Pontificia Universidad Católica de Chile
IIC3724 - Reconocimiento de Patrones 1-2019

Por: Felipe Neut
25 de marzo de 2019

Informe Tarea 1

La relevancia de esta primera tarea es fundamental, ya que nos entrega una introducción a la formulación y resolución de un problema de reconocimiento de patrones. Se tuvo que explorar fenómenos que serán utilizados a lo largo del curso, tales como: realizar una obtención y adquisición de imágenes, fijar un *segmentation threshold*, extracción, análisis y selección de características de las imágenes, uso de funciones de procesamiento de imágenes tales como *bwlabel()*, y una evaluación del desempeño del clasificador diseñado.

La solución propuesta se basa en rellenar los huecos (entiéndase hoyo, cavidad u orificio) de las letras, y dependiendo de cuantos huecos sean rellenados se identifica la letra. Para el caso de esta tarea, dos huecos rellenados corresponde a la letra S, mientras que un hueco rellenado corresponde a la letra C. Como los pixeles de la letra tienen un uno, el hueco a rellenar (compuesto de ceros) estará entre pares de unos para cada columna de la matriz. Dicho eso, lo que se hará será "pararnos" en una columna específica de la matriz "*for j=1:columnas*", y mediante "*filas_unos_encontrados = find(y((1:filas),j)==1;"* se obtendrá un "array" con los números de las filas que tengan un uno (para la columna "j" de la matriz). Con lo anterior, se irá iterando sobre los pares de unos hallados en la columna "j", con el objetivo de encontrar el número cero entremedio de ese par de unos mediante: *"find(y(filas_unos_encontrados(n)+1:filas_unos_encontrados(n+1)-1,j) == 0);"*
Entonces, si es que hay ceros entremedio de algún par de unos, se rellenan con unos a través de: "*y(filas_unos_encontrados(n)+1:filas_unos_encontrados(n+1)-1,j)=1;"*
Si no encontré ceros dentro del par de unos anterior, buscaré si existen ceros en el siguiente par de unos que exista en la columna "j" (en caso de existir). Finalmente, mediante *"Y2 = xor(y,X);",* se entrega una nueva matriz cuyos unos corresponden exclusivamente a las columnas rellenados previamente con unos, desapareciendo así los pixeles de la letra. De esta forma, mediante *"bwlabel(Y2,4);"* se entrega el número de huecos u objetos detectados, cuyo valor entrega si la letra es una S o C como se dijo antes.

Se hicieron cien pantallazos de letras escritas en *Word* con distintos *fonts* (Times New Roman, Arial y Courier), diferentes tamaños, y distintas ubicaciones. El resultado obtenido fue de un 100% de éxito, con una velocidad muy rápida dada la simplicidad del código.

A modo de conclusión, el método utilizado para esta tarea tuvo un gran efectividad. Si bien no se basa en la extracción de características vistas en clases (tales como centro de masa, área, perímetro, momentos de Hu, entre otras), se logró diseñar un clasificador eficiente, rápido, efectivo e invariante a la escala, rotación y a la ubicación de las letras.
Cabe mencionar que el método hubiera presentado fallas si se añaden ciertas dificultades, tales como imágenes con ruidos o imágenes que se encuentran cortadas. Esos detalles adicionales podrían provocar que se detecten una mayor o menor cantidad de huecos.
El método usado en esta tarea apunta exclusivamente a la letra S y C, ya que se basó en las características de esas dos letras, no obstante, su uso se puede extrapolar a más letras. Ahora, se presentarían ciertas dificultades si es que las letras poseen un mismo número de huecos. Es ahí cuando habría que buscar complementar con la extracción y selección de más características, o métodos más robustos como los momentos de Hu, Fourier o LBP.