura 5, series 1 a 16).

Se han realizado dos tipos de pruebas: (a) pasando las imágenes como un vector de niveles de gris, (b) pasando la imagen como un vector de histogramas por filas de la imagen (Moreno et al, 1999). Hemos elegido histogramas por filas para solucionar el problema que tiene la CPN cuando el rostro no aparece exactamente en la misma posición de la imagen que en la imagen aprendida.

Los resultados obtenidos para los casos (a) y (b) son muy buenos, superando en ambos casos el 80%.

	Imagen 1	Imagen 2	Imagen 3	Imagen 4	Imagen 5	Imagen 6
S 1	✓		✓		✓	✓
S 2	✓	✓	✓	✓	✓	
S 3	✓	✓	✓	✓		✓
S 4	✓	✓	✓	✓		
S 5	✓	✓	✓	✓	✓	✓
S 6	✓	✓	✓	✓	✓	✓
S 7	✓	✓	✓	✓	✓	✓
S 8	✓	✓	✓	✓	✓	✓
S 9	✓	✓	✓	✓	✓	
S 10	✓	✓				✓
S 11	✓	✓	✓			✓
S 12	✓	✓	✓	✓	✓	✓
S 13	✓	✓	✓	✓	✓	✓
S 14	✓	✓	✓	✓	✓	✓
S 15	✓		✓	✓		✓
S 16	✓			✓	✓	✓

Tabla 2. Resultados usando escala de niveles de gris

La tabla 2 muestra las imágenes acertadas para el caso (a). El acierto global es del 83.3%. Uno de los problemas detectados, pese a la normalización que realiza la capa 1, se da cuando la imagen de entrada tiene una luminosidad distinta a la imagen aprendida. Ejemplos de esto son las series 1 (imágenes 2 y 4), 15 (imágenes 2 y 5) y 16 (imágenes 2 y 3). Otro problema, menos importante, es la presencia o ausencia de gafas. Por ejemplo en la serie 4 (imágenes 5 y 6), aunque en esta misma serie, la ausencia de gafas en la imagen 4 no supone ningún problema. Por otro lado, en las series 7 y 13 esta situación no es problemática.

La tabla 3 muestra las imágenes acertadas para el caso (b). El acierto global en este caso es del 86.6%. En este caso el problema de la presencia o ausencia de gafas está solucionado por completo. Mientras que el cambio de luminosidad entre la imagen aprendida y la de entrada sigue provocando confusiones en la detección de las imágenes, como se muestra en las series 3 (imagen 6), 9 (imágenes 4, 5 y 6) y 16 (imágenes 2, 3, 4 y 5).

	Imagen 1	Imagen 2	Imagen 3	Imagen 4	Imagen 5	Imagen 6
S 1	✓	✓	✓	✓	✓	✓
S 2	✓	✓	✓	✓	✓	✓
S 3	✓	✓	✓	✓	✓	
S 4	✓	✓	✓	✓	✓	✓
S 5	✓	✓	✓	✓	✓	✓
S 6	✓	✓	✓	✓	✓	✓
S 7	✓	✓	✓	✓	✓	✓
S 8	✓	✓	✓	✓	✓	✓
S 9	✓	✓	✓			
S 10	✓	✓	✓	✓	✓	✓
S 11	✓	✓	✓	✓	✓	✓
S 12	✓					✓
S 13	✓	✓	✓	✓	✓	✓
S 14	✓	✓	✓	✓	✓	✓
S 15	✓		✓	✓		✓
S 16	✓					✓

Tabla 3. Resultados usando histogramas por fila

## 4 Conclusiones

La finalidad principal de este trabajo ha sido la presentación de dos enfoques neuronales aplicados al reconocimiento de rostros con el fin de proceder a su comparación. Se han descrito dos modelos, utilizando redes espacio-temporales (STN) el primero y redes de contrapropagación (CPN) el segundo.

