



PRÁCTICA EXTRA

DETERMINAR EL NÚMERO MÍNIMO DE SUPERPÍXELES ÓPTIMO

Sara Giménez Gómez

E.T.S.INFORMÁTICA – GRADO EN ING.DE LA SALUD

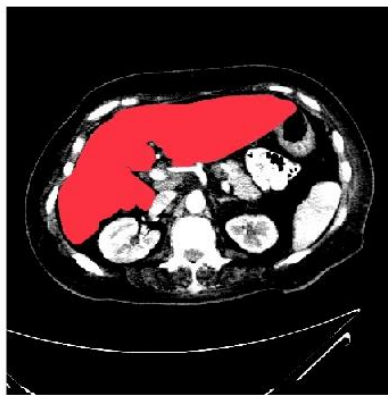
PRÁCTICA EXTRA

DETERMINA EL NÚMERO MÍNIMO DE SUPERPÍXELES ÓPTIMO

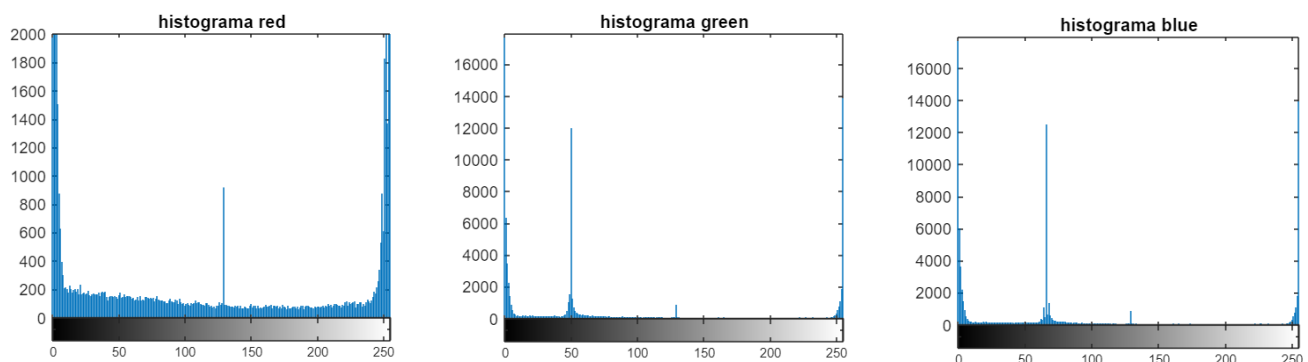
PRIMERA PARTE:

Lo primero que vamos a realizar será una segmentación manual del hígado. Esta técnica es muy utilizada actualmente sobre todo para entrenar a futuras Inteligencias Artificiales a la detección de anomalías.

Tras la segmentación manual visualizamos la ventana de estudio de la siguiente forma:

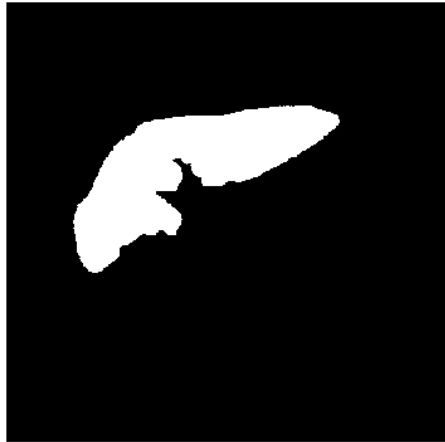


Estamos trabajando con RGB, por lo que vamos a tener que pasar esto a binario para poder trabajar con ella. Se hizo un estudio de sus histogramas para poder hacer una especie de umbralización, pero usando los tres canales disponibles:

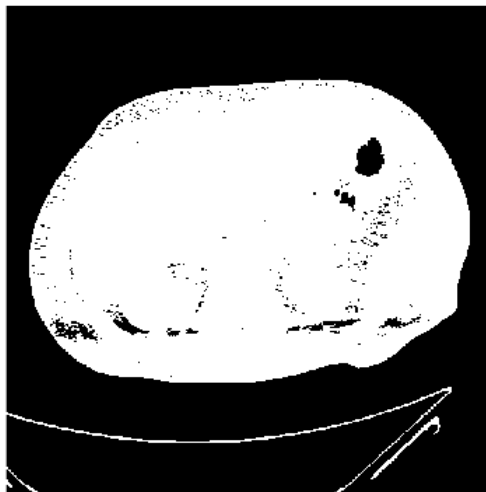


R- red, G-green y B-blue. Los valores utilizados fueron visualizados en los anteriores histogramas. El umbral seleccionado para el canal rojo fue de 150 y para el resto de los canales 100. De este modo cualquier píxel que esté por encima del valor 150 en la imagen RGB será pintado en blanco y cualquier color por debajo de este o de los 100 mencionado en los canales Green y Blue.

Tras esta binarización de la imagen RGB conseguimos lo siguiente:



Nuestra imagen original al que se le aplicó una binarización muy básica es esta:



Esta primera parte del proyecto que acaba al pasar de RGB a binaria la imagen, de esta forma ya somos capaces de trabajar con ambas imágenes.

SEGUNDA PARTE:

En esta segunda parte trabajaremos en dos conceptos claves, queremos buscar un número óptimo de superpíxeles para dividir, es decir, segmentar nuestra imagen. Y luego calcularemos como de bueno es nuestra segmentación manual frente una imagen binarizada dada.

Optimización del Número de superpíxeles

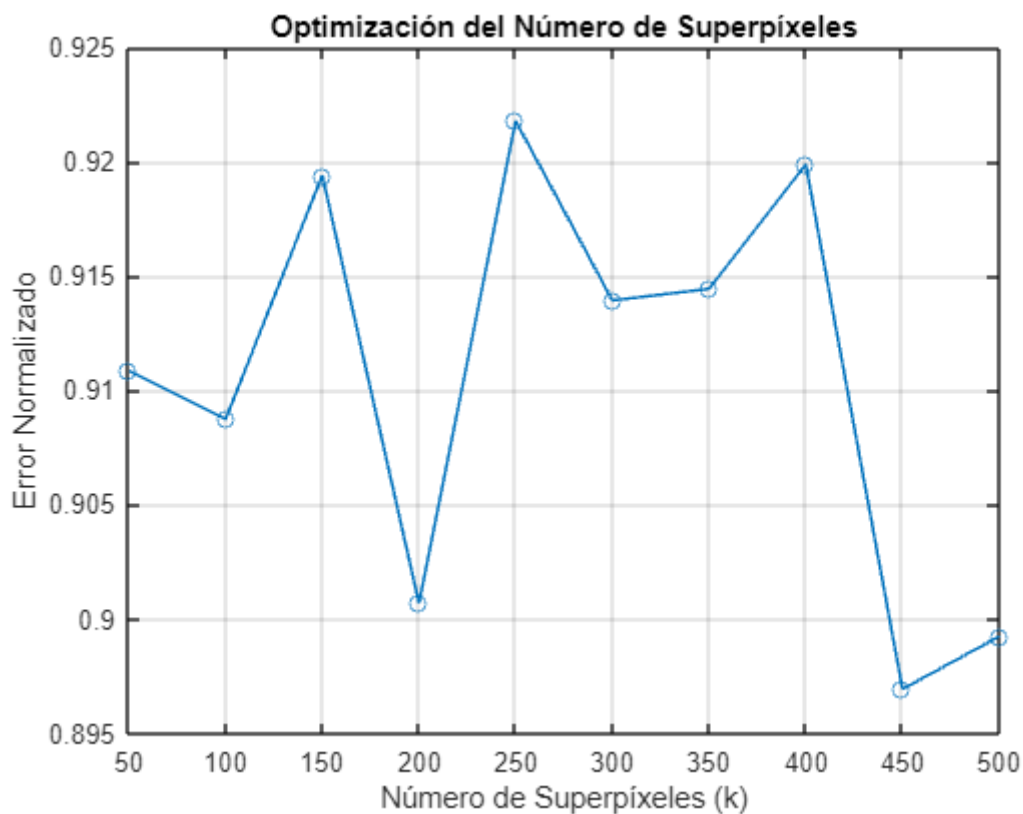
Queremos encontrar el número de k óptimos para los superpíxeles. Para ello vamos a realizar los siguientes pasos:

1. creamos un rango de diferentes valores de superpíxeles(k)
2. Usamos superpíxel para dividir la imagen binarizada cambiando los distintos valores de k.

