

Universidad Autónoma Metropolitana  
Unidad Azcapotzalco

División de Ciencias Básica e Ingeniería

Licenciatura en Ingeniería en  
Computación

Reporte Final del Proyecto Terminal  
Reconocimiento de Huellas Digitales

Asesor: Dr. Oscar Herrera Alcántara

Alumna: García García Elizabeth Adriana

Trimestre 13-P

## Contenido

Resumen.....	3
Objetivos .....	3
Objetivo General .....	3
Objetivos Particulares .....	3
Desarrollo .....	3
Recopilación de Huellas Digitales.....	3
Procesamiento de las imágenes.....	4
Ecuación del histograma de la imagen .....	5
Wavelet .....	6
Binarización .....	7
Dirección.....	8
Adelgazamiento.....	8
Eliminación de puntos aislados y picos .....	9
Extracción de minucias.....	10
Preparación de archivos.....	11
Red neuronal.....	14
Arquitectura .....	14
Método de aprendizaje .....	14
Pesos de inicialización .....	15
Salida .....	15
Introducción de una nueva huella.....	16
CONCLUSION .....	16
Bibliografía .....	19

## Resumen

En el presente documento se describe el proceso para recolectar, procesar y reconocer huellas digitales haciendo uso de wavelets por medio del entrenamiento de una red neuronal, del tipo perceptrón multicapa, y de este modo encontrar una metodología que permita asociar una huella digital con un individuo.

## Objetivos

### Objetivo General

Reconocer huellas digitales con wavelets.

### Objetivos Particulares

- Recolección de huellas digitales.
- Digitalización de huellas digitales.
- Definir características de las huellas digitales.
- Extracción de las características de las imágenes de las huellas digitales.
- Seleccionar un conjunto de entrenamiento de huellas digitales.
- Seleccionar un conjunto de prueba de huellas digitales.
- Presentar a una red neuronal las características de las huellas digitales.
- Entrenamiento de la red neuronal para reconocimiento de huellas digitales.

## Desarrollo

### Recopilación de Huellas Digitales

Se recolectaron huellas digitales de diferentes personas utilizando tinta y cartulina blanca. Hoy en día existen varios métodos para este propósito, el usado en este proyecto es rudimentario y económico pero suficiente para el propósito que se busca. A futuro podría utilizarse otro método de recolección (1).

Se procedió a digitalizar las imágenes con un scanner. Seleccionando las imágenes más claras y nítidas se procedió a trabajar con 23 imágenes de huellas diferentes las cuales pertenecen a 8 personas distintas. Las imágenes se pueden encontrar en los entregables de este proyecto.

Se eligió el formato PGM (Portable Graymap Format)<sup>1</sup> por su facilidad para trabajar una imagen en modo de texto plano.

Para asociar, de manera cómoda a los humanos, cada huella con su propietario se asignó el nombre de archivo de acuerdo a la Tabla 1. Esta tabla permite que en el futuro se pueda comprobar que realmente la imagen corresponde a su dueño.

---

<sup>1</sup>Formato portable de escala de grises. Es un formato que almacena la imagen como archivo de texto, indicando un valor para cada pixel. (7)

Persona	Nombre de imagen de cada huella
Angélica	101.pgm 102.pgm 103.pgm
Beatriz	151.pgm 152.pgm 153.pgm
Cristian	201.pgm 202.pgm 203.pgm
Cristina	251.pgm 252.pgm 253.pgm
Elizabeth	301.pgm 302.pgm 303.pgm
Mario	351.pgm 352.pgm 353.pgm
Marisol	401.pgm 402.pgm 403.pgm
Vicente	451.pgm 452.pgm

Tabla 1. Asignación de nombres de archivos a las imágenes.

Una vez que se obtuvieron y seleccionaron las imágenes se procedió a trabajar con ellas.

### Procesamiento de las imágenes

Por medio de la manipulación de las imágenes se consiguió mejorarlas para obtener las minucias y puntos clave de las huellas.

Este proceso se simplificó gracias al programa "Fingerprint Recognition" (2) al cual se le realizaron modificaciones al código para así poder aplicar un filtro Wavelet.

Para ilustrar el proceso al que se sometió a cada huella se tomará como ejemplo la huella de la Imagen 1.



Imagen 1 "302.pgm"

### Ecuación del histograma de la imagen

Cada imagen tiene un histograma que describe la distribución de los píxeles. Haciendo la ecualización se permite distribuir los valores para aumentar la información perceptiva de la ésta. Ver Imagen 2. El resultado de ecualizar se muestra en la Imagen 3.

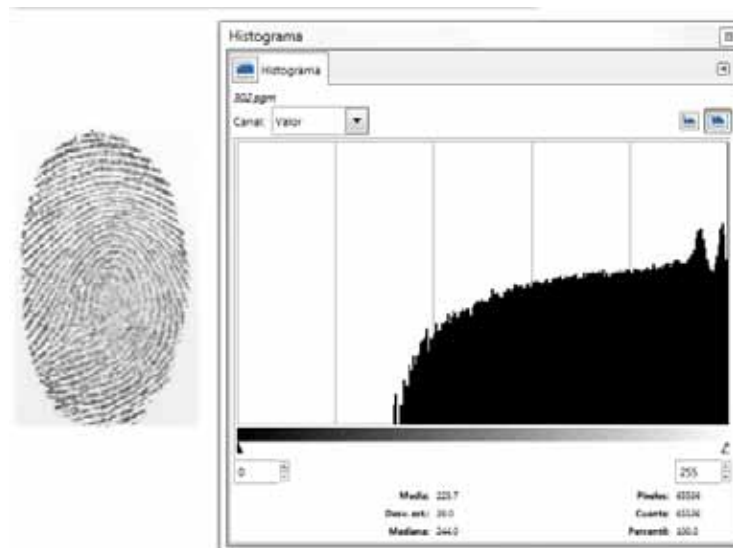


Imagen 2 Histograma de la huella original

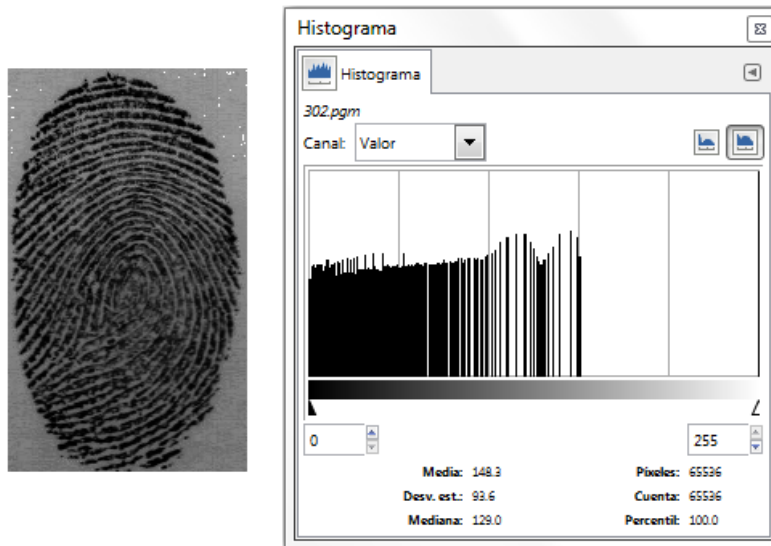


Imagen 3 Ecuación de la imagen original

Como se puede observar la imagen se oscurece esto es porque también se hizo una resta, es decir, al valor 255 se le restó el valor de cada pixel en formato double pero los bordes y minucias de la huella se pueden apreciar mejor.

### Wavelet

El wavelet que se usa es el “Daubechies 30” (3). La imagen se somete a este proceso y el resultado es como el que se muestra en la Imagen 4.



Imagen 4 Después de aplicar el wavelet

Como se puede observar, en este paso “se pierde” información, esto se debe a la naturaleza del wavelet. A continuación se explica este proceso con más detalle.

