Práctica obligatoria 2

Segmentación y reconocimiento de caracteres

El siguiente enunciado corresponde a la segunda práctica puntuable de la asignatura.

1 Normas

Las prácticas se realizarán en grupos de 2 alumnos y se presentará usando el aula virtual.

La fecha límite de presentación es el día del examen, pero si se presenta antes del 31 de Abril a las 23:00 la nota de esta práctica se incrementará en un 10%.

Para presentarla se deberá entregar un único fichero ZIP que contendrá el código fuente y un fichero PDF con la descripción del sistema desarrollado. Dicho fichero PDF incluirá una explicación con el algoritmo desarrollado, los métodos de OpenCv y scikit-learn utilizados, copias de las pantallas correspondientes a la ejecución del programa y estadísticas correspondientes al resultado de la ejecución del programa sobre la muestra de test.

La puntuación de esta práctica corresponde al 35% de la asignatura. En particular:

- El ejercicio 1 corresponde al 10%
- El ejercicio 2 al 20%
- El ejercicio 3 al 5%

La práctica deberá ejecutarse sobre Python 3.6 y OpenCV 3.4. y consistirá al menos en 2 ficheros python:

- leer_matricula.py
- leer_coche.py

Para ejecutar la práctica deberá escribirse en la consola de comandos “python” seguido del nombre del directorio de coches a procesar sin ningún otro parámetro adicional. Se supondrá que cualquier directorio utilizado por el programa estará en el mismo directorio que el fichero .py. Al ejecutar estos ficheros se mostrará por pantalla el resultado sobre las imágenes contenidas en dicho directorio.

El código desarrollado en las prácticas debe de ser original. La copia (total o parcial) de prácticas será sancionada, al menos, con el SUSPENSO global de la asignatura en la convocatoria correspondiente. En estos casos, además, no regirá la liberación de partes de la asignatura (habrá que volver a presentarse al examen) y podrá significar, en la siguiente convocatoria y a discreción del profesor, el tener que **resolver nuevas pruebas y la defensa de las mismas de forma oral**. **Las sanciones derivadas de la copia, afectarán tanto al alumno que copia como al alumno copiado.**

2 Localización de los dígitos de la matricula

Se desea construir un sistema que permita la localización de los dígitos de la matrícula de un coche dada la caja que contiene el frontal del coche (ver figura 1). Por tanto, partiremos de la detección del frontal del coche realizada en la práctica 3 y en esta sección realizaremos el siguiente paso necesario para la lectura de matrículas.

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

2.1 Desarrollo de la práctica

Para construir el sistema de segmentación de los dígitos se recomienda utilizar un enfoque basado en la umbralización de la imagen y en la detección de componentes conexas. Se recomienda seguir los siguientes pasos (aunque hay otras formas de obtener un resultado parecido):

1. Umbralizar la imagen (ver referencia [1] para el umbralizado en OpenCV con Python) con alguno de los métodos visto en clase (global o adaptativo). Por ejemplo, el umbralizado global dará como resultado una imagen del frontal binarizada (en negro los píxeles de los caracteres y en blanco el fondo de la matrícula o viceversa, dependiendo del umbralizado).
2. Localizar los objetos conexos de la imagen umbralizada utilizando la función cv2.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). Ver referencia [2] sobre cv2.findContours. Utilizando la función cv2.boundingRect se puede obtener el rectángulo que rodea al contorno detectado con cv2.findContours.
3. Se podrá filtrar los objetos que no parezcan corresponder a los caracteres. La información que puede utilizarse para este filtrado es que los dígitos serán 7 objetos de un tamaño similar, con una determinada relación de aspecto, alineados y agrupados. También se podrá utilizar el clasificador de Haar proporcionado con el enunciado (matriculas.xml) y que ha sido entrenado para buscar matrículas en la imagen. Podemos utilizar los rectángulos de localización de posibles matrículas para restringir la localización de los posibles caracteres.

3 Reconocimiento de los caracteres localizados (OCR)

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

3.1 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 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ y adicionalmente el símbolo con la E de España y la bandera de la UE que aparece a la izquierda de las matrículas actuales).

