

# SISTEMAS DE PERCEPCIÓN

Ejercicio práctico 4: Reconocimiento de objetos

Álvaro Calvo Matos  
Damián Jesús Pérez Morales

# Índice

1. Introducción .....	2
2. Imágenes de entrenamiento .....	3
3. Identificación de las matrículas .....	5
4. Clasificador .....	7
5. Código main.c .....	9
6. Conclusiones .....	12

# 1. INTRODUCCIÓN

En el cuarto ejercicio práctico de la asignatura, se pretende conseguir reconocer los dígitos contenidos dentro de una matrícula alemana, generada gracias a la página web <https://www.europeancustomplates.com/>, mencionada en el enunciado del ejercicio. Para ello, se recurrirán a técnicas adquiridas en clase para hacer el etiquetado de la imagen de partida, eliminar aquellas etiquetas que no sean de interés (todo aquello que no sean los dígitos de la matrícula) y poder operar directamente con una imagen en la que solo se contengan los números de la matrícula:

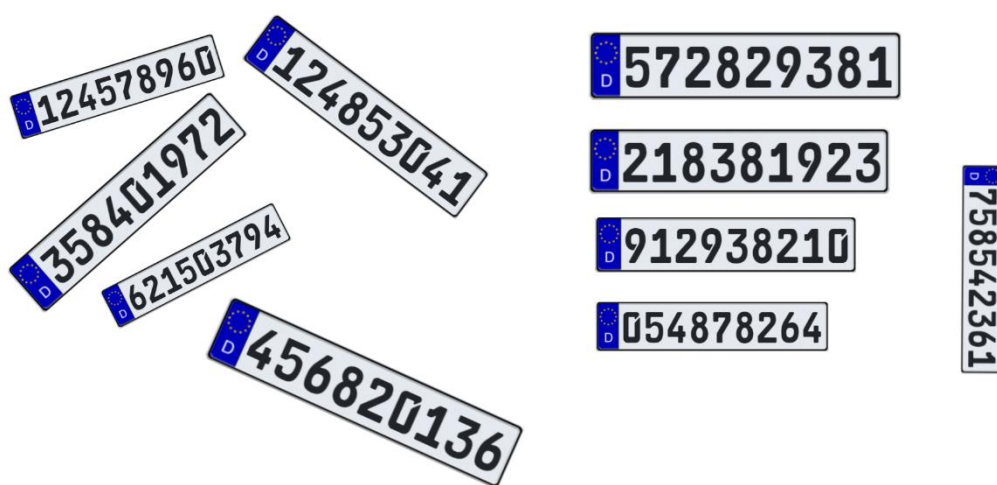


Ilustración 1. Imagen a identificar

Se empleará un subconjunto de momentos de Hu de los dígitos como características de cada clase y, para su reconocimiento, se utilizará el discriminante de mínima distancia.

La obtención de un Clasificador de elementos, en este caso, de números, se puede dividir en tres piezas con diferentes misiones:

**Constructor/Entrenador:** Usa un conjunto de muestras proporcionadas, cuya clase se conoce, con la que generará una matriz de prototipos a partir de las medias de cada clase. Este proceso se considera el entrenamiento del clasificador, y su resultado es un patrón representante de cada clase.

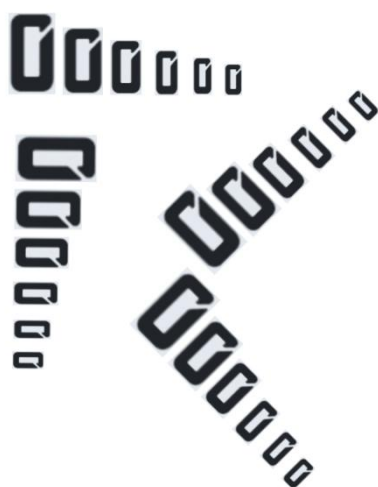
**Validador:** Comprueba el funcionamiento del clasificador entrenado. Mediante la matriz de prototipos obtenida se clasifican elementos cuya clase es conocida. Estos pueden ser los mismos usados para la construcción, o un conjunto mayor de muestras conocidas. El resultado es un porcentaje de fallos o clasificaciones incorrectas (PCI) que indica la fiabilidad del clasificador.

**Explotación:** Uso del clasificador entrenado y validado con muestras cuya clase se desconoce. El resultado será la asignación de una clase a cada elemento, y la confianza en esta asignación será inversamente proporcional al PCI obtenido en el paso de validación.

## 2. IMÁGENES DE ENTRENAMIENTO

Para poder realizar la identificación de dígitos, es necesario hacer el entrenamiento de un clasificador para dichos dígitos, cogiendo algunos patrones de cada uno de ellos. Para generar los patrones que se le pasarán al constructor se ha ido haciéndolos variar de escala y de rotación, ya que así se consigue obtener un baremo más diferido de características que identifiquen al dígito, un conjunto de representantes más variado. Cabe destacar que **no se ha mantenido en ninguno de los dígitos la relación de aspectos** original, con el objetivo de intentar reconocer los números si la imagen a identificar (las matrículas) no mantenía tampoco la relación de aspectos, por lo que **se espera que en algún caso no se reconozca correctamente algún dígito**.

Se expone a continuación algunas de las imágenes de entrenamiento:



*Ilustración 2. Imagen de entrenamiento para el dígito 0*



*Ilustración 3. Imagen de entrenamiento para el dígito 6*

Con estas imágenes se pretende sacar una plantilla para cada dígito, en el que se reconozcan los números por separado para, posteriormente, etiquetarlos. Seguidamente, se obtienen los siete momentos de Hu para cada uno de los patrones (cabe tener en cuenta que en cada imagen de entrenamiento hay 24 patrones).

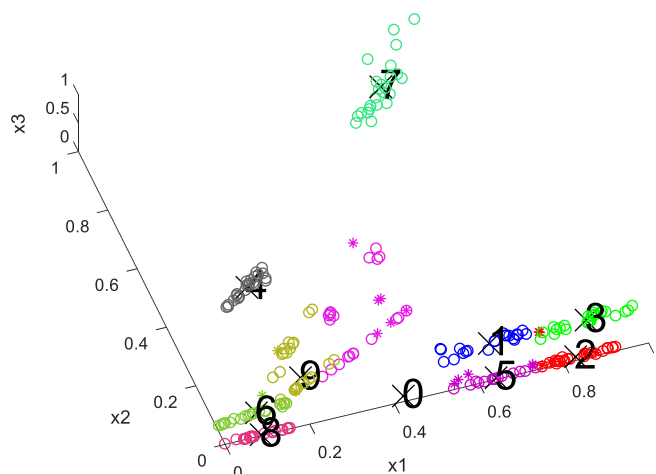
Obtenidos los momentos de Hu para cada patrón, y dispuestos en una matriz junto con la información de la clase a la que pertenecen, se hacen las medias por clase finalizando con el proceso de construcción del clasificador.

Nota: Para que cada uno de los momentos de Hu tenga el mismo peso a la hora del reconocimiento del dígito, ha sido necesario escalar cada uno de los momentos en un rango de 0 a 1.

Hasta este punto se ha estado trabajando con características extra, el siguiente paso, intermedio entre la construcción y la validación, será el de decidir qué subconjunto de características se usarán en el clasificador final, reduciendo la matriz de patrones de 7 a 4 filas.

Tras varias pruebas, se ha corroborado que **la mejor combinación de características para identificar los patrones de entrenamiento son los momentos de Hu 1, 3, 4 y 6** y serán los que se empleen para identificar los dígitos de la imagen de matrículas. Se ha obtenido con esto un PCI del 8'75%, lo cual es admisible, además teniendo en cuenta que se hizo adrede una deformación de los dígitos en las imágenes de entrenamiento por si en la imagen de las matrículas los dígitos no mantenían la relación de aspectos original por cualquier motivo errático a la hora de generar la imagen (cabe mencionar que dicha imagen se ha generado por computador y se han alterado las relaciones de aspectos para poner a prueba la eficacia de este método).

Cabe destacar que el PCI mencionado se ha obtenido tras el paso de validación. Para validar el clasificador construido se ha hecho uso del mismo conjunto de entrenamiento, no se ha generado uno nuevo. El motivo por el que el mismo conjunto de patrones no da un porcentaje de acierto del 100% es no haber conservado la relación de aspecto y quizás haber reducido en exceso el tamaño de los dígitos en las imágenes de entrenamiento, resultando en unos momentos de Hu menos precisos dada la pérdida de resolución en la imagen.