Las imágenes son arreglos bidimensionales de tamaño NxM que contienen datos numéricos que representan la intensidad de color. Para este proyecto se cuidó que el tamaño de las imágenes fueran de 256x256 píxeles. Se decidió trabajar con este tamaño para poder trabajar con potencias de 2 y aplicar el wavelet sin restricciones de dimensión de la imagen.

Al aplicar el wavelet se generan cuatro cuadrantes:

- HH se compone de los coeficientes que pasaron por el filtro pasa altas.
- LH es el cuadrante que contiene los bordes verticales de la imagen original.
- HL Contiene los bordes horizontales de la imagen original.
- LL este cuadrante se compone por los coeficientes que pasaron por el filtro pasa bajas y representa una versión aproximada de la original con la mitad de resolución. Es la imagen que nos interesa porque ya se ha descartado parte de la información innecesaria.

En la Imagen 5 se pueden apreciar los cuatro cuadrantes de frecuencias que contienen las imágenes (4).

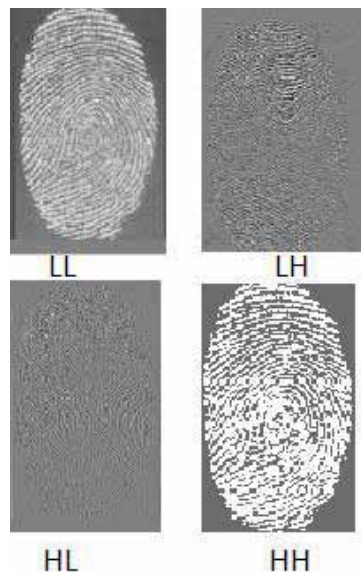


Imagen 5 Aplicación del Wavelet.

Ya que nos quedamos con la imagen filtrada. Se vuelve a aplicar una ecualización de histograma. Y se obtiene algo como lo que se muestra en la Imagen 4.

### Binarización

El siguiente paso es binarizar la imagen para poder identificar contrastes en la imagen que permitan determinar la presencia de minucias.

Ya que la imagen se ha sometido varias veces a modificaciones, ésta pierde cada vez más información tal como se muestra en la imagen.



Imagen 6 Binarización

### Dirección

Se procede a marcar la dirección del gradiente en la imagen. Se obtiene algo como en la Imagen 7.



Imagen 7 Dirección del gradiente.

### Adelgazamiento

El siguiente paso es adelgazar los bordes de la imagen para poder detectar más fácilmente la mayor cantidad de minucias. Este paso se realiza hasta este punto porque ya se tienen las direcciones de los bordes. Esto se puede observar en la Imagen 8.



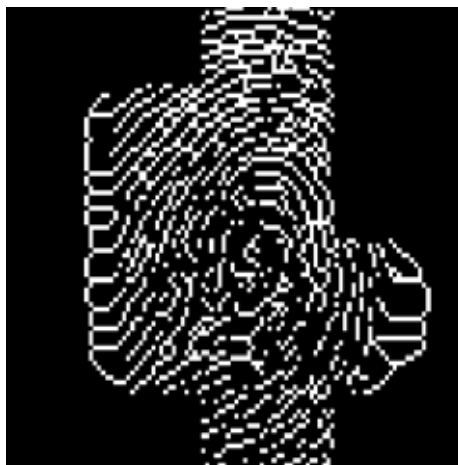


Imagen 8 Adelgazamiento de los bordes

### Eliminación de puntos aislados y picos

Los puntos aislados y picos no aportan demasiada información a la huella pero si pueden causar errores al calcular las minucias.

En la Imagen 9, de lado izquierdo, se muestran algunas de esos puntos resaltados con un círculo amarillo, en el lado derecho se puede apreciar que ya no se encuentran presentes en la imagen que se procesa.



