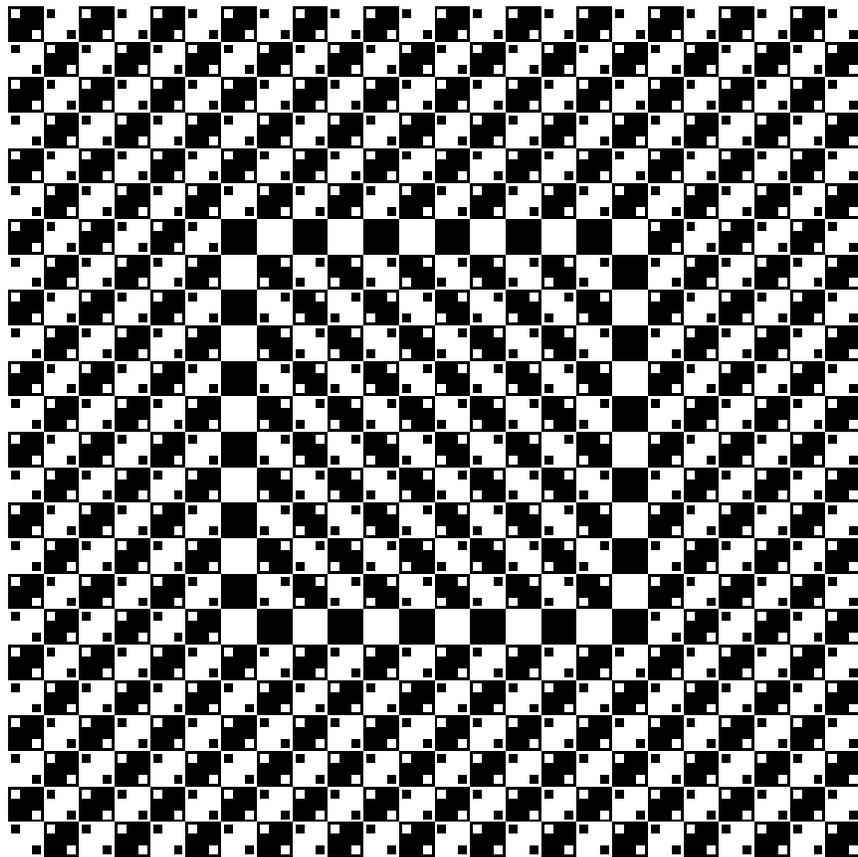


5° INGENIERÍA DE TELECOMUNICACIÓN **INTELIGENCIA ARTIFICIAL Y RECONOCIMIENTO DE PATRONES**

Práctica 11: **Desarrollo de un control de acceso. Parte II: reconocimiento.**

Objetivos:

- Desarrollar la etapa de reconocimiento de un control de acceso.
 - Probar los resultados obtenidos por un reconocedor de caras simple.
-



1. RESULTADOS PREVIOS: FASE DE ENTRENAMIENTO.

De acuerdo con lo explicado en la práctica previa, un sistema de control de acceso consta de dos fases: entrenamiento y reconocimiento. Esta práctica se centra en la segunda de las fases.

En la práctica anterior se determinó el método óptimo de entrenamiento, llegándose a la conclusión de que el mejor clasificador para esta aplicación es el vecino más cercano y el número de componentes de PCA a utilizar debe ser lo más elevado posible (9 en este ejemplo).

De acuerdo con esos resultados, se comenzará la práctica realizando el entrenamiento en las condiciones descritas anteriormente y almacenando el modelo generado, para que pueda ser usado posteriormente. El siguiente programa de Matlab realiza este proceso:

```
% control de acceso basado en reconocimiento de caras
% primera fase: entrenamiento
function acceso_entrena

% elige escenario y numero de componentes
escribe_arff('caras.arff', 'datos.mat');

% lanza el clasificador elegido y guarda el modelo
!java weka.classifiers.lazy.IBk -t caras.arff -d mod.out

return
```

Algunos aspectos a tener en cuenta:

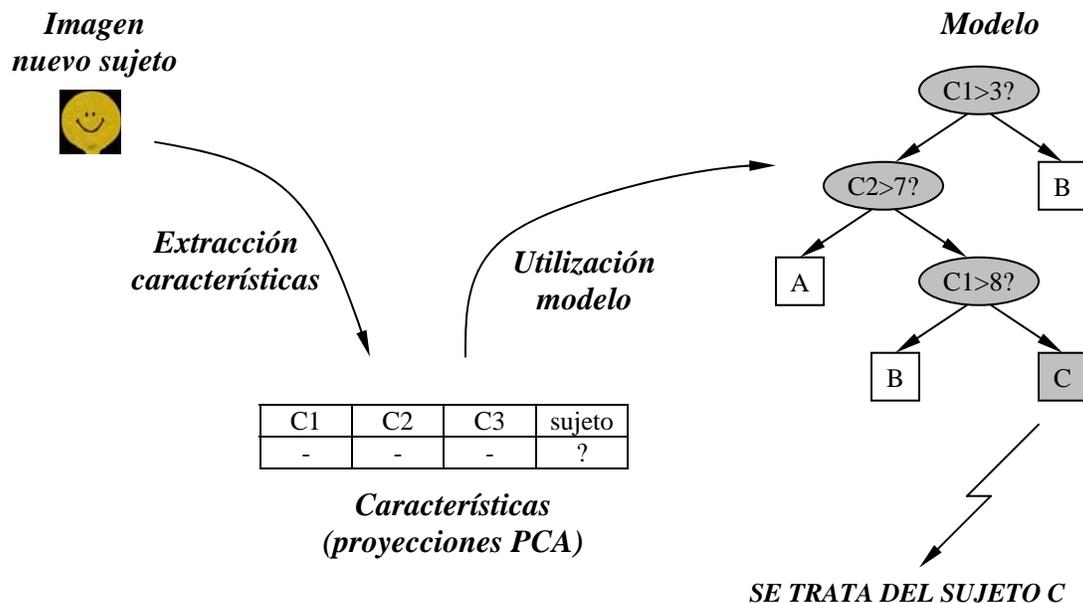
- En la instrucción `escribe_arff` se utiliza el fichero de Matlab `datos.mat`, que contiene las proyecciones sobre un espacio de 9 componentes (el que ha ofrecido mejores resultados).
- En la instrucción de llamada a WEKA puede verse cómo el modelo se almacena en el fichero `mod.out`.

El programa, las funciones a las que llama, y el fichero de datos están accesibles desde la página web de la asignatura:

http://lorca.umh.es/isa/es/asignaturas/iarp/practicas/P11/acceso_entrena.zip

2. SEGUNDA FASE: RECONOCIMIENTO.

En esta fase se capta la imagen de un sujeto que supuestamente desea acceder a un edificio, y se intenta determinar su identidad. Para ello, primero se extraen la características de la imagen (proyectándola sobre el mismo espacio generado con las imágenes de entrenamiento) y se aplica el modelo también generado anteriormente para determinar qué clasificación se obtiene (a qué sujeto se parece más la imagen).



El programa de reconocimiento tiene que realizar, por tanto, las siguientes operaciones:

- Capturar una imagen.
- Recortar a tamaño estándar (al igual que en el entrenamiento).
- Proyectar sobre un subespacio PCA de 9 componentes.
- Crear un fichero WEKA (formato arff) con las proyecciones.
- Lanzar WEKA con ese fichero y el modelo almacenado anteriormente.
- Leer el resultado de WEKA.
- Mostrar el resultado en pantalla.

Todas esas operaciones se llevan a cabo mediante un programa de Matlab que se ejecuta cíclicamente, de forma que se realiza de modo continuo el control de acceso:

```

% control de acceso basado en reconocimiento de caras
% segunda fase: verificacion
function acceso_verifica

while(1)

    % lee la ultima imagen capturada
    imagen = espera_imagen('Image001.bmp');

    % recorta la imagen al tamaño estandar
    recorte = recorta_imagen(imagen, 120, 150);

    % obtiene las componentes PCA proyectando sobre el subespacio
elegido
    proy = proyecta_PCA(recorte);

    % escribe el fichero de test para WEKA
    escribe_test_arff('test.arff', proy);

    % lanza WEKA
    !java weka.classifiers.lazy.IBk -T test.arff -l mod.out -p 0
> out.txt

    % lee resultado de WEKA
    sujeto = lee_resultado('out.txt');

    % muestra resultado en pantalla
    muestra_resultado(sujeto);

end

return

```

Este programa y todas las funciones a las que se llaman desde él están disponibles en la página web de la asignatura:

http://lorca.umh.es/isa/es/asignaturas/iarp/practicas/P11/acceso_verifica.zip

Deberán descargarse para poder realizar la práctica.

3. FORMA DE UTILIZACIÓN DEL PROGRAMA.

El programa se debe utilizar de la forma siguiente: en primer lugar, se deben lanzar tanto Matlab como la aplicación de captura de imágenes mediante la Webcam (PC-CAM Center Lite), de modo que se reparta el espacio de pantalla entre las dos aplicaciones, tal y como muestra la figura siguiente:

El objetivo de la práctica es realizar pruebas y determinar la fiabilidad de los resultados. En principio, existen múltiples fuentes de error posibles, desde diferencias de escalado, iluminación, fondos, etc. Por eso no es de extrañar que los porcentajes de reconocimientos correctos resulten bajos.

El mensaje que ofrece Matlab indicando cuál es el sujeto reconocido sólo indica un número. Para conocer la correspondencia entre los números y los distintos alumnos puede utilizarse la guía siguiente:

