Practica 4

Segmentación y reconocimiento de caracteres

David Vacas Miguel (GII+GIS)
Pedro Redondo Rabanal (GII+GIS)

Índice

1.-Introducción

Esta práctica consiste en localizar los dígitos de la matrícula de los coches y reconocer dichos dígitos. Una vez realizado esto debíamos escribir los datos adquiridos en un fichero de texto.

El clasificador utilizado para el reconocimiento de las matriculas es el proporcionado en el enunciado de las practicas.

2.-Explicación del algoritmo desarrollado

En este apartado pasaremos a explicar más detalladamente el funcionamiento de nuestro algoritmo.

En primer lugar pedimos por consola la dirección donde se encuentren las imágenes para entrenar nuestro clasificador. A cada una de estas imágenes las realizamos un suavizado (cv2.GaussianBlur(img,(3,3),0)) para eliminar posible ruido y realizamos un umbralizado. El umbralizado que finalmente utilizamos fue un umbralizado adaptativo mediante el método OTSU (cv2.threshold(blur,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)), puesto que probamos diferentes valores para distintos umbralizados y el que mejor resultado nos daba era éste. Una vez umbralizada la imagen, sacamos sus contornos (cv2.findContours(newImg, 1, method=cv2.cv.CV_CHAIN_APPROX_NONE)) y nos quedamos con aquellos que cumplan ciertas condiciones: que su altura sea mayor que ¼ de la imagen y que su altura sea mayor que su anchura. Gracias a esto podemos eliminar ciertos contornos que no son los que deberíamos coger. Una vez realizado esto ordenamos los contornos que quedan por área, cogemos el mayor y recortamos de la imagen el carácter. Finalmente redimensionamos la imagen resultante a una matriz de 10x10 (cv2.resize(newImg,(10,10),interpolation=cv2.INTER_LINEAR)) y devolvemos esta matriz como un vector de características de 1x100. Con todos estos vectores creamos un nuevo array llamado “C” y con LDA reducimos sus dimensiones (sklearn_lda.fit(C,E) y CR = sklearn_lda.transform(C)) y lo guardamos en un array llamado CR. El array denominado “E” es un array con las etiquetas posibles.

A continuación, inicializamos y entrenamos un clasificador con la matriz “CR” y el array de etiquetas “E”. En nuestro caso utilizamos knn con k=5 puesto que al realizar ciertas estadísticas era el que mejor precisión nos daba (estadísticas adjuntas al final del documento). Una vez entrenado el clasificador, pedimos al usuario por consola que introduzca tanto la dirección de donde se encuentran los archivos para realizar las pruebas como la dirección del clasificador de matrículas. Una vez hecho esto, por cada imagen en el directorio detectamos las posibles matriculas que haya y a estas les aplicamos el mismo método de localización de dígitos aplicado para las imágenes de entrenamiento con la diferencia de que ahora cogemos 8 caracteres en vez de 1. Por cada carácter reconocido obtenemos su array de características y reducimos sus dimensiones con lda para finalmente obtener el vecino más cercano a partir del clasificador (knn.find_nearest(np.asarray(newDes,np.float32),k=5)). Por ultimo escribimos los datos en un archivo .txt.

3.-Capturas de pantalla

Puesto que en esta práctica no se visualiza ninguna imagen por pantalla hemos decidido adjuntar dos archivos .txt resultantes de ejecutar el programa sobre las imágenes de prueba dadas en el enunciado de la práctica.

4.-Estadísticas

Precisión obtenida sobre los distintos clasificadores evaluados:

- Normal Bayes -> 79.2%
- Knn (k = 1) -> 76.4%
- Knn (k=3) -> 79.8%
- Knn (k=5) -> 81.4%
- Knn (k=7) -> 80.5%
- EM -> no hemos obtenido ningún resultado puesto que no hemos conseguido aplicarlo a la práctica.