Imagen 9 Eliminación de islas

En la imagen se puede apreciar que de lado derecho se encuentra la imagen con los puntos aislados eliminados pero esta vez se resaltan los picos, de lado derecho se encuentra la imagen limpia de “impurezas”.



Imagen 10 Eliminación de picos y resultado final.

### Extracción de minucias

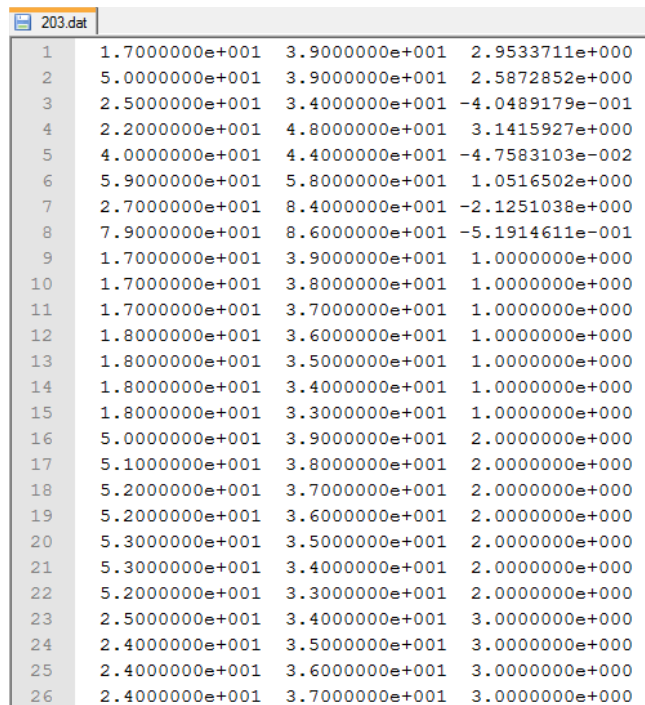
Ya con todo esto se pueden ver las minucias que se detectan en la imagen final. El resultado se puede apreciar en la Imagen 11. Se genera un archivo de texto almacenado con extensión “\*.dat”



Imagen 11 Minucias

El programa permite comparar los archivos “\*.dat”. Sin embargo se decidió que el conjunto de los archivos con las minucias de cada huella se presentarían, todas a la vez, a una red neuronal para entrenarla y verificar que se realiza correctamente la identificación.

En la Imagen 12 se muestra parte del archivo que se genera en este paso. Es importante observar que se generan varias líneas y, a la vez, cada renglón contiene tres valores en notación científica. El número de líneas es variable para cada huella, el archivo con más líneas contiene 279; para el caso de la imagen “302” en total resultaron 64 líneas.



Linea	Valor 1	Valor 2	Valor 3	Valor 4
1	1.7000000e+001	3.9000000e+001	2.9533711e+000	
2	5.0000000e+001	3.9000000e+001	2.5872852e+000	
3	2.5000000e+001	3.4000000e+001	-4.0489179e-001	
4	2.2000000e+001	4.8000000e+001	3.1415927e+000	
5	4.0000000e+001	4.4000000e+001	-4.7583103e-002	
6	5.9000000e+001	5.8000000e+001	1.0516502e+000	
7	2.7000000e+001	8.4000000e+001	-2.1251038e+000	
8	7.9000000e+001	8.6000000e+001	-5.1914611e-001	
9	1.7000000e+001	3.9000000e+001	1.0000000e+000	
10	1.7000000e+001	3.8000000e+001	1.0000000e+000	
11	1.7000000e+001	3.7000000e+001	1.0000000e+000	
12	1.8000000e+001	3.6000000e+001	1.0000000e+000	
13	1.8000000e+001	3.5000000e+001	1.0000000e+000	
14	1.8000000e+001	3.4000000e+001	1.0000000e+000	
15	1.8000000e+001	3.3000000e+001	1.0000000e+000	
16	5.0000000e+001	3.9000000e+001	2.0000000e+000	
17	5.1000000e+001	3.8000000e+001	2.0000000e+000	
18	5.2000000e+001	3.7000000e+001	2.0000000e+000	
19	5.2000000e+001	3.6000000e+001	2.0000000e+000	
20	5.3000000e+001	3.5000000e+001	2.0000000e+000	
21	5.3000000e+001	3.4000000e+001	2.0000000e+000	
22	5.2000000e+001	3.3000000e+001	2.0000000e+000	
23	2.5000000e+001	3.4000000e+001	3.0000000e+000	
24	2.4000000e+001	3.5000000e+001	3.0000000e+000	
25	2.4000000e+001	3.6000000e+001	3.0000000e+000	
26	2.4000000e+001	3.7000000e+001	3.0000000e+000	