Con la matriz de etiquetas sacamos una matriz del tamaño de la imagen, donde cada píxel tiene un número que indica a qué superpíxel pertenece. Cada número es como una isla en la imagen.

3. Evaluamos el error para cada superpíxel.
Para cada etiqueta calculo cuantos de esos pixeles están dentro de la región de referencia. Sumo este error al error total para k.

Esta fue la idea para el algoritmo usado, lo que obtuvimos fue lo siguiente:



Si interpretamos esta gráfica vemos que el número óptimo de superpíxeles es 450 con un error de 0.89692. Cuanto mayor es el error peor alineado están los superpíxeles.

El error es una medida de cuánto se equivocan los superpíxeles al representar las regiones de interés. Un superpíxel perfecto debería estar totalmente contenido en una sola región de la imagen de referencia. Consideramos error cuando un superpíxel está fuera o medio dentro y medio fuera.

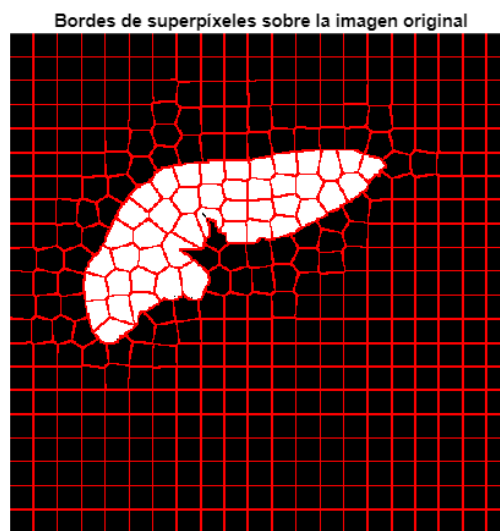
Matemáticamente el código hace esto:

$$\text{Error} = 1 - \frac{\text{Intersección(aciertos)}}{\text{Total de píxeles del superpíxel}}$$

- Si todos los píxeles del superpíxel están dentro de la región de referencia, el error es 0.
- Si algunos píxeles están fuera, el error será un valor entre 0 y 1, dependiendo del porcentaje que esté mal.

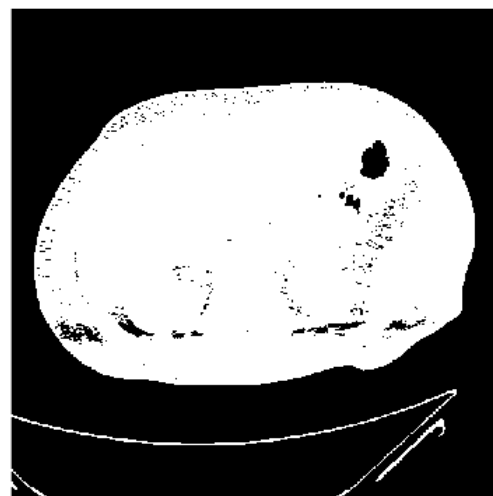
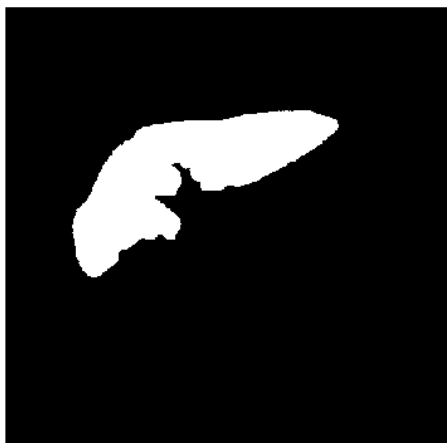
El error total es el promedio de los errores de todos los superpíxeles.

