

Trabajo práctico - Machine Learning e Imágenes en Python

Detección de bordes en imágenes de microscopía óptica de transmisión

Victor Manuel Juarez

Introducción

La microscopía óptica de transmisión permite observar células y vesículas sin tener que modificarlas con moléculas fluorescentes, como es el caso de la microscopía confocal. Para mejorar el contraste de la imagen, se puede utilizar Interferencia diferencial de contraste (DIC, por sus siglas en inglés Differential interference contrast) o contraste de fase (PC, por sus siglas en inglés Phase Contrast).

Si se cuenta con una cámara de rápida adquisición y guardado de imágenes se podría obtener datos de procesos dinámicos de mejor calidad que empleando microscopía confocal, la cual es de barrido. El análisis de las imágenes es más complejo que en el caso de imágenes confocales. Poder detectar el contorno de la vesícula y seguirlo en el tiempo permite conocer la rigidez de la vesícula y como este parámetro puede ser afectado con diferentes estresores (como etanol, analgésicos, antibióticos y otros fármacos) a los cuales están sometidas las membranas celulares, llegando a conocer su sensibilidad.

En particular, nos interesa detectar el borde de vesículas gigantes (entre 5-20 micrómetros de diámetro), y poder seguir este contorno en el tiempo. Esto es un desafío debido a que las vesículas se visualizan claramente al ojo, pero son muy difíciles de binarizar debido a que el contorno no corresponde a un nivel único de gris en toda la vesícula, sino que varía a lo largo de la misma. Lo que se observa es que el borde de la vesícula corresponde a un fuerte gradiente en los niveles de grises, lo que implica un desafío a nivel de algoritmos que permitan detectar el contorno.

Desarrollo

Armado de la base de datos: Partiendo de los videos se extraen los frames para la BDD obteniendo de los 3 videos 700 imágenes (300,300,100 de videos 1, 2 y 3 respectivamente). Se guardan las imágenes en una variable "BDD". Todas las imágenes de la BDD son monocromas. Dimensiones de los imágenes/videos: primera (137,137), 18769 pixeles; segunda (512,512), 262144 pixeles; tercera (181,175), 31675 pixeles. Tipo de datos: uint8.

Análisis de histogramas: De cada video se extrae una imagen aleatoria para graficar sus histogramas y conocer los niveles de gris con los que se está trabajando. Se encuentra que las dos primeras imágenes tienen intensidades similares, mientras que la última está 100 niveles de gris/intensidad por encima.

También se realiza un histograma con máscara circular para encerrar a la célula y poder comparar el ambos histogramas (original y con máscara). Se encontró que los niveles de gris de la imagen completa están superpuestos con los niveles de gris en el borde celular. En el histograma con máscara, a la derecha, tenemos pequeños picos que pueden corresponderse con el borde celular. Se busca intensificar estos niveles de gris para poder usarlos en el entrenamiento del modelo.

Búsqueda de nuevas características (filtrado y binarizado): Con el objetivo de aumentar el número de características por imagen y realzar el borde celular se aplican filtros a las imágenes del primer video. Filtros Prewitt y Sobel para detección de bordes; Filtro Laplaciano para aumentar valores de intensidad altos y disminuir los bajos. Al final se realiza un proceso de erosión y binarización de la imagen.