En el caso de la STN, la principal novedad aportada es el hecho de abordar el reconocimiento de rostros considerando éstos como una secuencia de histogramas, es decir, de luminosidad. Para nuestro sistema, la secuencia de histogramas por filas, desde la parte más alta del rostro hasta la parte inferior, configura una trama espacio-temporal que identifica dicho rostro. Como puede verse, la capacidad de la red espacio-temporal para absorber variaciones en las secuencias de entradas se adecúa muy satisfactoriamente a las variaciones que aparecen al variar la posición o el gesto del rostro, siendo la información de la secuencia de luminosidad almacenada en la STN suficiente para diferenciarlo del resto de rostros que han configurado la secuencia de entrenamiento.

Con respecto a los resultados de simulación, destacar que el índice de aciertos es del 70%. Además, se debe recordar que no se ha hecho ningún tipo de preprocesado a las imágenes de entrada. Comentar también que este método es bueno en los casos en que el rostro aparece girado o desplazado. Conviene hacer notar que uno de los puntos clave del diseño del sistema es el ajuste de los parámetros de funcionamiento, ya que un mal ajuste de éstos puede llevar a que la STN funcione inadecuadamente.

A la vista de los resultados obtenidos, consideramos que la solución propuesta puede ser adecuada para reconocimiento de rostros en aplicaciones donde el número de rostros a reconocer no sea excesivo, ya que esto haría a la STN computacionalmente inabordable a sus dimensiones.

