

# Clase práctica 4.- Segmentación y reconocimiento

El siguiente enunciado corresponde a la segunda práctica puntuable de la asignatura. La puntuación de esta práctica corresponde al 30% de la asignatura. En particular:

- El ejercicio 1 corresponde al 10%
- El ejercicio 2 al 15%
- El ejercicio 3 al 5%

## 1 Segmentación de los dígitos de la matricula

Se desea construir un sistema que permita la segmentación de la dígitos de la matricula de un coche dada la caja que contiene el frontal del coche (ver figura 1).



Figura 1.- Ejemplo de imagen del frontal de un coche.

### 1.1 Desarrollo de la práctica

Para construir el sistema de segmentación de los dígitos se recomienda utilizar un enfoque basado en la binarización de la imagen y en la detección de objetos conexos. Se recomienda seguir los siguientes pasos:

1. Binarizar la imagen del frontal utilizando un umbral de valor 128, ya que este valor umbral permite dejar en negro los dígitos y en blanco el fondo.
2. Localizar los objetos conexos de la imagen binarizada utilizando la función `findContours` de `opencv`. En este caso debe solicitarse que devuelva todos los contornos (no solo los externos) y que no realice ninguna aproximación (pues no es necesaria la simplificación del contorno devuelto).

3. Se podrá filtrar los objetos que no parezcan corresponder a los dígitos. La información que puede utilizarse para este filtrado es que los dígitos serán 7 objetos de un tamaño similar y alineados.

## 2 Reconocimiento de los dígitos segmentados

Se desea construir un sistema que clasifique correctamente los dígitos extraídos de una matrícula.

### 2.1 Desarrollo de la práctica

Para construir el sistema de clasificación se proporciona un conjunto de entrenamiento que contiene 250 variaciones de cada uno de los caracteres que puede aparecer en una matrícula (en concreto los caracteres 0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ).

Para conseguirlo se puede seguir el siguiente guión:

- Cargar y umbralizar los caracteres a de aprendizaje para que correspondan a una imagen que tenga solo píxeles completamente blancos o negros.
- Convertir los caracteres de entrenamiento a un tamaño de 10x10, manteniendo la relación de aspecto (sin deformar los caracteres) y centrándolos en la caja de 10x10 cuando sea necesario. Se recomienda utilizar la función `resize` de OpenCV y la interpolación lineal.
- Convertir la matriz de cada imagen de un dígito, que ya tendrán un tamaño de 10x10 y valores entre 0 y 255, en una matriz con una sola fila y 100 columnas y valores entre 0 y 1. Cada una de estas matrices filas se usará como vector de características.
- Reducir la dimensión de los vectores de características usando LDA. Para ello se debe:
  - Agrupar en una matriz *C* (características) todas las filas generadas en el paso anterior. En este caso será una matriz con tantas filas como dígitos usemos para aprender y 100 columnas.
  - Crear un vector de enteros *E* (etiquetas) con las clase en la que se debe clasificar cada uno de los dígitos de aprendizaje. Obsérvese que la fila *f* de *C* contiene la imagen y la posición *f* de *E* contendrá la clase a la que pertenece.
  - Utilizar la clase LDA suministrada en “`subspace.hpp`” (obsérvese que está dentro del namespace `libfacerec`) para crear un objeto que recibirá la matriz *C* y el vector *E*.
  - Utilizar el método `project` del objeto LDA para proyectar *C* en un espacio de dimensión menor, creando una nueva matrix *CR* (características reducidas) .
- Crear un clasificador entrenado con la matriz *CR* y usarlo para reconocer los dígitos de cualquier matrícula. En particular se propone:
  - Utilizar un índice `flann` para construir un clasificador basado en *k* vecinos más cercanos. En este caso se recomienda utilizar como tipo de índice el `KDTreeIndexParams` y como distancia la distancia euclídea.

- Utilizar  $k=10$  y devolver como resultado la clase más votada o aquella que supere por ejemplo 5 votos.
- Generar una tabla que muestre los porcentajes de acierto obtenidos.

### **3 Integrar todos los algoritmos en una aplicación**

Desarrollar un sistema que utilice los detectores desarrollados en la práctica anterior y en esta para construir un programa que lea la matrícula de los coches que aparezcan en las secuencias de vídeo suministradas.

### **4 Normas de presentación**

Su presentación se realizará a través del campus virtual antes del examen de Mayo. Para presentarla se deberá entregar un único fichero ZIP que contendrá el código fuente, el ejecutable y un fichero PDF con la descripción del sistema desarrollado. Dicho fichero PDF incluirá una explicación con el algoritmo desarrollado, los métodos de OpenCv utilizados, copias de las pantallas correspondientes a la ejecución del programa y unas estadísticas correspondientes a los resultados de la ejecución de los algoritmos. Se permitirán grupos de hasta 3 alumnos.