Para conseguirlo se puede seguir el siguiente guión:

- Cargar y umbralizar las imágenes de los caracteres de entrenamiento y realizar el mismo proceso sobre ellos que el realizado para la localización (por ejemplo, cv2.adaptiveThreshold + cv2.findContours + cv2.boundingRect).
- Construir el vector de características a partir de las imágenes de entrenamiento. Una posibilidad es utilizar directamente los niveles de gris. Todos los vectores de características deben de ser del mismo tamaño con lo que habrá que redimensionar la imagen del carácter a ese tamaño fijo (por ejemplo a 10x10 píxeles). Se recomienda utilizar la función cv2.resize de OpenCV y la interpolación lineal. Convertir la matriz con los niveles de gris del carácter ya redimensionada, que ya tendrán un tamaño de 10x10 y valores entre 0 y 255, en una matriz con una sola fila y 100 columnas. Cada una de estas matrices filas se usará como vector de características asociado al carácter de entrenamiento.
- Reducir la dimensión de los vectores de características usando LDA (Linear Discriminant Analysis). 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 clases 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 implementación de LDA disponible en el paquete de scikit-learn [3]. El método LDA.fit se utilizará para entrenar la matriz de proyección.
    - Utilizar el método transform del objeto LDA para proyectar C en un espacio de dimensión menor, creando una nueva matriz CR (características reducidas) .
- Entrenar un clasificador con la matriz CR y las etiquetas E para usarlo para reconocer los caracteres de cualquier matrícula. En la biblioteca scikit-learn y también en la biblioteca OpenCV (en el módulo de “machine learning”) tenemos implementados los clasificadores vistos en la asignatura. En la práctica se trata de probar los diferentes clasificadores y escoger el que de el mejor resultado.
- La evaluación de la práctica ser hará sobre imágenes de las que se conoce la posición de la matrícula y el contenido correcto de la misma (en un fichero de texto). La evaluación de la práctica se realizará con tres conjuntos de test:
    - testing_ocr (disponible para los alumnos). El conjunto sencillo de test con imágenes frontales y recortadas de coches y que servirá para establecer cómo se localizan los caracteres de la matrícula y se reconocen los mismos en condiciones favorables.
    - testing_full_system (disponible para los alumnos). Es el conjunto de test en condiciones más reales y que por tanto será más complicado.
    - testing_final (no disponible para los alumnos). Es el conjunto de test que los alumnos no tendrán disponible y que los profesores usarán para obtener un resultado adicional a los dos anteriores.

4 Integrar todos los algoritmos (localizar coche, matrícula y el OCR)

Desarrollar un sistema que utilice los detectores desarrollados en la práctica anterior y en esta para construir un programa que lea la matricula de los coches que aparezcan en imágenes y vídeo.

Todas las prácticas proporcionarán un script principal llamado “leer_coche.py”:

- El script leer_coche.py debe de poder ejecutarse con un parámetro que sea el directorio donde se encuentran las imágenes de coches:

  python leer_coche.py <path_completo_directorio_coches>

- El resultado de procesar el directorio será un fichero de texto con el nombre del directorio de pruebas. Por ejemplo, la llamada:

  python main.py C:\usuario\pepito\testing_ocr

  escribirá un fichero llamado testing_ocr.txt en el directorio desde el que se llamó a leer_coche.py.

- El contenido del fichero de texto será una línea por cada matrícula detectada (puede haber más de una en cada imagen) con el siguiente formato:

  <imagen> <x_centro_matrícula> <y_centro_matrícula> <matrícula> <largo_matrícula/2>

  donde:

  - <imagen> es el nombre del fichero de la imagen procesada sin el path
  - <x> e <y> son las coordenadas del centro de la matrícula detectada.
  - <matrícula> es el texto resultado de la clasificación de los caracteres de la matrícula
  - <largo_matrícula/2> Es el largo en píxeles de la matrícula detectada dividido entre 2.

5 Referencias

1. Tutorial sobre umbralizado con cv2 en python: http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html?highlight=threshold
2. Tutorial sobre la función cv2.findContours para encontrar los grupos de píxeles en imágenes umbralizadas: http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_imgproc/py_contours/py_contour_features/py_contour_features.html
3. Documentación sobre Linear Discriminant Analysis en sklearn: http://scikit-learn.org/0.16/modules/generated/sklearn.lda.LDA.html
4. Tutorial LDA vs QDA: http://scikit-learn.org/0.16/modules/lda_qda.html