

DISEÑO E IMPLEMENTACION DE UN SISTEMA DE SEGURIDAD MEDIANTE RECONOCIMIENTO FACIAL APLICANDO REDES NEURONALES

J. Choque *Student Ricardo Palma University—Perú*
jhonchoque.a@gmail.com

I. ABSTRACT—

This research presents the design and implementation of a security system through facial recognition, image processing techniques, pattern recognition and neural networks. For image processing and pattern recognition we used the selfguiding algorithm also used a red multilayer type neuron. The Arduino platform allows us to establish a communication between the MATLAB software and a final actuator

Palabras Claves— Redes neuronales artificiales, algoritmo de eigenfaces, PCA, Procesamiento de imágenes, entrenamiento supervisado

II. INTRODUCCIÓN

En los últimos años, el desarrollo de nuevo Hardware y Software informático para sistemas de seguridad ha experimentado un gran impulso, tal es el caso de los sistemas de reconocimiento por huella dactilar, voz, iris y facial. Entre estos, destaca por perfilarse como el más prometedor el reconocimiento facial. La identificación de características faciales ha recibido un fuerte impulso gracias al avance en la tecnología de vídeo multimedia propiciándose así un aumento de cámaras en los lugares de trabajo, hogar y dispositivos móviles con un reducido coste.

El reconocimiento facial se puede aplicar en el control de accesos a edificios públicos y privados, cajeros automáticos, laboratorios de investigación, como clave secreta de acceso para el uso de ordenadores personales o terminales móviles de última generación así como para servir de tarjeta de visita de una persona.

El presente trabajo de investigación contempla el diseño e implementación del sistema de reconocimiento para permitir el acceso a un lugar determinado, para lograr esto se utilizará el algoritmo de eigenfaces para poder extraer los vectores característicos de un rostro y utilizarlos como entradas para una red neuronal multicapa, y a través de un sistema embebido arduino establecer la comunicación entre la salida de la red neuronal programada en matlab y un actuador final que tiene la función de permitir el acceso a un determinado lugar.

III. DESARROLLO

SISTEMA DE RECONOCIMIENTO FACIAL

El objetivo de un sistema de reconocimiento facial radica en encontrar y comparar características faciales iguales de una imagen, en una base de datos de imágenes conocidas, mediante un análisis de las características faciales.

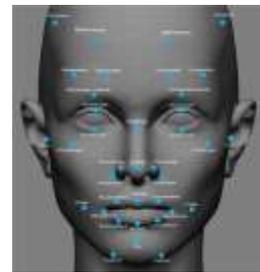


Figura N° 1.Reconocimiento facial

Partes de un sistema de reconocimiento facial:

Todo sistema de reconocimiento facial puede dividirse en tres partes:

- **Preprocesado de las imágenes:** Consiste en compensar todo lo que puede provocar que dos imágenes de la misma cara sean diferentes. Esto incluye normalizar el tamaño y el contraste de la imagen. A veces se intentan compensar los cambios de iluminación, la rotación y otras características de la imagen que pueden perjudicar los resultados del sistema.
- **Extracción de características:** En esta fase se extraen una serie de valores característicos de cada imagen, como pueden ser los coeficientes de algún desarrollo, la salida de un filtro, etc. Independientemente de su origen estos valores deben intentar caracterizar con la mayor exactitud cada cara y, al mismo tiempo, deben tener capacidad de discriminación.
- **Comparación de características:** Se comparan los valores característicos de la imagen de prueba (la que se quiere reconocer) con los de las imágenes de entrenamiento y se calcula una medida de semejanza. La imagen de entrenamiento que más semejante sea a la de test se considerará que es de la misma persona

ANALISIS DE COMPONENTES PRICIPALES (PCA)

Es una técnica estadística de síntesis de la información, o reducción de la dimensión (número de variables). Es decir, ante un banco de datos con muchas variables, el objetivo será reducirlas a un menor número perdiendo la menor cantidad de información posible.

Se parte de una matriz U la cual contiene a los sujetos descritos por (j) y las variables de cada uno de ellos estarán descritas por (i) conformando una matriz de i x j:

$$U = \begin{bmatrix} u_{11} & \dots & u_{1j} \\ \vdots & \ddots & \vdots \\ u_{i1} & \dots & u_{ij} \end{bmatrix} \quad (1)$$

Por tanto esta matriz U se puede transformar en una nueva matriz D de distancias, debido a los puntos que forman los datos de los usuarios en j= dimensional, con las mismas dimensiones i x j.

$$D = \begin{bmatrix} d_{11} & \dots & d_{1j} \\ \vdots & \ddots & \vdots \\ d_{i1} & \dots & d_{ij} \end{bmatrix} \quad (2)$$

Considerando que la matriz D parte de la matriz U, Cada línea de (D) corresponde a los componentes de cada punto de (U). Ahora encontramos que para poder relacionar (U) con (D), se crea una nueva matriz M como matriz de paso de (U), esta matriz será de tamaño j x j con sus componentes de la siguiente manera:

$$\begin{aligned} m_1 &= m_{11}u_1 + \dots + m_{k1}u_k \\ m_2 &= m_{12}u_1 + \dots + m_{k2}u_k \\ &\dots \\ m_k &= m_{1k}u_1 + \dots + m_{kk}u_k \end{aligned} \quad (3)$$

Dando como resultado la matriz M:

$$M = \begin{bmatrix} m_{11} & \dots & m_{1j} \\ \vdots & \ddots & \vdots \\ m_{j1} & \dots & m_{jj} \end{bmatrix} \quad (4)$$

De tal manera encontramos que la relación entre U, D y M es.

$$U = D * M^T \quad (5)$$

ALGORITMO EIGENFACES

Este algoritmo realiza una proyección lineal del espacio de imágenes a un espacio de características de menor dimensión. Esta reducción se realiza utilizando la técnica PCA la cual toma aquella proyección lineal que maximiza la dispersión de todas las imágenes proyectadas.

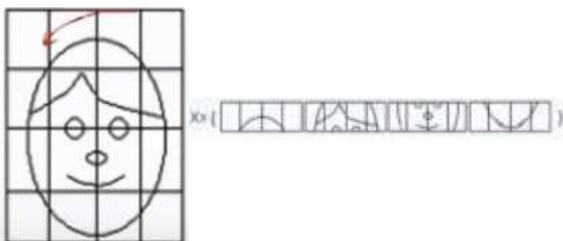


Figura N° 2. Proyección lineal del espacio de la imagen

Para poder aplicar el algoritmo de eigenfaces debemos seguir los siguientes pasos:

1. Tomar imágenes para la base de datos="M" imágenes
2. Cambiar el tamaño de las imágenes (x*y) ;para que todas las muestras tengan la misma dimensión
3. Cambiar imágenes a escala de grises, para que disminuir los valores a analizar
4. Definimos la matriz X, de todas las imágenes (x*y)*M.
5. Se calcula el vector promedio de la matriz X:

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Phi_i \quad (6)$$

6. Normalizamos: Quitamos el promedio hallado en el paso 5 a cada una de las imágenes de la matriz X:

$$\Gamma_i = \Phi_i - \Psi \quad (7)$$

7. Hallar la matriz de covarianza para encontrar los eigenvalores y eigenvectores

$$C((\Phi_i)) = \frac{1}{M} \sum_{i=1}^M \Gamma_i^T \Gamma_i = AA^T \quad (8)$$

$$A^T A v_i = u_i v_i \quad (9)$$

AA^T = Son los eigenvalores de longitud (M x N)

$u_i v_i$ = Son los eigenvectores

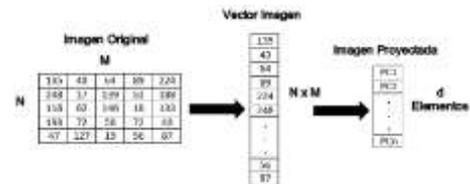


Figura N° 3. Imagen proyectada

PROCESAMIENTO DE IMÁGENES:

Iniciamos con el procesamiento de las imágenes, se tomó como muestra 5 imágenes, las cuales se pasaron a escala de grises y se las redujo para que tengan las mismas dimensiones.



Figura N° 4. Muestras en escala de grises

Hallamos el promedio de las 5 muestras:

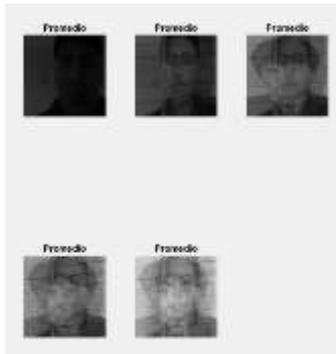


Figura N° 5. Promedio de las muestras

Luego de hallar el promedio tenemos que normalizar las muestras, restando el promedio a cada una de las imágenes:



Figura N° 6. Muestras normalizadas

Hallamos la matriz de covarianza:

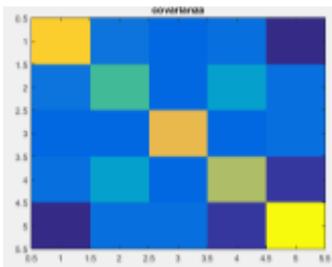


Figura N° 7. Matriz de covarianza

Hallamos los eigenvectores que representan los valores característicos de las cinco muestras:



Figura N° 8. Eigenvectores

	1	2	3	4	5
1	1.7370e+04	2.2683e+03	-2.2003e+03	8.2944e+03	-2.5732e+04
2	1.2178e+04	-8.1039e+03	6.4070e+03	-1.3188e+04	2.7072e+03
3	8.4837e+03	2.0443e+03	-1.7410e+04	-388.1374	7.2705e+03
4	-476.9941	5.8567e+03	56.5670	-4.2543e+03	-1.1819e+03

Figura N° 9. Matriz de los eigenvectores

La matriz de la Figura N°9 representa los vectores característicos de cada imagen los cuales serán usados como entradas para el entrenamiento de la red neuronal artificial

DISEÑO DE LA RED NEURONAL ARTIFICIAL

Antes de empezar con el diseño de la red neuronal debemos tener en consideración que se deben usar la cantidad mínima de neuronas en la capa oculta, la mayoría de problemas se puede resolver utilizando solo una capa oculta, utilizar capas adicionales puede propagar el error haciendo el gradiente más inestable.

Si se utilizan un gran número de neuronas en la capa oculta y no se soluciona el problema satisfactoriamente, entonces se debe usar una segunda capa reduciendo el número de neuronas en cada capa.

Regla de la pirámide geométrica:

Regla aproximada para seleccionar el número de neuronas en las capas ocultas, el número de neuronas sigue una forma piramidal, con un número decreciente de neuronas de la entrada hacia la salida.

Si deseamos diseñar una red neuronal artificial con dos capas ocultas debemos seguir la siguiente fórmula:

$$r = \sqrt[3]{\frac{n}{m}} \quad (10)$$

Donde:

n: es el número de neuronas de entrada
m: es el número de neuronas de salida

Para la primera capa oculta:

$$h_1 = m \times r^2 \quad (11)$$

Donde:

h1: número de neuronas en la primera capa oculta

Para la segunda capa oculta:

$$h_2 = m \times r \quad (12)$$

Donde:

h2: número de neuronas en la segunda capa oculta

CONSTRUCCIÓN DE LA RED NEURONAL ARTIFICIAL

De acuerdo a las fórmulas anteriores podemos definir que nuestra red neuronal va a tener los siguientes parámetros:

Entradas=4

Salida=2

H1=3

H2=2

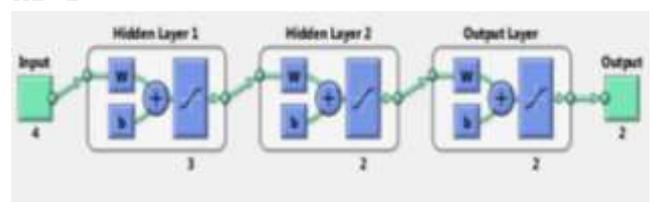


Figura N° 10. Red neuronal artificial

ENTRENAMIENTO DE LA RED NEURONAL ARTIFICIAL

Como es un entrenamiento supervisado debemos tener la salida deseada:

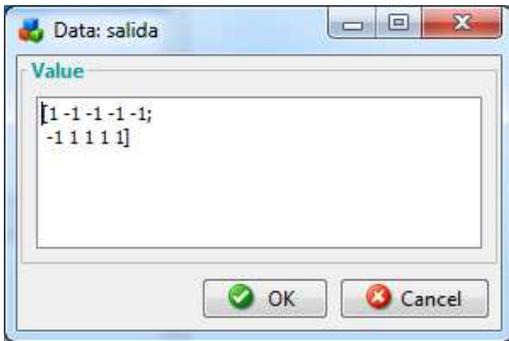


Figura N° 11.Salida deseada

Ingresamos los vectores característicos hallados anteriormente:

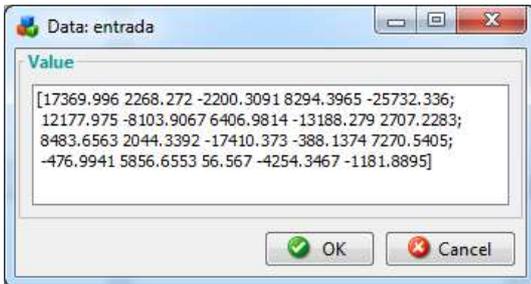


Figura N° 12.Entrada

Para el entrenamiento de la red neuronal se utilizó el método de feed-forward backprop con gradiente descendente.

Luego del entrenamiento se obtuvo la matriz error como se puede apreciar en la Figura N°13 el cual nos muestra un error pequeño lo cual es aceptable.

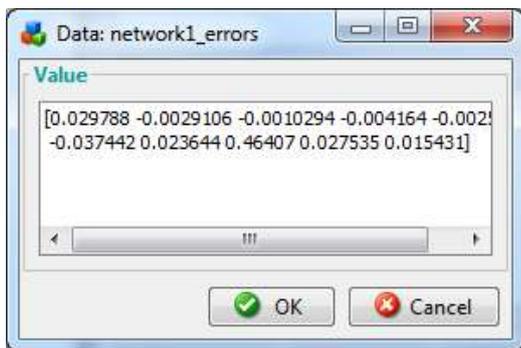


Figura N° 13.Matriz error

Se hizo las pruebas luego del entrenamiento para la validación y se obtuvo la matriz salida la cual es muy parecida para la matriz salida deseada por lo tanto podemos concluir que la red neuronal fue entrenada satisfactoriamente, se puede apreciar la salida en la Figura N°14

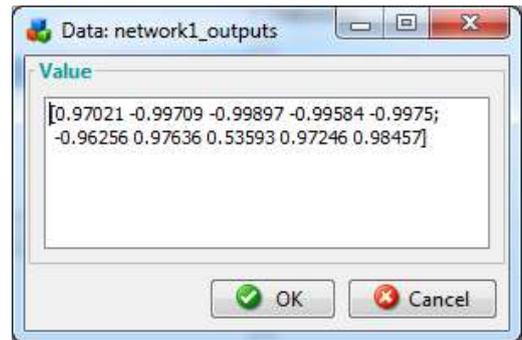


Figura N° 14.Matriz de la salida

RESULTADOS

Interfaz grafica para el ingreso de la entrada y su comprobación con la base de datos.



Figura N° 135.Interfaz gráfica

Como se puede apreciar en la FiguraN°14 el resultado es correcto ya que dicha imagen es la salida deseada .

```

dd=arduino('COM4', 'Uno')
set(handles.edit1, 'visible', 'on')
dd=arduino('COM4', 'Uno')
%%SALIDA
if sal(1,1) > sal(2,1)
    fprintf('La muestra es aceptable');
    writeDigitalPin(dd,13,1)
    mitexto = 'La muestra es aceptable';
    set(handles.edit1, 'string', mitexto);
else
    fprintf('La muestra es no aceptable');
    writeDigitalPin(dd,13,0)
    mitexto = 'La muestra es no aceptable';
    set(handles.edit1, 'string', mitexto);
end
    
```

Figura N° 14.Código para activar el actuador

Con el código de la FiguraN°16 podemos activar un actuador que nos permita el acceso a un lugar determinado en donde solo personas registradas en el banco de muestra puede entrar.

IV. CONCLUSIONES Y RECOMENDACIONES

- Se concluye que el entrenamiento de la red neuronal fue realizada exitosamente ya que pudo reconocer las 5 muestras con éxito
- La red neuronal solo fue entrenada para estas 5 muestras, si una entrada que no está en el banco de datos ingresa, la red actuará de manera impredecible esto se puede evitar aumentando el número de muestras
- Se recomienda el uso de redes neurales multicapa para el reconocimiento de patrones ya que se puede lograr resultados con márgenes de error muy pequeños.
- Se recomienda tener el máximo cuidado en elegir la muestra de entrenamiento ya que de ello depende el éxito del entrenamiento de la RNA.

V. REFERENCIAS

[1] A.Ocsa . Reconocimiento de Rostros mediante Puntos, Simposio Peruano de computación gráfica y procesamiento de imágenes, 2008.

[2]R. Flórez. *Las Redes Neuronales Artificiales: Fundamentos Teóricos y Aplicaciones Prácticas*. España: Netbiblo, 2006.

[3]V. Martínez. *Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones*. Madrid: Alfaomega, 1995.

VI. BIOGRAFÍA



Jhon Choque Alave (Diciembre 1994,13) nació en Tacna, Perú el 13 de diciembre de 1994. Se graduó en el colegio San Martín de Porres; está cursando su pregrado de ingeniería Mecatrónica en la Universidad Ricardo Palma. Áreas de interés :

Email: jhonchoque@gmail.com