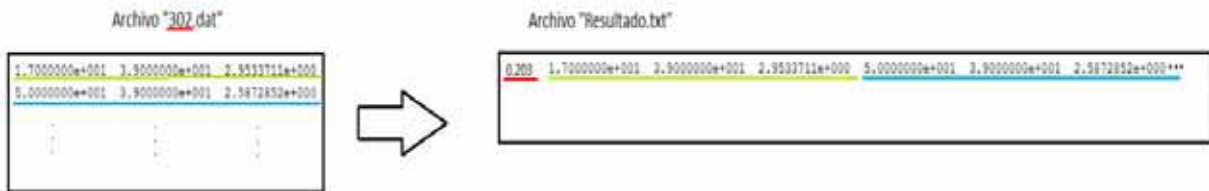
Imagen 12 Parte del archivo generado.

## Preparación de archivos

Para poder presentar todas las huellas a la red neuronal, es necesario que todos los archivos \*.dat se guarden en un único archivo de texto plano. Para ello se creó un programa que convierte a cada archivo en una línea del archivo que se presentará a la red neuronal. El código fuente de ese programa se presenta en el Anexo .

Este programa recibe como parámetros una etiqueta, el nombre del archivo a procesar y el nombre del archivo resultado. La etiqueta permite identificar la línea y por ende el archivo al que pertenece, por ejemplo, si nuestro archivo es el “302.dat” la línea debemos poder identificarla,

para ello el número de huella se divide entre 100 y así se obtiene un valor decimal que nos permitirá cumplir con esa tarea. De esta manera la información del archivo "302.dat" podrá ser identificado dentro del archivo que será presentado a la red neuronal en un entrenamiento supervisado. Este paso se ilustra en la Imagen 13.



**Imagen 13** Proceso de creación del archivo "Resultado.txt"

Cada vez que se ejecuta el programa, en caso de que el archivo de salida sea el mismo, éste simplemente agrega la nueva línea al final. Esto permitirá que en el futuro se agreguen más huellas sin sobrescribir o eliminar las ya existentes.

Para facilitar el manejo de todos los archivos se hizo uso de un script, mismo que se puede consultar en el

Anexo .

Al finalizar este proceso se obtuvo el archivo que se presentó a la red neuronal. Sin embargo se presentó un inconveniente, había renglones llenos de ceros porque la huella era de mala calidad y prácticamente no se determinaron minucias. Ese problema se solucionó eliminando esos renglones y al finalizar el archivo quedó con la información de 19 huellas digitales.

El archivo final es muy grande, se encuentra en los archivos que se entregan con este proyecto.

Con el 100% de los datos con los que se cuenta se hizo una sencilla operación con los datos: el 80% de ellos se usaron para entrenamiento de la red (de los 19 archivos válidos se usaron 15) y el 20% se ocupó para prueba. Obsérvese la Imagen 14.

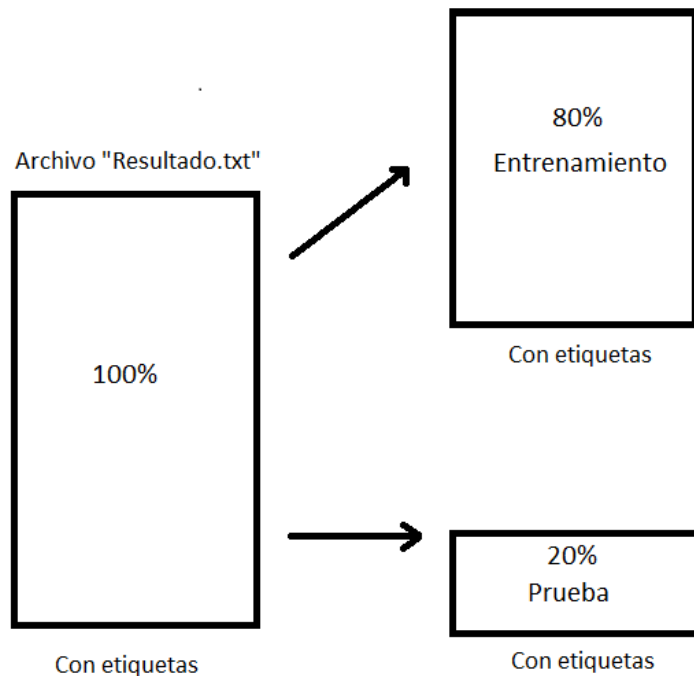


Imagen 14 Proporción de datos de entrenamiento y prueba.

## Red neuronal

### Arquitectura

Con el archivo que contiene la información de las huellas digitales se procede a trabajar en la red neuronal. Se utilizó un Perceptrón Multicapa (MLP, Multilayer Perseptron) (5) (6). La Imagen 15 muestra, de manera general, la arquitectura.

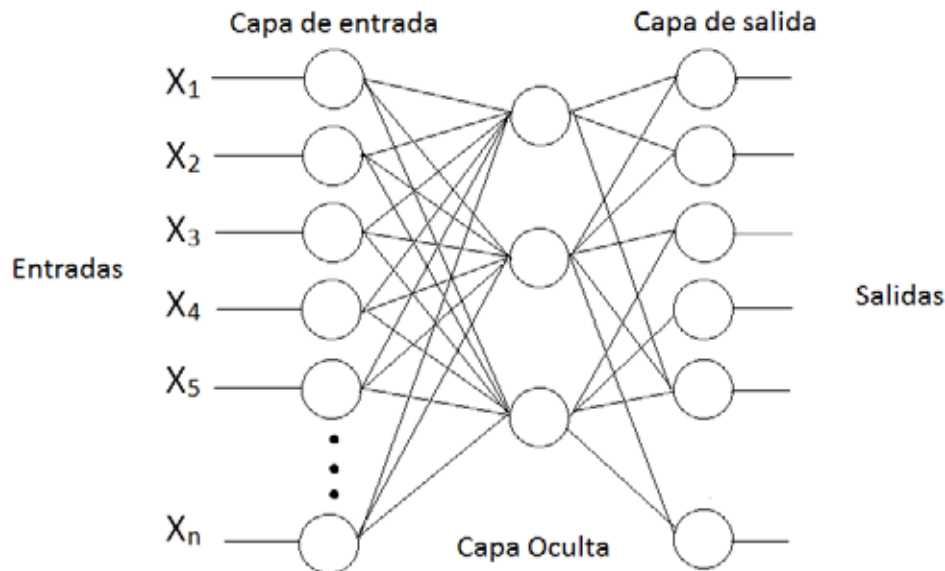


Imagen 15 Arquitectura del perceptrón multicapa

Cabe aclarar que el entrenamiento es supervisado, ya que de esta manera se podrá comprobar el resultado.

### Capa de entrada

Para la capa de entrada se utilizaron 257 entradas, es decir, 257 neuronas las cuales corresponden al número máximo de columnas con las que cuenta el archivo "resultado.txt" más la columna de etiquetas. Se utilizó una función lineal como función de transferencia.

### Capa oculta

Para esta capa se utilizaron 4 neuronas y una función tangencial de transferencia. Este número es el que mejor permitió que la red entrenara con el mínimo de error.

