

Entrega 2. Herramientas para Procesamiento de Imágenes

Santiago Villalba
Universidad de los Andes
Cra. 1 18a 12, Bogotá, Colombia
sh.villalba@uniandes.edu.co

Nicolás Garnica
Universidad de los Andes
Cra. 1 18a 12, Bogotá, Colombia
pn.garnicar@uniandes.edu.co

1. Como citar un codigo

De acuerdo al formato de referencias IEEE, un codigo autoria de un tercero posee una estructuración de referencia particular. Primero debe ir el nombre de los autores (Primer apellido, seguido de la inicial del primer nombre de cada autor) o de la institución que genero el codigo. Posterior a ello se debe colocar el titulo del codigo. Luego se debe agregar el url donde se obtuvo el codigo (esto hace referencia al repositorio donde se encuentra, usualmente github es utilizado como repositorio). Por ultimo, se agrega el año donde el codigo fue creado. [1] Un ejemplo de como se cita se muestra a continuación:



Figure 1. Componentes conexos con vecindad 4, resultado sea igual a cuando la función se aplica con vecindad 8.

2. Preprocesamiento de Imágenes

2.1. Función de componentes conexos

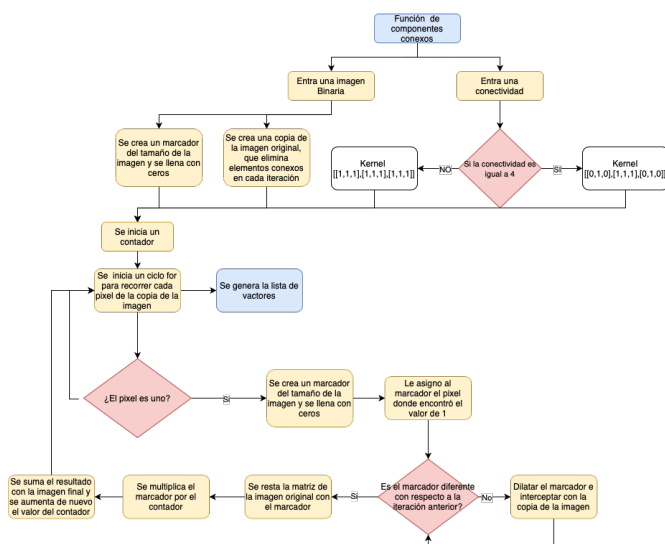


Figure 2. Diagrama de flujo del algoritmo definido.

2.2. ¿Qué características deberían tener las imágenes para que no se dificulte la tarea de extraer componentes conexos?

En una primera instancia, la imagen a la que se le quiere aplicar este tipo de algoritmos debe estar debidamente binarizada, es decir 0 para el fondo y 1 para los elementos de interes. Por otra parte, asegurarnos de que la imagen no tenga ningun tipo de ruido dado que esto puede afectar la etiqueta de los elementos conexos. Adicionalmente la distancia juega un rol importante en evitar que dos elementos conexos se puedan unir equivocadamente, de acuerdo al concepto de conectividad.

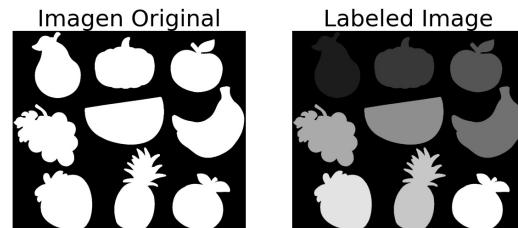


Figure 3. Componentes conexos con vecindad 4, resultado sea igual a cuando la función se aplica con vecindad 8.

Una vez aplicado el algoritmo propuesto sobre la imagen de las frutas, observamos que la salida nos devuelve una imagen cuyos elementos tienen una etiqueta en particular. Estas etiquetas hacen que, al graficar la imagen, sean diferentes escalas de grises para cada una de las frutas. Esto refleja el correcto funcionamiento del algoritmo dentro de tiempos de computo razonables (menos de 2 minutos).

Despues de haber verificado el correcto funcionamiento del algoritmo con la imagen de las frutas, se procede a evaluar los coceptos de conectividad 4 y 8. Para ello se proponen dos imagenes 20x20.

