Aseguramiento de la Calidad del Software: Proyecto Semestral I

M. Sc. Saúl Calderón Ramírez
Instituto Tecnológico de Costa Rica,
Escuela de Ingeniería en Computación,
PAttern Recognition and MAchine Learning Group (PARMA-Group)

7 de agosto de 2018

El presente proyecto pretende ser desarrollado en grupos de tres personas, a lo largo del curso de aseguramiento de la calidad del software. El proyecto será insumo para aplicar distintos estándares de calidad en sus varias etapas, por lo que a lo largo del curso se realizarán tareas y agregados específicos al proyecto. El objetivo es desarrollar un sistema que preprocese, segmente y detecte células en tejido de glioblastoma en imágenes digitales, tomadas por un microscopio basado en fluorescencia.

Fecha de entrega: 19 de noviembre

1. Introducción y motivación

Los dispositivos de captura y procesamiento de imágenes biomédicas, como resonancias magnéticas, radiografías y tomografías computarizadas, cobran cada vez mayor protagonismo los procesos de toma de decisiones para los profesionales de salud. Más recientemente, la industria biomédica ha desarrollado interesantes sistemas de software que realizan diagnósticos automáticos a partir de imágenes y señales biomédicas.

Un ejemplo de ello es el Laboratorio de Quimiosensibilidad tumoral, de la Universidad de Costa Rica, donde los doctores Steve Quirós y Rodrigo Mora desarrollan proyectos de investigación relacionados con la terapia personalizada para pacientes con cáncer. Su premisa consiste en desarrollar tratamientos quimioterapeúticos personalizados a cada paciente, de forma que los químicos utilizados sean optimizados. Para ello es necesario realizar experimentos *invivo* para estudiar la reacción de a nivel celular en el tejido cancerígeno a un tratamiento específico. El laboratorio cuenta con microscopios electrónicos de fluorescencia, los cuales permiten capturar imágenes digitales de la actividad celular en un tejido, posibilitando la visualización de las distintas reacciones de un conjunto de células a un tratamiento específico.

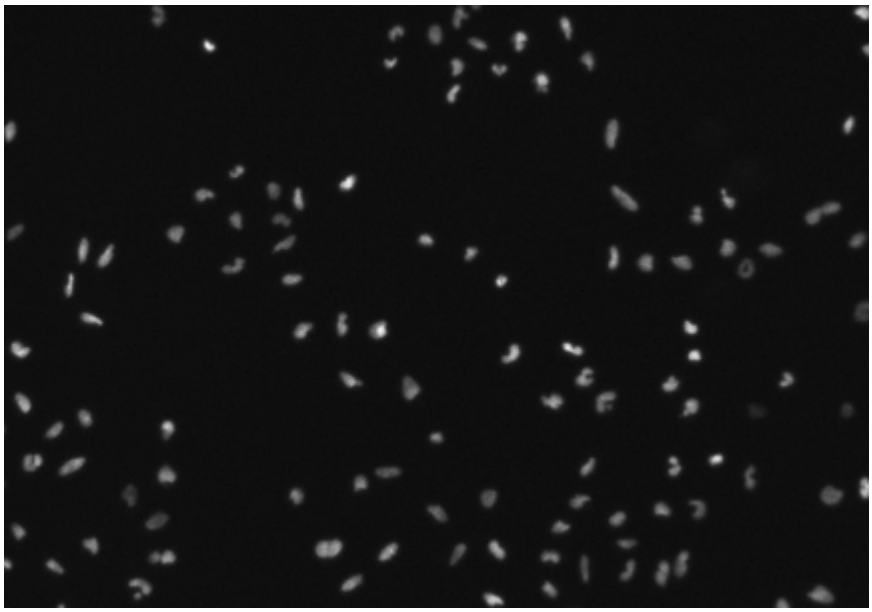

Figura 1: Células en tejido de glioblastoma, en las primeras horas del experimento.

Los experimentos típicos tardan de 72 a 96 horas, y se toman muestras cada 10 minutos, por lo que es usual la generación de miles de imágenes para cada experimento. El análisis manual de esas miles de imágenes para determinar la cantidad de células en un momento específico, o incluso determinar la descendencia de una o más células, son tareas difíciles (e incluso muchas veces imposibles) para el experto. Un sistema que automatice la detección, rastreo y determinación de la línea hereditaria de cada célula tiene el potencial de ahorrar una cantidad importante de tiempo a los expertos, además de la posibilidad de analizar muchos más experimentos, facilitándoles el enfocarse en aspectos de mayor relevancia.

2. Segmentación de imágenes

El presente proyecto consistirá en la segmentación y conteo de células a partir de imágenes digitales tomadas de un microscopio basado en fluorescencia, a tejidos de glioblastoma (tejido cerebral canceroso). La detección de los objetos de interés es una etapa necesaria para realizar el rastreo y acumulación de datos de la dinámica celular.