*Ilustración 4. Plantilla de matrículas completa etiquetada.*

### 3. IDENTIFICACIÓN DE LAS MATRÍCULAS

Para pasar a la explotación del clasificador, es necesario tratar la imagen objetivo, ya que esta contiene elementos que no se desean identificar pero que, al hacer la plantilla y etiquetado, se mostrarían.

Primero binarizamos la imagen a color mostrada en la ilustración 1, necesario para poder realizar el etiquetado. Notar que el binarizado se ha invertido para que resulten en color blanco los elementos de interés.

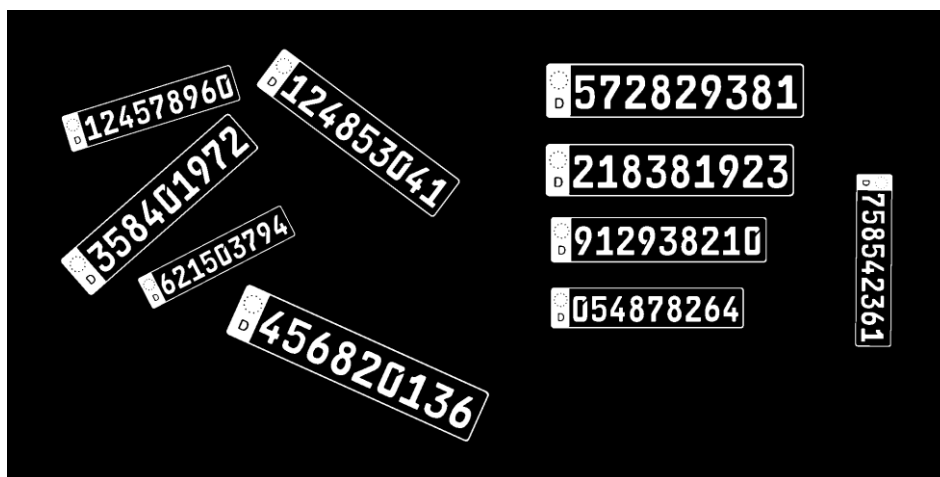


Ilustración 5. Plantilla de matrículas completa

Haciendo el primer etiquetado de la imagen a identificar, resulta:



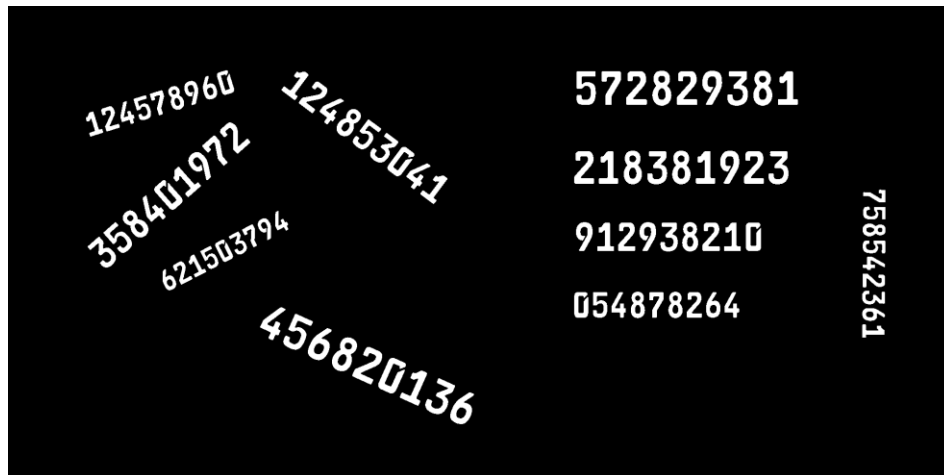
Ilustración 6. Plantilla de matrículas completa etiquetada.

Dado que contiene información irrelevante, se tiene que saber identificar de algún modo el marco de la matrícula y la *D* contenida en ella para poder eliminarla. Para ello, se ha extraído de la imagen etiquetada la información de los semiejes de la elipse de inercia equivalente que describe a cada etiqueta, con misión de obtener de esta forma las dimensiones del semieje de máxima inercia y del semieje de mínima inercia, a modo de describir el área de la etiqueta en

cuestión. Además, se extrae también la información de la masa de cada etiqueta, es decir, la cantidad de píxeles contenidos en cada etiqueta.