La primera imagen (ver figura 3) devuelve el mismo resultado, tanto, para conectividad 8 y conectividad 4. Esto

se debe a que la imagen propuesta tienen dos componentes conexos que están debidamente distanciados entre ellos y no afecta su resultado a pesar de sus diferentes conectividades.

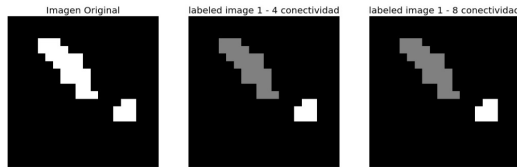


Figure 4. Componentes conexos con vecindad 4, resultado sea igual a cuando la función se aplica con vecindad 8.

La segunda imagen propuesta (ver figura 4), devuelve resultados diferentes para conectividad 4 y conectividad 8. Esto debido a que no existe una pertinente separación de los componentes conexos de la imagen, es decir, el segundo elemento conexo contiene un pixel el cual posee una posición cercana que no es reconocible por la conectividad 4 pero si por la conectividad 8.

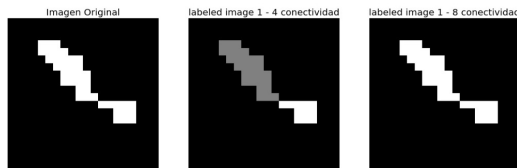


Figure 5. Componentes conexos con vecindad 4, resultado sea diferente a cuando la función se aplica con vecindad 8.

3. Watersheds

Lo primero fue filtrar las imágenes con un filtro mediano del paquete de *OpenCV*, eliminando estructuras pequeñas no deseadas y el ruido sal y pimienta. Como parámetro se utilizó un kernel de 5x5. Posterior a esto, se utilizó la función *rgb2gray* de la librería *skimage.color* para pasar las imágenes RGB a escala de grises. Para obtener el gradiente morfológico de la imagen, se utilizaron las funciones *dilate* y *erosion* del paquete de *skimage.morphology* con todos los parámetros predeterminados, es decir, utilizando el elemento estructurante predeterminado de las funciones que en este caso es una cruz. Posteriormente se restó el resultado de la erosión al resultado obtenido con la dilatación de la imagen.

3.1. ¿Qué está utilizando la función watershed como "marcador" para realizar esta segmentación cuando no hay marcadores pre-establecidos?

Cuando no hay marcadores establecidos, el algoritmo de watersheds utilizará los mínimos locales de la imagen para crear agujeros y comenzar el proceso de inundación, indicando que los mínimos locales de la imagen serán los marcadores tomados por el algoritmo.

3.2. ¿Cuál es el efecto de emplear watersheds de esta manera?

Como se puede observar en la Figura 6, el proceso de inundación tendría en cuenta todos los mínimos locales de la imagen y el resultado de esto será una imagen sobresegmentada.

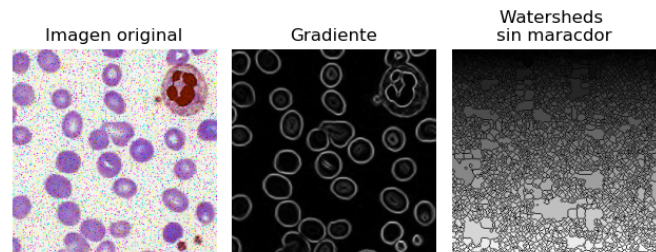


Figure 6. Análisis de la imagen 9 de la base de datos, utilizando el gradiente morfológico y watersheds

3.3. Obtención de marcadores

Los marcadores de la imagen se obtuvieron con la función *hminima*, cuyo parámetro *h* fue fijado en 65 y el kernel utilizado fue uno en forma de cruz de tamaño 3x3. El valor de *h* fue escogido después de realizar varias pruebas con distintos umbrales de manera cualitativa, en donde escogimos el umbral que eliminaba las estructuras pequeñas que no hacen parte de los globulos rojos y que al mismo tiempo conservaba la mayor cantidad de estos últimos.

3.3.1 ¿Qué ventajas presenta este método de obtención de marcadores automática sobre un posible método manual?

La ventaja de automatizar el método de obtención de los marcadores, es la velocidad de análisis y autonomía de la máquina para realizar el procesamiento de las imágenes. Se esperaría que los algoritmos implementados pudieran identificar correctamente cada globulo rojo de forma autónoma sin necesidad de intervención humana. Teniendo en cuenta esto, la Figura 7 muestra como debería verse un marcador manual que posteriormente debería ser insertado en la función de componentes conexos para realizar watersheds únicamente en los globulos rojos.

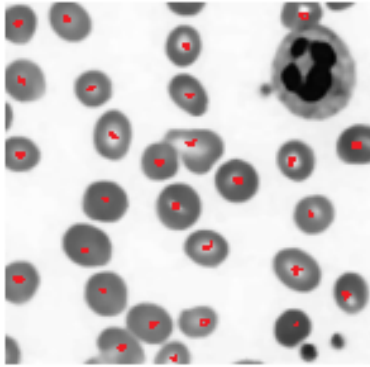


Figure 7. Marcadores manuales para watersheds en la imagen 9 de la base de datos

3.4. Watersheds con marcadores

Después de obtener los minimos con la función *hminimos*, se utilizó la función de *watersheds*. El resultado obtenido con los marcadores, puede observarse en la Figura 8.

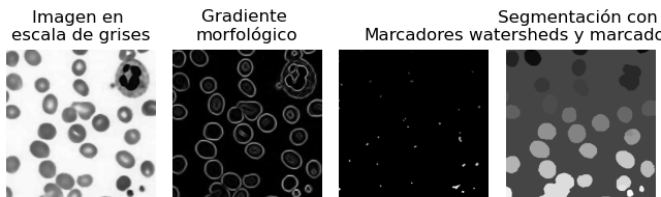


Figure 8. Aplicación de watersheds en la imagen 9 de la base de datos con sus respectivos marcadores.

3.4.1 ¿Cuál de los métodos genera un mejor resultado cualitativo de segmentación? ¿Por qué?

Cualitativamente, la mejor segmentación fue dada por el método de watersheds con marcadores, ya que el algoritmo de watersheds puede crear lagos y represas en los diferentes componentes conexos que se identifican con los marcadores. En contraste con watershed sin marcadores, se observa que este método no diferencia entre el fondo y los globulos rojos.

3.4.2 ¿Cuál es el rango de intensidades en estos resultados? ¿Qué significa cada intensidad?

Los rangos de intensidad sera la cantidad de componentes conexos que existan en la figura, es decir, la cantidad de marcadores. Cada intensidad hace referencia a un componente conexo determinado.

3.4.3 A partir de estos resultados, ¿cómo podría aproximar cuántos glóbulos rojos hay?

Teniendo en cuenta lo dicho en la sección anterior, si la segmentacion fuera perfecta, se podria aproximar la cantidad de globulos rojos con la cantidad de componentes conexos obtenidos con ayuda de los marcadores y el algortimo de componentes conexos con *watersheds*.

3.4.4 ¿Qué tipo de segmentación me permite hacer estos dos métodos?

La segmentacion binaria permite la utilización de estos dos algoritmos de watersheds, debido a la naturaleza de los componentes conexos y la segmentacion del fondo de estos, en caso de que los marcadores esten correctamente ubicados sobre la imagen y en las estructuras que se desean segmentar del fondo.

4. Comparación de métodos de segmentación

La binarizacion de las imágenes obtenidas genero problemas ya que en cada imagen, el fondo tiene un valor distinto de gris, así como el valor de gris para los componentes conexos que representan los globulos blancos. Esto qgeneró problemas para separar los globulos rojos del fondo con unico umbral al mismo tiempo que se separan tambien los globulos blancos de los rojos. Esto conlleva a la utilizacion de un umbral que se adapte a cada imagen, ya que seria imposible encontrar un umbral o umbrales especificos para cada imagen para de esta forma segementar los globulos rejos del resto. Por lo tanto el umbral que se adapta a cada imagen fue el umbral de Otsu, y fue este el que utilizamos para hallar las máscaras binarias. Los resultados de la segmentacion en comparacion con las imágenes de la carpeta *Ground Truth* se muestran en la Figura 9.