### Capa de salida

Para la capa de salida se utilizó una sola neurona, cuando se intentó utilizar más neuronas la red presentaba un error considerable.

### Método de aprendizaje

Como método de aprendizaje se utilizó retropropagación .

Los datos fueron presentados en forma aleatoria.

### Parámetros de aprendizaje

Los parámetros empleados, tanto para la capa de entrada como para la de salida de la red neuronal fueron los que se listan a continuación.

- Tasa de aprendizaje: 0.1
- Momento: 0.1
- *weight decay*: 0.999

### Pesos de inicialización

Se utilizaron los siguientes pesos de inicio:

- Límite inferior: -0.1
- Límite superior: 0.1

### Salida

Como valores finales la red neuronal entregó los siguientes resultados:

- Numero de épocas: 2000
- Error máximo de aprendizaje:  $2.2204460 \times 10^{-16}$ .
- RMS (Raíz Cuadrada Media) del error de entrenamiento:  $8.0115312 \times 10^{-17}$ .
- Máximo error de prueba: 0.715666326587.
- RMS (Raíz Cuadrada Media) del error de prueba: 0.501023037151

Para probar el funcionamiento de la red neuronal se debe introducir un archivo "Recall" de entrada, el cual contiene el 100% de las huellas sin etiquetar. La red entregará un archivo "Recall" de salida y por sí misma entregará el mismo archivo pero con las huellas etiquetadas, esto porque previamente se hizo la fase de entrenamiento.

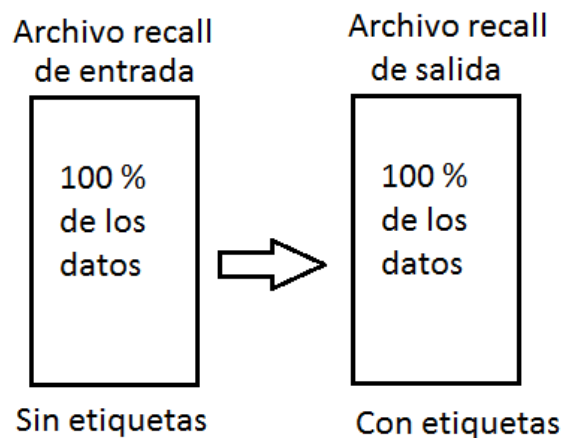


Imagen 16 Archivos "Recall"

## **Introducción de una nueva huella**

Para este paso se eligió la misma huella 302.pgm y se escaló 5% y rotó 20° para hacerla pasar por el mismo proceso descrito. Se observó que la red entregó la etiqueta 0.204. Lo que nos permite observar que la red etiquetó a esta huella erróneamente.

## **CONCLUSIÓN**

Después de un largo proceso al que se debe someter una huella digital se debe ser muy cuidadoso a la hora de elegir un método. El descrito en este trabajo permite observar que uno de los principales inconvenientes es la calidad de la imagen. Es probable que si se aumenta la resolución de las imágenes el resultado sea más aproximado al deseado.

Igualmente es necesario probar con un número mayor de huellas digitales este proceso.



## Anexo 1 Código fuente del programa que permite ordenar los archivos a presentar a la red neuronal.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX 255 // es el tamaño máximo tendrán los renglones, si no se completa los llena
con ceros, el mayor es de 279; 279*3=837
int main(int argc, char ** argv) //PARAMETROS: ETIQUETA ARCHIVOAPROCESAR ARCHIVORESULTADO
{
    float etiqueta=0.0;
    int contador=0,j=0;
    char Valor1[15]="\0";
    float Valor2=2.5f;
    FILE * ArchivoAProcesar=NULL; //el archivo que voy a procesar
    FILE * ArchivoResultado=NULL; //quiero escribir en el final del archivo una linea
completa con su respectiva etiqueta
    etiqueta=atof(argv[1]);
    //printf("\nEtiqueta: %f", etiqueta);

    ArchivoAProcesar=fopen(argv[2],"r"); printf("\n¡Archivo origen abierto! \n");
    ArchivoResultado=fopen(argv[3],"r+b"); printf("¡Archivo destino abierto! \n");

    if (ArchivoAProcesar == NULL)
    {
        printf("\nError de apertura del archivo a procesar. \n\n");
    }

    if (ArchivoResultado == NULL)
    {
        printf("\nError de apertura del archivo resultado. \n\n");
    }

    else
    {
        fseek(ArchivoResultado, 0, SEEK_END); //aquí hay que colocarse en el último
renglón para escribir el nuevo renglón
        fprintf(ArchivoResultado, "%.3f ", etiqueta);
        while (! feof(ArchivoAProcesar))
        {
            fscanf(ArchivoAProcesar, "%s ", Valor1);
            //printf("%s \n", Valor1);
            Valor2=atof(Valor1);
            if (contador<MAX)
            {
                fprintf(ArchivoResultado, "%.3f ", Valor2);
            }
            contador=contador+1;
        }
        if(contador<MAX) //En caso de tener que llenar el renglón con ceros
            for(j=contador;j<MAX;j++)
                fprintf(ArchivoResultado, "0.0 ");
            printf("\nOK\n");
        }

        fprintf(ArchivoResultado, "\n");
        fclose ( ArchivoAProcesar );
        fclose ( ArchivoResultado );
        return 0;
    }
}
```

## Anexo 2 Script

```
#!/bin/bash/  
  
# Este script me servirá para procesar todos los archivos.dat y enviarlos al archivo de texto para  
#ser llevados a la red neuronal  
  
./main3 0.101 101.dat salida.txt  
./main3 0.102 102.dat salida.txt  
./main3 0.103 103.dat salida.txt  
./main3 0.151 151.dat salida.txt  
./main3 0.152 152.dat salida.txt  
./main3 0.153 153.dat salida.txt  
./main3 0.201 201.dat salida.txt  
./main3 0.202 202.dat salida.txt  
./main3 0.203 203.dat salida.txt  
./main3 0.251 251.dat salida.txt  
./main3 0.252 252.dat salida.txt  
./main3 0.253 253.dat salida.txt  
./main3 0.301 301.dat salida.txt  
./main3 0.302 302.dat salida.txt  
./main3 0.303 303.dat salida.txt  
./main3 0.351 351.dat salida.txt  
./main3 0.352 352.dat salida.txt  
./main3 0.353 353.dat salida.txt  
./main3 0.401 401.dat salida.txt  
./main3 0.402 402.dat salida.txt  
./main3 0.403 403.dat salida.txt  
./main3 0.451 451.dat salida.txt  
./main3 0.452 452.dat salida.txt
```

## Bibliografía

1. **GARCIA, Victor.** Sistema de Reconocimiento de Huellas digitales para el Control de Acceso a Recintos. [En línea] 5 de Noviembre de 2008. <http://hdl.handle.net/123456789/1290>.
2. **WUZHILI.** comp.hkbu.edu.hk. [En línea]  
<http://www.comp.hkbu.edu.hk/~vincent/hp/fingerprintRecognition.doc>.
3. **DAUBECHIES, Ingrid.** <http://wavelets.pybytes.com/wavelet/db15/>. [En línea]  
<http://wavelets.pybytes.com/wavelet/db15/>.
4. **Ritter, Jörg.** [En línea] Abril de 1971. <http://sundoc.bibliothek.uni-halle.de/diss-online/02/03H033/prom.pdf>.
5. **MINSKY, Marvin.** *Perceptrons : an introduction to computational geometry*. Cambridge, Mass : MIT Press , 1972.
6. **HAYKIN, Simon S.** *Neural networks and learning machines*. New York, NY : Pearson Education, , 2009.
7. **Poskanzer, Jef.** pgm. [En línea] Agosto de 2013. <http://netpbm.sourceforge.net/doc/pgm.html>.
8. wavelet.org. [En línea] <http://www.wavelet.org/>.