Se establece así una relación entre área y masa que, experimentalmente, se pueden obtener valores para la  $D$  y para el borde de la matrícula, que se emplearán posteriormente para la discriminación de estas etiquetas.

Aplicando esta técnica de discriminación, se obtiene la siguiente plantilla:



*Ilustración 7. Plantilla de la matrícula con solo dígitos.*

## 4. CLASIFICADOR

Una vez obtenida la plantilla filtrada de la imagen a identificar, es necesario etiquetar la imagen y calcular de cada uno de los elementos etiquetados los momentos de Hu escalados que se usan para la clasificación.

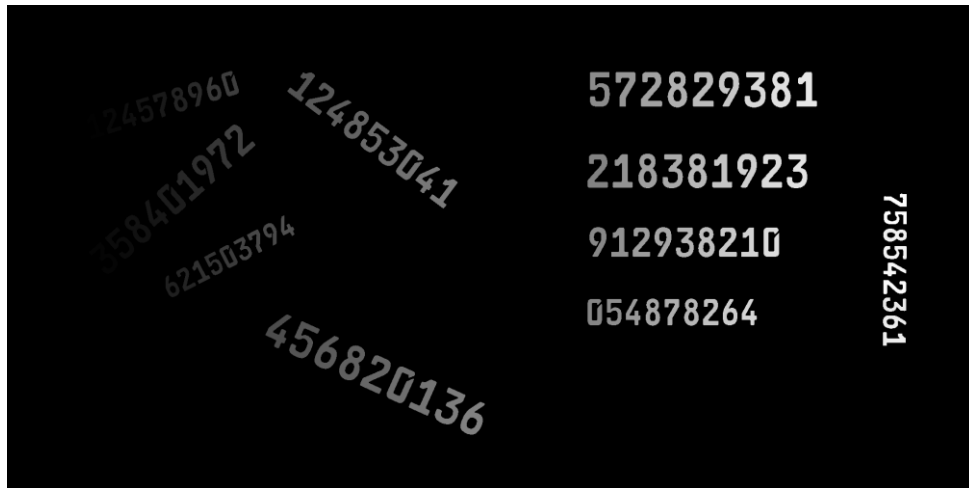


Ilustración 8. Plantilla de matrículas filtrada y etiquetada.

Después, será necesario relacionar las características obtenidas para cada etiqueta con una clase y se empleará para ello el discriminante de mínima distancia. Como se mencionó en el apartado de *Entrenamiento*, se emplearon como características los momentos de Hu 1, 3, 4 y 6, por lo que se deben emplear también dichas características para que todo funcione correctamente.

A continuación, se expone una imagen en la que se representan los dígitos que reconoce el clasificador sobre la imagen original.

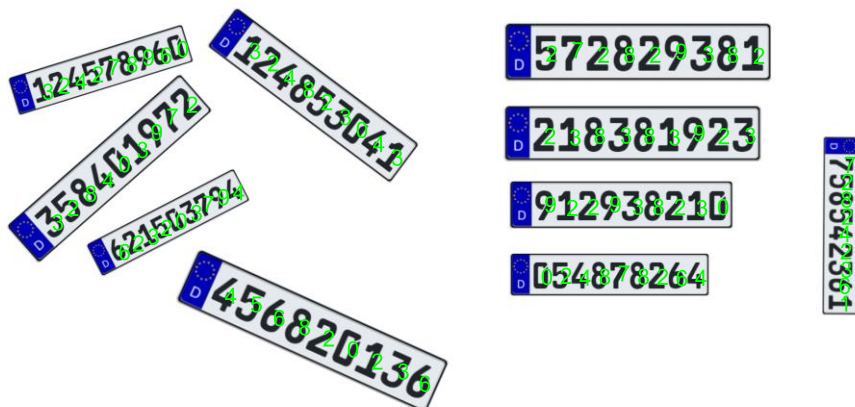


Ilustración 9. Resultado de la explotación con un PCI de 8,75%.