Para la segmentación de imágenes se utilizará una variante de la arquitectura *fully convolutional network (FCN),* llamada *U-net*, la cual se detallará más adelante *[1].*

3. Métricas para la comparación entre señales binarias

Para evaluar la precisión de la segmentación, las métricas usuales evalúan la diferencia de señales binarias (donde por ejemplo los pixeles etiquetados con 1 se refieren a los objetos de interés y los etiquetados con 0 al fondo). Una métrica muy utilizada para tales efectos es el coeficiente de *dice* o de Sørensen, el cual se puede usar para evaluar la similitud estructural entre dos imágenes binarias $U \in \mathbb{R}^{m \times n}$ y $V \in \mathbb{R}^{m \times n}$, y viene dado por:

$$S = \frac{2\left(\sum_j^m \sum_i^n U\left[i,j\right]\ V\left[i,j\right]\right)}{\left(\sum_j^m \sum_i^n U\left[i,j\right]\right) + \left(\sum_j^m \sum_i^n V\left[i,j\right]\right)}$$

donde el numerador equivale conceptualmente a la cantidad de pixeles del objeto de interés intersecados en ambas imágenes U y V : $2\left|U \cap V\right|$(donde las barras denotan la cantidad de pixeles con 1's en la matriz) y el denominador a la suma de los pixeles del objeto de interés en ambas imágenes: $|U| + |V|$, lo que entonces significa que se puede reescribir S como:

$$S = \frac{2\left|U \cap V\right|}{|U| + |V|}$$

La imagen U puede asociarse a la imagen segmentada por el algoritmo, y V la imagen segmentada manualmente, también conocida como imagen de *ground-truth.* De este modo, una segmentación perfecta según el *groundtruth* resulta en un coeficiente de *dice* $S = 1$, y una segmentación completamente errada un coeficiente $S = 0$. La Figura 2 ilustra la segmentación automática de una radiografía pulmonar, y el *groundtruth* de la misma.

4. Requerimientos de la aplicación

La aplicación debe proveer una interfaz gráfica de usuario amigable, que permita (por orden de prioridad):

1. Cargar una secuencia de imágenes almacenada en la dirección provista por el usuario.
2. Detectar y contar los objetos de interés en las imágenes (células), de forma que sea fácil de visualizar desde la interfaz gráfica.
    a) Opcionalmente (no opcional para los equipos de 4 personas) colorear y etiquetar cada objeto.
3. Genere un informe que indique por cada objeto encontrado, su centroide y su área, en formato *.csv*.

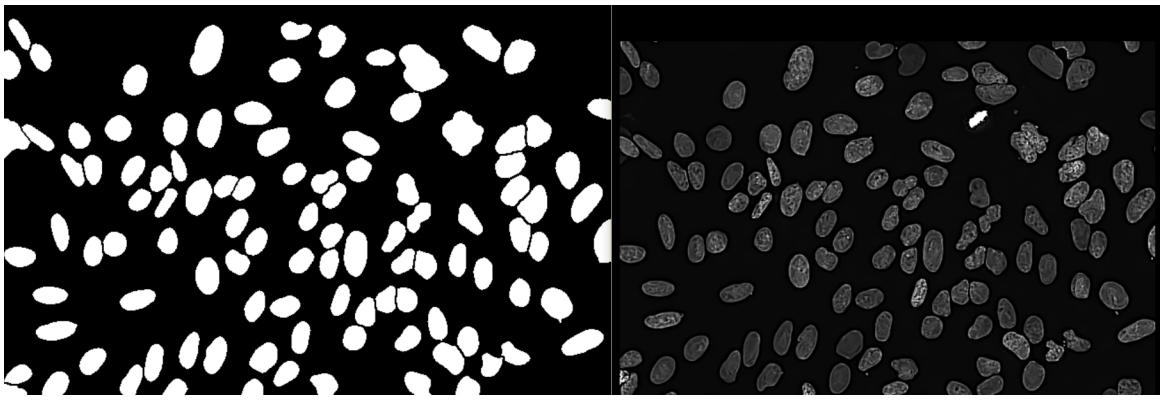

Figura 2: *Groundtruth* a la izquierda, e imagen de entrada a la derecha, tomadas del dataset BBBC06 `https://data.broadinstitute.org/bbbc/`, a utilizarse en el curso .

4. Permitir la navegación entre imágenes y resultados.
   a) Debe permitirse el procesamiento de un lote de imágenes en una sola acción.
5. Visualizar el tiempo de ejecución por imagen y por el lote de imágenes.
6. **Métrica de exactitud en los resultados:** Cargar un archivo de *ground truth* con los blobs de los objetos de interés manualmente marcados, para cuantificar la precisión de la segmentación, usando la métrica de Dice por ejemplo.

5. Implementación

Para el diseño e implementación del proyecto, es necesario usar la metodología de desarrollo Scrum, usando alguna herramienta para gestión de proyectos en línea como *zoho projects.* Como herramientas de diseño, se recomienda usar *StarUML*, *Altova*, *Visio*, etc. Para promover el atributo de mantenibilidad del proyecto, al menos un patrón de diseño debe ser implementado.

Respecto a la codificación del proyecto, es muy obligatorio el uso del lenguaje Python, usando Eclipse como ambiente de desarrollo, pues existen herramientas y *plugins* relacionados con el cálculo de métricas y gestión de la calidad del proyecto, lo cual será un rubro importante a evaluar. **Los estándares de calidad a aplicar se agregarán a lo largo del curso.** Es conocido el soporte de Python con Keras, numpy y OpenCV (opcional) para la implementación de redes convolucionales. Además se recomienda utilizar *Pandas* para la manipulación de archivos *.csv*.

La documentación final del proyecto debe incluir todos los estándares implementados a lo largo del curso, incluyendo los realizados en tareas, de manera consistente. Es el obligatorio el uso de LaTeX o algún editor basado en esa tecnología como LyX para la documentación del proyecto. Para facilitar la edición colaborativa, se recomienda el uso de la herramienta *Overleaf*.

Referencias

[1] S. Calderon J. Torrents-Barrena D. Puig S. Romani D. Riccio A. Garcia-Pedrero D. Marin S. Quiros W. Zamora. Assessing the impact of the deceived non local means filter as a preprocessing stage for a convolutional neural network based approach for cell segmentation in fluorescence based microscopy images. *CIARP 2018*.