Imagen 1 Imagen 2 Imagen 3 Imagen 4 Imagen 5 Imagen 6

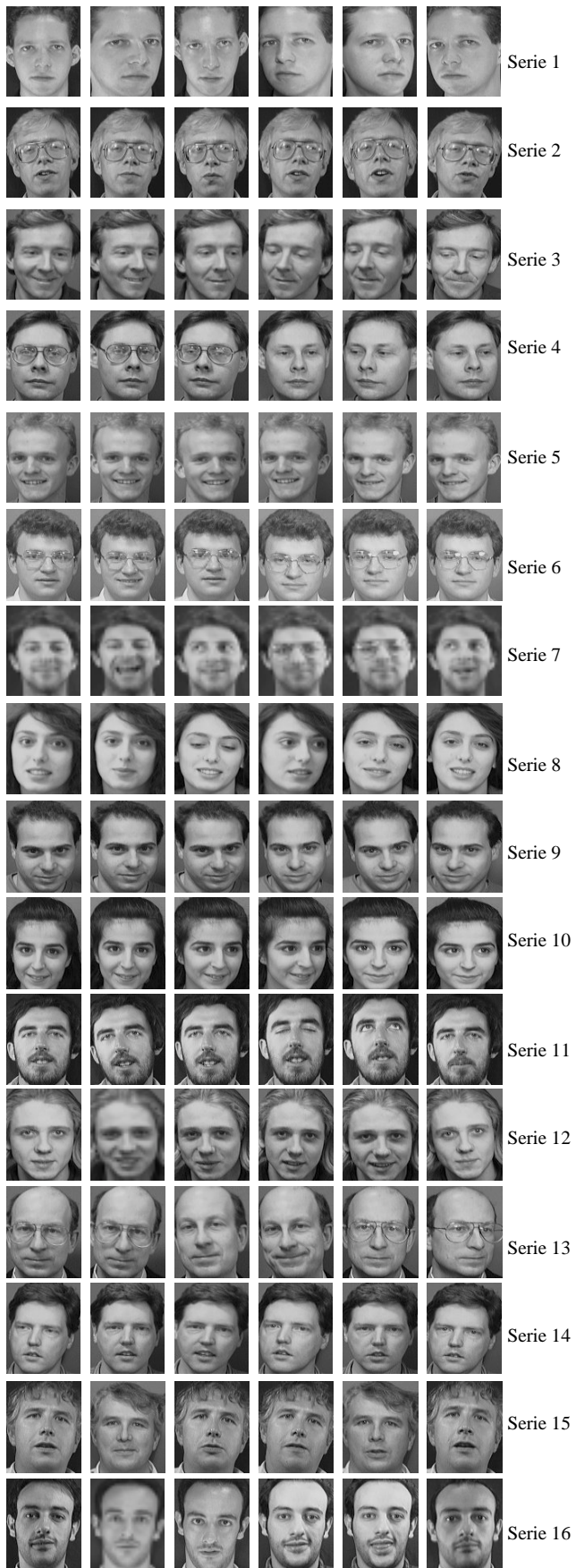


Figura 5. Imagen de rostros

En cuanto a la CPN, podemos concluir que es una buena herramienta para el reconocimiento facial, como demuestran los resultados obtenidos, tanto en el caso de tratar la imagen como un vector de niveles de gris (a), como en el caso de tratar la imagen como un vector de histogramas por filas (b).

Hemos elegido la red de contrapropagación para implementar este sistema ya que son buenas para la detección de vectores. En Moreno et al. (1998) se obtenían unos buenos resultados. Además, hemos de tener en cuenta que cuanto mayor es el número de coordenadas del vector mejor repartidas pueden estar las clases dentro del espacio. Las redes de contrapropagación emplean algoritmos de aprendizaje distintos para cada capa, que permiten a la red entrenarse muy rápidamente.

En el caso (a), la diferencia de luminosidad entre la imagen aprendida y la probada puede provocar en ocasiones confusiones, si bien la normalización de la capa 1 evita el problema en muchos casos.

En el caso (b) se superan los resultados del caso anterior. Esto se debe a que la entrada es el histograma por fila, y la posición relativa del rostro en la imagen no altera el resultado.

Tras ver que los resultados obtenidos con ambos enfoques son buenos, tenemos que destacar, finalmente, que los resultados conseguidos con la CPN superan ligeramente los conseguidos con la STN por nuestro grupo. En estudios anteriores de nuestro grupo de trabajo (Moreno et al., 1999, "Comparison of counter-propagation and spatio-temporal nets in the detection of sequences of numbers"), con la STN se obtuvieron unos malos resultados para el caso de tratar la imagen como un vector de niveles de gris, mientras que para el caso de tratar la imagen como un vector de histogramas por filas se consigue aproximadamente un 70% de aciertos. Este grado de aciertos, aunque sensiblemente mejor, sigue siendo muy bajo en comparación con el 86.6% de la CPN. Más aún si tenemos en cuenta que nuestro actual algoritmo de aprendizaje es muy sencillo, y que las imágenes no sufren ningún tipo de preprocesamiento.

## Agradecimientos

Damos las gracias por la cortesía de *The ORL Database of Faces* ([www.cam-orl.co.uk/facedatabase.html](http://www.cam-orl.co.uk/facedatabase.html)) por el uso público de estas imágenes.

## Referencias

**Brunelli R. & Poggio T.**, "Face recognition: Features vs. Templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 10, 1993.

**Chellappa R., Wilson C.L. & Sihorey S.**, "Human and machine recognition of faces, a survey", *Proceedings of the IEEE*, Vol. 83, 1995, 705-740.

**Duc B., Fischer S. & Bigün J.**, "Face identification with sparse grig gabor information", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP'97*, Vol. IV, Munich, 1997, pp. 3053-3056.

**Etemad K. & Chellappa R.**, "Discriminant analysis for recognition of human face images", *Journal of the Optical Society of America A*, Agosto 1997, 1724-1733.



**Freeman J.A. & Skapura D.M.**, *Redes neuronales. Algoritmos, aplicaciones y técnicas de programación*, Addison-Wesley/Díaz de Santos, 1993.

**Gordon G.G.**, “Face recognition based on depth maps and surface curvature”, SPIE, Vol. 1570, *Geometric Methods i Computer Vision*, 1991.

**Grossberg S.**, “Learning by neural networks”, *Studies of Mind and Brain*, Vol. 70, Boston Studies in the Philosophy of Science, D. Reidel Publishing Company, Boston, 1982.

**Gutta S., Huang J., Imam I. & Wechsler H.**, “Face and gesture recognition using hybrid classifiers”, *International Conference on Automatic Face and Gesture Recognition*, Killington, VT, 1996.