Tal y como se dijo en el entrenamiento, se esperaba que no se detectaran todos los dígitos a la perfección, ya que se hizo de forma forzada un cambio de relación de aspectos en diferentes muestras, con lo que se obtienen clases cuya varianza es alta. Se puede notar que solo se confunden dos números, el 1, al que a veces se les asigna el 2 o el 3, y el 5, al que a veces se le asigna el 2. El resto de números, incluidos el 2 y el 3, son identificados correctamente el 100% de las veces en esta imagen.

## 5. CÓDIGO MAIN.C

El código principal a ejecutar es *main.c*, donde se recurre a todas las funciones anteriormente mencionadas y sigue el procedimiento anteriormente expuesto.

```
% Ejercicio práctico 4 Sistemas de percepción: Reconocimiento de
objetos
% Programa principal
% Preparación:
clc
% Se añade la ruta de las imágenes de entrenamiento
addpath ./Entrenamiento;

% Comenzaremos con la preparacion de las muestras conocidas que servirán
de
% entrenamiento para construir el clasificador. El proceso de preparación
% consistirá en tratar y etiquetar la imagen, seguido del calculo de las
% características a usar para la clasificacion, en este caso, los
momentos
% de Hu.

% En primer lugar, será necesario cargar las imágenes de entrenamiento y
% las pasamos a blanco y negro con un umbral normalizado a elegir:
e0 = imread ('e0.png');
e1 = imread ('e1.png');
e2 = imread ('e2.png');
e3 = imread ('e3.png');
e4 = imread ('e4.png');
e5 = imread ('e5.png');
e6 = imread ('e6.png');
e7 = imread ('e7.png');
e8 = imread ('e8.png');
e9 = imread ('e9.png');

% A blanco y negro con un umbral normalizado de 0.5:
umbralBW = 0.3;

e0bw = uint8(not(im2bw(e0,umbralBW)));
e1bw = uint8(not(im2bw(e1,umbralBW)));
e2bw = uint8(not(im2bw(e2,umbralBW)));
e3bw = uint8(not(im2bw(e3,umbralBW)));
e4bw = uint8(not(im2bw(e4,umbralBW)));
e5bw = uint8(not(im2bw(e5,umbralBW)));
e6bw = uint8(not(im2bw(e6,umbralBW)));
e7bw = uint8(not(im2bw(e7,umbralBW)));
e8bw = uint8(not(im2bw(e8,umbralBW)));
e9bw = uint8(not(im2bw(e9,umbralBW)));

% return;

% Ahora, es necesario hacer el etiquetado de las imágenes para poder
% hacer el entrenamiento de los dígitos:
```

```

    % Se van a meter en una matriz de tres dimensiones la información de
    los
    % etiquetados para que sea más cómodo de trabajar con ella
    nClases=10;
    [M,N]=size(e0bw);
    et=zeros(M,N,nClases);

    et(:,:,1)=bwlabel(e1bw);
    et(:,:,2)=bwlabel(e2bw);
    et(:,:,3)=bwlabel(e3bw);
    et(:,:,4)=bwlabel(e4bw);
    et(:,:,5)=bwlabel(e5bw);
    et(:,:,6)=bwlabel(e6bw);
    et(:,:,7)=bwlabel(e7bw);
    et(:,:,8)=bwlabel(e8bw);
    et(:,:,9)=bwlabel(e9bw);
    et(:,:,10)=bwlabel(e0bw);
    % return;

    % Ahora, será necesario hacer el entrenamiento de cada uno de los
    dígitos.

    % Teniendo en cuenta que estamos trabajando exclusivamente con números
    del
    % 0 al 9, serán necesarios hacer la clasificación con 10 clases
    distintas.

    % Se puede notar que en cada una de las imágenes de entrenamiento hay 24
    % patrones, por lo que serán necesarios 24 patrones * 10 clases = 240
    % columnas.

    % En cuanto a las características a usar para hacer el clasificador, se
    % recurrirán a los momentos de Hu que mayor información nos ofrezcan por
    lo
    % que, a priori, se van a considerar los 8 momentos de Hu (7 filas + 1
    fila
    % para identificar la clase del patrón)

    % MatrizPatrones = zeros (8,240);
    MatrizPatrones = zeros(7,24,10);
    MediasPatrones = zeros(7,10); %Filas--> momentos de Hu 7; Columnas -->
    dígitos 10
    MatrizEntrenamiento = []; %PARA CONSTRUCTOR

    for i=1:10
        MatrizPatrones(:,:,i) = Entrenador(et(:,:,i));
        MatrizEntrenamiento(1:7,1+24*(i-1):24*i) = MatrizPatrones(:,:,i);
        MatrizEntrenamiento(8,1+24*(i-1):24*i) = i*ones(1,24);
        MediasPatrones(:,i) = calculaMediaPatron(MatrizPatrones(:,:,i));
    end

    MatrizEntrenamiento(1:7,:) = escaladoHu(MatrizEntrenamiento);

    %return;
    % Las medias están puestas como vector columna y nos interesa que esté
    como
    % vector fila:
    MediasPatrones = MediasPatrones';
    % return;

