

CC5509: *Optical Character Recognition* usando Histogramas de Orientaciones y K-Nearest Neighbor

Profs. José M. Saavedra / Mauricio Cerda
Ayudante: Camila Álvarez

13 agosto 2017

1. Objetivo

El objetivo de esta tarea es comprender y aplicar el descriptor basado en histogramas de orientaciones junto con el clasificador K-NN en el contexto del *reconocimiento óptico de caracteres* (OCR).

2. Descripción

La tarea consiste en implementar una herramienta OCR para reconocer el RUT presente en una imagen, como el ejemplo de la Figura 1.



Figura 1: Ejemplo de imagen conteniendo un RUT

Para lo anterior, se pide implementar un clasificador K-NN que utilice como vector de características el histograma de orientaciones de una imagen de dígito. Para este fin, se adjunta un *dataset* (digits) de dígitos sobre el cual se procede a realizar la búsqueda de los vecinos más cercanos.

Para facilitar el proceso, se entrega una implementación base en Python, que recibe como entrada una imagen que contiene un RUT, la transforma a una imagen binaria y la descompone en cada uno de los componentes que deben ser clasificados independientemente. El código debe ser completado de modo que el programa escriba por salida estándar el RUT reconocido, los puntos y guiones pueden ser ignorados.

2.1. Detalle

- Implementar un método de descripción basado en el histograma de orientaciones. El método debe ser implementado usando hard-assignment y soft-assignment (interpolación lineal).
- Implemente un clasificador basado en K-NN y evalúelo para K=1, 3 y 6.
- Evalúe las propuestas sobre las cuatro imágenes que se incluyen con la tarea.
- La tarea se debe entregar junto con un informe en formato paper como se describe en la siguiente sección.

3. Esquema de Informe

1. **Abstract o Resumen:** es el resumen del trabajo.
2. **Introducción:** se describe el problema y el contexto. (10 %)
3. **Desarrollo:** se describe el diseño e implementación del programa. (40 %)
4. **Resultados Experimentales y Discusión:** se debe presentar los resultados y hacer un análisis de los mismos. (40 %)
5. **Conclusiones** (10 %)

4. Restricciones y Condiciones

1. NO se aceptan tareas sin informe.
2. NO hay atrasos.
3. La tarea es individual.
4. Poner mucho esfuerzo en la redacción del informe.
5. La implementación se realizará en Python.

5. Entrega

La entrega se realiza por u-cursos hasta el 27 de agosto, 2017, 23:50 hrs.
Se debe incluir:

1. Código fuente (en Python)
2. Informe

6. Consultas

Escribir a la ayudante Camila Álvarez [camila.alvarez.i@ug.uchile.cl].