**Hecht-Nielsen R.**, “Nearest matched filter classification of spatiotemporal patterns”, Informe Técnico, Hecht-Nielsen Neurocomputer Corporation, San Diego, CA, 1986.

**Hecht-Nielsen R.**, “Counterpropagation networks”, *Applied Optics*, Vol. 26, No. 23, 1987, pp. 4979-4984.

**Hecht-Nielsen R.**, *Neurocomputing*, Addison-Welsey, Reading, MA, 1990.

**Howell J. & Buxton H.**, “Invariance in radial basis function neural networks in human face recognition”, *Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition*, 1995, pp. 221-226.

**Kirby M. & Sirovich L.**, “Aplication of the Karhunen-Loeve procedure for the characterization of human faces”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 1, 1990.

**Kosugi M.**, “Human-face recognition using mosaic pattern and neural networks”, *IEICE Transactions (D-II)*, Vol. J76, No. 6, 1993, pp. 1132-1139.

**Kotropoulos C. & Pitas I.**, “Face authentication based on morphological grid matching”, *Proceedings of the International Conference on Image Processing ICIP'97*, Vol. I, California, 1997, pp. 105-108.

**Lanitis A., Taylor C.J. & Cootes T.F.**, “A unified approach to coding and interpreting face images”, *Proceedings of the International Conference on Computer Vision ICCV'95*, Cambridge, 1995.

**Moghaddam B. & Pentland A.**, “Face recognition using view-based and modular eigenspaces”, *Automatic Systems for the Identification and Inspection of Humans*, 2277, 1994.

**Moghaddam B., Wahid W. & Pentland A.**, “Beyond eigenfaces: Probabilistic matching for face recognition”, *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, Nara, 1998, pp. 30-35.

**Moreno J., Sebastián G., Fernández M.A. & Fernández-Caballero A.,** “A neural architecture for the identification of number sequences”, *Proceedings of the Fifth Brazilian Symposium of Neural Networks SBRN'98*, Belo Horizonte, 1998.

**Moreno J., Sebastián G., Fernández M.A. & Fernández-Caballero A.,** “Comparison of counter-propagation and spatio-temporal nets in the detection of sequences of numbers”, *Proceedings of the Eighth Turkish Symposium on Artificial Intelligence and Neural Networks TAINN'99*, Estambul, 1999.

**Moreno J., Gómez F.J., Fernández M.A. & Fernández-Caballero A.,** “Reconocimiento de rostros utilizando secuencias de histogramas como tramas espacio-temporales”, *Actas del IV Simposio Iberoamericano de Reconocimiento de Patrones SIARP'99*, La Habana, 1999.

**Moreno J., Gómez F.J., Fernández M.A. & Fernández-Caballero A.,** “Reconocimiento facial utilizando redes de contrapropagación”, *Actas del Segundo Encuentro Nacional de Computación ENC'99*, Pachuca, 1999.

**Moreno J., Fernández M.A., Gómez F.J. & Fernández-Caballero A.,** “Face recognition by counter-propagation networks”, *Actas del V Simposio Iberoamericano de Reconocimiento de Patrones SIARP'2000*, Lisboa, 2000.

**Poggio T. & Beymer D.,** Learning networks for face analysis and synthesis”, *International Conference on Automatic Face and Gesture Recognition*, Zurich, 1995.

**Sato K., Shah S. & Aggarwal J.K.,** “Partial face recognition using radial basis function networks”, *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, Nara, 1998, pp. 288-293.

**Tanaka H.T., Ikeda M & Chiaki H.,** “Curvature-based face surface recognition using spherical correlation – Principal directions for curved object recognition”, *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, Nara, 1998, pp. 372-377.

**Turk M. & Pentland A.,** “Eigenfaces for recognition”, *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, 1991.

**Zhao W., Chellappa R. & Krishnaswamy A.,** “Discriminant analysis of principal components for face recognition”, *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, Nara, 1998, pp. 336-341.