Visualmente quedaría así:



Cálculo de métricas con la imagen binarizada básica:

Una vez hemos sacado la imagen superpíxel deseamos saber cómo de buena fue nuestra segmentación manual respecto a la imagen original. Recordemos que actualmente tenemos lo siguiente:



La primera imagen recordemos que era la segmentación manual binarizada y la segunda imagen no es más que una binarización poco trabajada de la imagen original.

Con estas dos vamos a calcular el AND lógico de tal forma que conseguiremos:

- VP (Verdaderos Positivos): Píxeles donde A y B son 1.
- VN (Verdaderos Negativos): Píxeles donde A y B son 0.
- FP (Falsos Positivos): Píxeles donde A es 0 y B es 1.
- FN (Falsos Negativos): Píxeles donde A es 1 y B es 0.

Siendo A la imagen segmentada manualmente y B la imagen original.

Los resultados obtenidos fueron:

VP = 23456	
VN = 143347	Precision: 0.20457
FP = 91203	Recall: 0.99996
FN = 1	F1-Score: 0.33966

Para hallar la precisión aplicamos la fórmula:

$$\text{Precision} = \frac{\text{Verdaderos Positivos}}{(\text{Verdaderos Positivos} + \text{Falsos Positivos})}$$

Con esta precisión tan baja el algoritmo tiene muchos falsos positivos. Esto refleja que la imagen original tiene mucha cantidad de regiones adicionales fuera del hígado.

Para calcular el Recall:

$$\text{Recall} = \frac{\text{Verdaderos Positivos}}{(\text{Verdaderos Positivos} + \text{Falsos Negativos})}$$

Vemos un recall bastante alto, es decir detecta la mayor parte del hígado presente. Podemos traducir esto en que todos los píxeles de la máscara manual están incluidos en la imagen original. Esto es positivo porque significa que no se omiten áreas relevantes del hígado.

Por último, la formula usada para F1-score:

$$\text{F1 - Score} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

Conseguimos combinar ambas métricas, como tenemos una precisión bastante baja a pesar del gran valor de Recall es de esperar un valor bajo de este balance.

Me gustaría comprobar que pasaría si el AND lógico se hiciera con una imagen un poco más trabajada.

Cálculo de métricas con la imagen binarización más trabajada

Durante la práctica 3 del curso hicimos una umbralización manual haciendo un estudio previo de su histograma para así poder buscar un rango de valores en los que la imagen tenga un “mejor aspecto”.



Aquí visualmente se ve mejor la ventana de interés, es decir el hígado.

Si hiciéramos de nuevo el cálculo de las métricas obtendríamos lo siguiente:

VP = 22815
VN = 195401
FP = 39149
FN = 642

Precisión: 0.3682
Recall: 0.97263
F1-Score: 0.53418

Como conclusión, aunque la precisión sigue siendo baja, ha mejorado respecto al primer resultado. Esto indica que el modelo reduce la cantidad de falsos positivos. Además, el recall sigue siendo muy alto, lo que sugiere que la mayoría de los píxeles del hígado reales han sido detectados correctamente.

Pero realmente el valor que más me interesa a la hora de comparar es el F1-score. A diferencia del cálculo de la métrica anterior ahora nos encontramos un valor intermedio que refleja el balance entre precisión y recall. Esto se debe a la mejora de la precisión por supuesto.

Como conclusión de esta prueba que realicé quería comprobar y demostrar que una mejor binarización reduce el ruido y aumenta la capacidad del modelo para distinguir entre las áreas relevantes, en este caso el hígado, y no relevantes.

CONCLUSIÓN:

Al principio de la práctica quise recalcar la importancia de la buena segmentación en el campo de las imágenes biomédicas. A lo largo del trabajo, se buscó determinar el número óptimo de superpíxeles para lograr una segmentación adecuada, aplicando diversas métricas que evaluaron la calidad de las segmentaciones manuales en comparación con diferentes técnicas de umbralización.

Demostrando también que una binarización más trabajada y ajustada reduce significativamente el ruido mejorando así la precisión y permite al modelo identificar con mayor exactitud las regiones relevantes, como el hígado, frente a las no relevantes.