```

```

%% Preparacion del conjunto a explotar.

% Lectura de matrícula:
f = imread('matriculas.png');
umbralMatBW = 0.45;
fbw = uint8(not(im2bw(f,umbralMatBW)));
figure();
imshow(fbw, []);

% Ahora, es necesario identificar el marco de la matrícula para
eliminarla
% de los objetos de estudio. Para ello, se va a identificar dentro de
todas
% las etiquetas de esta última imagen, cuál no cumple cierta condición.
La
% matrícula se caracteriza por ser, por ejemplo, (área / masa) es mucho
% mayor que cualquier dígito.

% Para calcular el área, se van a emplear los ejes principales de cada
% número, ya que las matrículas se encuentran giradas y, de esta forma,
se
% calcularía de forma más precisa el área de cada etiqueta.

fet = bwlabel(fbw);
figure();
imshow(fet, []);

% Ahora, sacamos la información necesaria para poder obtener la masa de
% cada etiqueta, además de sus ejes principales de inercia y también se
% obtendrán los ángulos girados de cada etiqueta en caso de que sea
% necesario posteriormente operar con ellos.

[masa_et,a_et,b_et,ang_et] = informacionGeometrica(fet);

% return;

fet2 = limpiaPlantilla(fet,masa_et,a_et,b_et);

figure();
imshow(fet2, []);

% return;

%%
% Una vez etiquetados los dígitos de la imagen a explotar, es necesario
% analizar uno a uno e ir interpretando gracias al entrenamiento a qué
% dígito se corresponde.

% Para ello, será necesario calcular los momentos de Hu de cada dígito:

MomentosHuDigitos = calculaMomentosHu(fet2);

% Se puede apreciar que cada momento de Hu tiene un rango muy distinto de
% valores numéricos, por lo que es necesario hacer un escalado de ellas,
% por ejemplo, entre 0 y 1:

MomentosHuDigitosEscalados = zeros(size(fet2));

MomentosHuDigitosEscalados = escaladoHu(MomentosHuDigitos);

```

```
%% Clasificador

% Ahora ya disponemos de todos los datos necesarios cargados en memoria,
% tanto de los patrones como de la imagen que se quiere clasificar.
%
% Usaremos lo datos (momentos de Hu) de los patrones para construir el
% clasificador y validarlo, con el paso intermedio de decidir que
% subconjunto de características nos arroja un PCI mas favorable.

Constructor;
Validacion;

% Hecho esto pasamos a la explotacion, cotejando las características de
los
% digitos a explotar con la disponible en la matriz de prototipos
% generada por el constructor.

Explotacion;
```

El resto de código usado en el desarrollo del ejercicio práctico 4, tanto funciones como scripts debidamente comentados, se adjunta junto con la memoria en la entrega.

## 6. CONCLUSIONES

---

El proceso de clasificación de números de matrículas ha supuesto cierta complejidad técnica que ha sido además agravada por el hecho de deformar los números en relación de aspecto, ya que no es equivalente esta deformación a la que se realizaría por un simple cambio de perspectiva al tomar una imagen real.

El uso de los momentos de Hu como características para la clasificación mejoraría notablemente de haberse conservado la relación de aspecto tanto de los patrones como de las muestras a identificar.

Para mejorar la clasificación, sin arreglar la relación de aspecto, se necesitaría algún otro método de clasificación, otras características, como puede ser el escalado y disposición de los dígitos sobre una cuadrícula para identificar los píxeles de cada dígito, o el uso de derivadas para identificar los puntos clave de cada número.