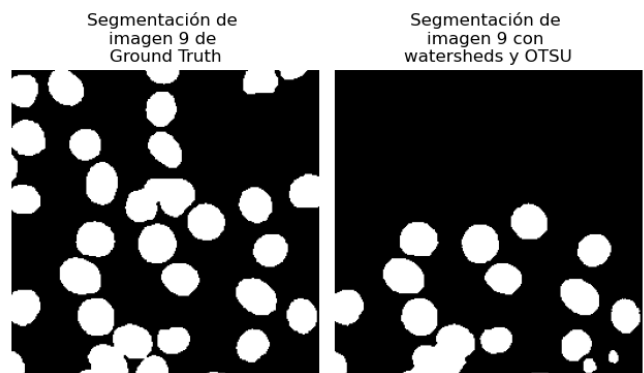


Figure 9. Comparación de la mascara binaria obtenida para la imagen 9 de la base de datos

Como se puede apreciar en la Figura 9. en las máscaras binarias obtenidas se pierde mucha información de los glóbulos rojos presentes en las imágenes.

4.1. ¿Por qué se deben binarizar las máscaras obtenidas con los métodos de watersheds?

La binarización ayuda a la segmentación del fondo de las estructuras que se desean, obteniendo así una imagen en blanco y negro que ayudará a evaluar la eficacia del método. además de que permite eliminar algunos niveles de gris determinados, y ya que cada componente conexo producto del algoritmo de watersheds es un nivel de gris en específico, se pueden eliminar elementos conexos que no sean de interés, como los glóbulos blancos en este caso. Sin embargo, la automatización de este método se dificulta para el análisis de varias imágenes con cuando los valores del fondo y de los componentes conexos, son diferentes a los de las demás imágenes, haciendo difícil la escogencia de intervalos o umbrales que permitan una correcta segmentación.

4.2. ¿Cuál de los métodos genera un mejor resultado de segmentación? ¿Por qué?

Los resultados que obtuvimos, utilizando el índice de Jaccard como medida de semejanza entre las máscaras obtenidas con las diferentes metodologías y las máscaras de la carpeta Ground Truth, se muestran en la Figura 10.

Imagen	Watersheds + marcador	Watersheds	Relleno huecos (Pre-procesamiento)
1	0.6047	0.2429	0.8121
2	0.3287	0.1696	0.7394
3	0.3926	0.2241	0.7933
4	0.37	0.2339	0.763
5	0.2456	0.1859	0.7384
6	0.3336	0.2009	0.8205
7	0.6087	0.2064	0.8258
8	0.4144	0.2042	0.7602
9	0.4929	0.2302	0.7444
10	0.4136	0.1652	0.7024
Promedio	0.4285	0.2063	0.7699
Desviación	0.1267	0.0254	0.0391

Figure 10. Comparación de la máscara binaria obtenida para la imagen 9 de la base de datos

Donde se observa que en general, el método que tuvo una mejor puntuación en cuanto al índice de Jaccard, fue el método de dilatación geodésica con las imágenes pre-procesadas. Esto puede deberse a que en la binarización de las imágenes segmentadas con watersheds y marcadores, se pierde mucha información como se mencionó en las secciones anteriores. De estos resultados concluimos que la eliminación de los glóbulos blancos debe realizarse en el pre-procesamiento, siendo éste el proceso esencial para una correcta segmentación independientemente del método de segmentación. Posterior a la eliminación de los glóbulos blancos, se podrían haber hallado las estructuras conexas con watersheds y modificar las intensidades con transformaciones de intensidad para poder automatizar la segmentación de los glóbulos rojos del fondo, con un único

umbral.

4.3. ¿Cree que incluir estadísticas como el promedio y la desviación estándar de sus resultados cuantitativos es útil para analizar y comparar sus métodos? ¿Por qué?

Si es útil, ya que el promedio demuestra que tan eficaz fue el método de segmentación en general. mientras que la desviación estándar muestra si la eficacia del método de segmentación varía de acuerdo a la imagen, lo que nos da una medida de que tan bueno es dicho método para generalizarlo en cualquier set de imágenes. Según los resultados obtenidos en la Figura 10. el método de watersheds con marcadores es el método con mayor desviación estándar, lo que indica que dicho método varía su eficacia dependiendo de la imagen a la cual éste fue aplicado.

References

- [1] W. staff, "How to cite a github repository," 2020. [Online]. Available: <https://www.wikihow.com/Cite-a-GitHub-Repository>

Realizado en L^AT_EX