Segmentación de Tumor Cerebral
Taller de Procesamiento de Imágenes Médicas

1st Stalin Garrido Parra
Ingeniería Civil Electrónica
Universidad de Concepción
Concepción, Chile
stgarrido@udec.cl

2nd Carlos Kublik Muñoz
Ingeniería Civil Electrónica
Universidad de Concepción
Concepción, Chile
carlkublik@udec.cl

I. Introducción

En el siguiente documento se detallará el proceso de desarrollo del algoritmo implementado para completar el mini-proyecto n°1. Se trabajará con imágenes de resonancia magnética para un total de siete sujetos que presentan un meningioma diagnosticado. Para manejar las imágenes en formato **NIFTI** se utilizara la biblioteca **NiBabel**[1].

Se utilizaran técnicas de segmentación aplicando el método de K-means y técnicas de detección de borde aplicando el algoritmo Canny (ambos métodos implementados en la librería **OpenCV**[2] además de apoyarnos del software **BrainVISA Anatomist**[3] para poder seleccionar las laminas que nos permitirán desarrollar el algoritmo.

II. Desarrollo

El primer paso es elegir un corte dentro de los datos de la imagen donde el tumor sea visible y se pueda obtener la mayor información. Para esto nos ayudamos del programa BrainVISA mencionado anteriormente. Luego de tener claro el slice a analizar, cargamos el paquete a nuestro algoritmo con la función *nib.load*() y obtenemos los datos de cada slice con *.get_data*(). En las siguientes sub-secciones (II-A–II-E) detallaremos el procesamiento de imágenes realizado en el algoritmo.

II-A. Imagen original

Una vez cargado el paquete guardamos el slice, perteneciente al corte axial, para trabajar sobre la segmentación del tumor. Al hacer una inspección visual de la imagen vemos que posee distintas intensidades lo cual será fundamental para el trabajo de segmentación.

II-B. K-means

Es importante destacar que K-means es un método de agrupamiento de un conjunto de elementos en k grupos. La ventaja de ocupar K-means para el procesamiento de imágenes es que podemos agrupar la imagen según las intensidades de los píxeles (en este caso vóxeles), separándola así en las estructuras que la componen.

En nuestro caso, encontramos que agrupar la imagen en 3 grupos es la mejor manera para segmentar el tumor, teniendo así 3 estructuras claramente visibles: Tumor y corteza cerebral, materia gris junto con materia blanca y el fondo negro. Destacamos que al elegir un número de agrupamiento igual o superior a 4, se toman ciertas intensidades de vóxeles dentro del tumor como una clase distinta, por lo cual al segmentar el tumor se perderá información dentro del mismo.

Para aplicar esto al algoritmo, es necesario transformar el tipo de dato del corte elegido a una lista de *float32* para poder ingresarlo a la función de K-means proporcionada por la librería OpenCV. Con esto obtenemos un arreglo con las 3 etiquetas generadas por la función junto a la lista con la intensidades de los vóxeles convertidas a los 3 posibles valores de las etiquetas generadas. Luego, se hace el procedimiento necesario para convertir esta lista en una matriz numpy de tipo *uint8* para poder ocupar las funciones de OpenCV y generar la máscara que se explicará en la siguiente sub-sección.

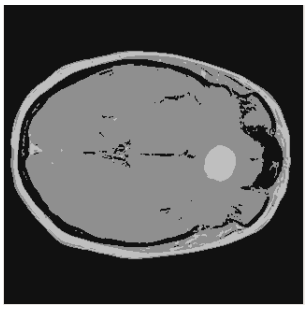
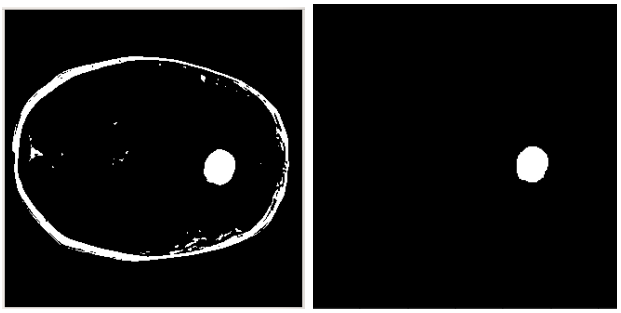


Figura 1. Etiquetas K-means.

II-C. Creación de la máscara

Visualizando la imagen generada por K-means, determinamos que la intensidad del tumor y corteza es más alta que el resto, es por esto que para generar la máscara aplicamos una umbralización, eligiendo de la lista de etiquetas generadas el valor más alto como valor umbral. Con esto obtenemos una máscara que contiene la corteza cerebral y tumor, entonces para obtener la máscara del tumor se procede a realizar una erosión a la imagen para así eliminar la parte de la corteza con

la que no deseamos, al realizar este proceso se pierde cierta información perteneciente al tumor (disminuye su tamaño) por lo cual se utiliza el proceso de dilatación de la imagen para acercarnos lo más posible al tamaño real del tumor.

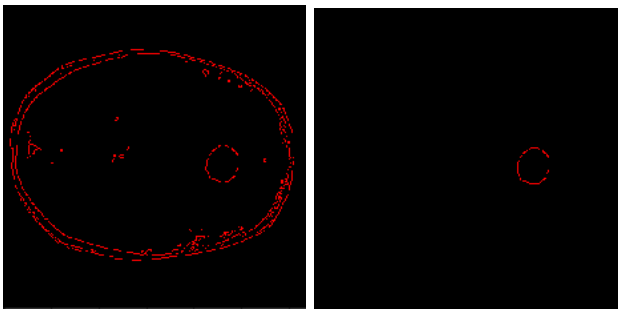


(a) Máscara Umbralizada (b) Máscara Tumor

Figura 2. Generación máscara tumor.

II-D. Máscara de delineado

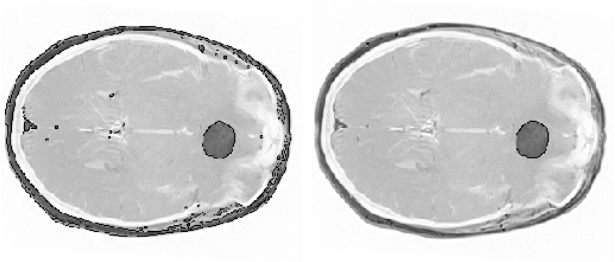
Una vez que obtuvimos las mascaras necesarias para la segmentación del tumor debemos realizar las máscaras de delineado de la corteza y del tumor. Si bien detectamos los bordes con Canny de OpenCv, estos quedan con un valor de 255 (blanco) por lo que necesitamos pasarlos a 0 y dejar el resto de los píxeles de la mascara en 1. Con esto podemos multiplicar este arreglo con el original y generar así la imagen con los contornos de los bordes en valor 0. Luego de realizado esto procedemos a crear una máscara con los bordes en color, esto se logro transformando la imagen al plano RGB", separando los canales y quedándonos solo con uno de estos (elegimos rojo).



(a) Mascara Bordes (b) Mascara Tumor

Figura 3. Mascaras para Delinear.

Ya que tenemos la imagen tratada como deseamos y las máscaras con los delineados procedemos a realizar la multiplicación de los dos arreglos y obtener un resultado con los bordes delineados.

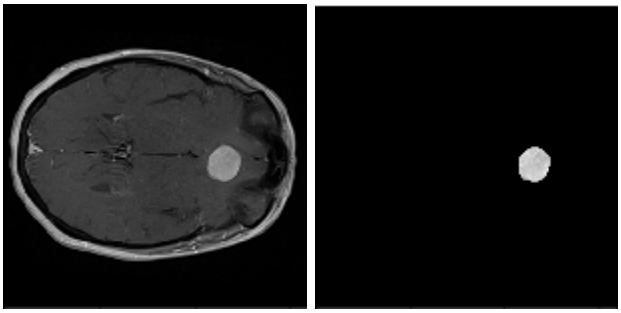


(a) Corte con bordes (b) Tumor con bordes

Figura 4. Corte con sus delineados.

II-E. Aplicación de la máscara

La máscara generada tiene valores de 0 y 255, entonces para aplicar la máscara al corte original se transformar a 0 y 1. Luego realizamos el productor punto entre la matriz del corte original y la máscara obteniendo así el tumor segmentado.



(a) Corte original (b) Tumor segmentado

Figura 5. Resultado final.

Con esto se logra segmentar el tumor del corte seleccionado. El procedimiento que utilizamos para los otros cortes (superiores e inferiores) es ocupar la máscara del corte más grande (idealmente la primera máscara es la más grande) y aplicarla al corte superior, luego revisamos la intensidad de vóxeles y los que no estén dentro del rango de intensidad de tumor les asignamos un valor igual a 0, entonces al segmentar el corte con una vista del tumor más pequeña generamos una nueva máscara para segmentar el siguiente corte más pequeño, esto se hace con la finalidad de que si ocupamos la misma máscara para todos los slices puede darse el caso que en un slice donde no se presente tumor, la intensidad de vóxeles de alguna otra parte sea igual a la del primer tumor segmentado y el algoritmo lo reconozca como parte de este y lo segmente. Este procedimiento lo ocupamos tanto para los slices superiores e inferiores del corte principal, entonces separamos el procedimiento en 2 ciclos *while*.
Para el cálculo de volumen del tumor, una vez segmentado completo recorremos todos los slices y guardamos la intensidad de cada vóxel distinto de 0 en una lista y también la cantidad de estos en otra variable, entonces para determinar la medida en mm de cada vóxel ocupamos la función *.header.get_zooms()* y multiplicamos las medidas en X, Y,

Z y el número total de vóxel para determinar el volumen el cual es de 7069 mm^3 para el caso 14. Además con la lista que contiene la intensidad de cada vóxel del tumor podemos calcular la intensidad promedio y desviación estándar con las funciones de numpy *.mean()* y *.std()* obteniendo: 944.84 $\pm 125{,}52$.

III. Conclusiones

Es importante mencionar que todo el algoritmo se realizo en base al archivo case_014_2.nii.gz, al momento de utilizar nuestra función en los otros casos nos encontramos con algunos problemas por las intensidades y las formas en que K-means agrupo las etiquetas. Este problema lo atacamos realizando una normalización de la imagen original para así asegurarnos de que el tumor quedase lo más cerca de 255 y así quedara almacenado en la etiqueta correcta.

Si bien no funciona para la totalidad los casos logramos aislar el tumor en otros 3 casos y generar así el nuevo archivo *.nii* con la información requerida, aún así, en alguno de estos casos la corteza cerebral es mucho más gruesa que el corte más grande del tumor, por lo que al erosionar la primera parte de la máscara esta sigue presente junto al tumor y luego al dilatar la imagen, queda presente un segmento de esta en la máscara final.

Notas

[1] https://nipy.org/nibabel/
[2] https://docs.opencv.org/master/index.html
[3] http://brainvisa.info/web/index.html