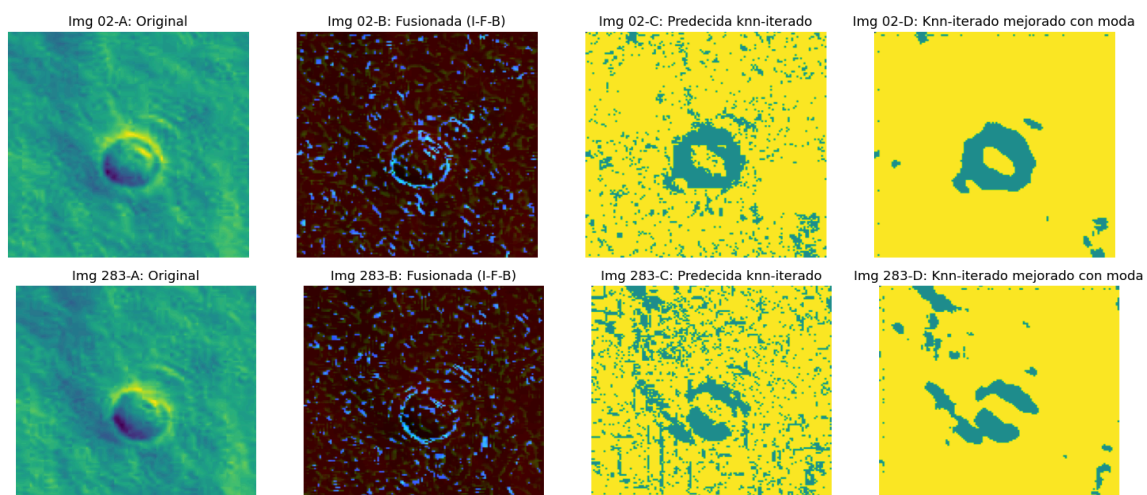
Imagen a data frame: Se utilizan la imagen original "Img 02-A: Original", y las imágenes procesadas "Img 02-D: SPL Suavizada" y "Img 02-E: Binarizada" para armar un dataframe de la imagen 02. Columnas del data frame: "pixel_id", "intensidad", "filtrada", "binarizada".

KNN: Se intenta con un modelo de KNN con n-neighbors igual a 1, 3, y 7 (valores seleccionados de acuerdo a la precisión del modelo con el conjunto de entrenamiento). Se obtienen imágenes con mucho ruido y al aplicar el ruido de moda, el mismo queda en la imagen.

KNN con co-learning: Se usó para el entrenamiento el modelo de K-Nearest Neighbors con asignaciones progresivas (co-learning). Los n-neighbors fueron: 3, 7, 17 y 21. El input fue el dataframe ya descrito. El output detecto bien el borde aunque con ruido de fondo, por lo tanto se realizó un filtro de moda para limpiar estos valores llegando a resultados óptimos.

Luego se intentó extrapolar este modelo a otras 3 imágenes (una de cada video) para analizar el comportamiento del modelo, obteniéndose resultados poco prometedores en cuanto a imágenes del video 3. Esto último debido a los niveles de intensidad altos y poco contraste que tiene este video. Variando parámetros como el nivel al que se umbralizar las imágenes y llevando los niveles de intensidad al rango de la imagen usada para el entrenamiento, se observa una mejoría en los resultados. Se necesita mucha intervención por parte del usuario para este ajuste.

Resultados: Se pudo detectar los bordes de imágenes pertenecientes a videos 1 y 2. Las imágenes del video 2, tienen pérdida del borde celular. Respecto a la imagen del vídeo 3, el modelo no pudo realizar ninguna detección.



Conclusión

Los filtros Prewitt y Sobel al ser filtros basados en los gradientes de intensidades son buenos para detectar los bordes de una imagen, pero en nuestro caso primeramente no se detectó con cada borde por separado la totalidad del borde celular. Al haber un tipo de “iluminación” hacia la célula, ésta tenía regiones que iban de blanco a negro y de negro a blanco. Esto hacía que pueda detectar solamente la mitad del borde celular.

Los modelos de knn son específicos para la imagen en análisis, no se pueden extrapolar a otras imágenes. Se tendría que reentrenar otro modelo distinto para tener una buena precisión en la detección.

Al requerir mucha intervención por parte del usuario y un tiempo de procesamiento muy lento, no es práctico el uso de este tipo de modelo. Se podría utilizar los resultados obtenidos de este trabajo para realizar un dataset con los bordes “taggeados” y entrenar una red neuronal similar a la “Red Neuronal SAM” (una versión más simple) que apunte a esta detección de borde celular.

Links

Drive: <https://drive.google.com/drive/u/0/folders/1ayJoCLxuT5hZpxrSjD9CoXyf53G1dn17>