

Laboratorio 1 - Introducción al procesamiento de imágenes

Mariana Rodríguez Serrano
Universidad de los Andes
m.rodriguez21@uniandes.edu.co

Rafael Camilo Tejón Rojas
Universidad de los Andes
rc.tejon@uniandes.edu.co

1. Introducción

Un computador representa las imágenes como una matriz (o un conjunto de 3 matrices) de enteros, en la cual cada casilla de la matriz indica que intensidad de color hay que mostrar en cada píxel de una imagen. Utilizando estos valores se puede crear el histograma de una imagen, este nos muestra como se distribuyen los colores de una imagen. Este hecho permite segmentar partes una imagen utilizando mascarar que únicamente muestren cierta parte de interés de una imagen.

2. Histograma de una imagen

1. **¿Qué pasa si usa el comando `plt.plot` para visualizar la imagen?** Se va a intentar graficar la matriz mediante la cual se encuentra representada la imagen. Dado que esta matriz tiene mas de 2 dimensiones (pues consiste de 3 matrices 2D, una por cada canal) se genera un error al intentar construir la gráfica. En la Figura 1 se observa el error generado.

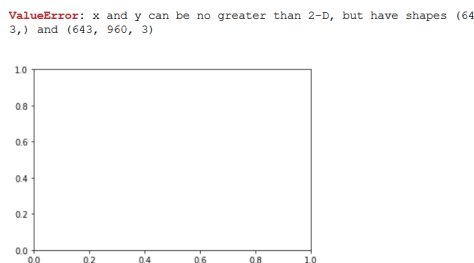


Figure 1. Error obtenido al intentar visualizar la imagen usando el comando `py.plot`

2. **¿Cómo se representa una imagen en un computador?** Usualmente se utiliza una representación conocida como bitmap. Consiste en dividir la imagen original en un conjunto finito de puntos, llamados píxeles, tomados a intervalos regulares. Cada uno de estos píxeles es representado por medio de uno o mas números y el resultado final es una matriz numérica. Para la representación de cada píxel, la más común es

el sistema RGB en el cual se usan tres números entre 0 y 255.

3. **¿Qué diferencias existen entre una imagen a color y una en escalas de grises?** En una imagen a escala de grises cada píxel se representa usando un valor numérico individual que representa su iluminación o brillo, en una escala que se extiende entre blanco y negro [1].
4. **¿De qué clase son estas estructuras (utilice la función `type`)? ¿Hay otra clases posibles para representarlos?** Cada estructura es un Array (`imageio.core.util.Array`) cuyas entradas son de tipo entero (`numpy.uint8`). Se podrían representar con cualquier otra clase de tipo numérico como `uint16`, `double` o `single`.
5. **¿Cuáles son las ventajas y desventajas de usar las clases `uint8`, `uint16`, `double` y `single`?** Lo que realmente genera una diferencia al usar cualquiera de estas clases para representar enteros es el número de bits que se usan para el almacenamiento. Por ejemplo, `uint8` utiliza 8 bits, permitiendo representar números del 0 al 255 ($2^8 - 1$), mientras que `uint16` utiliza 16 y permite representar del 0 al 65535 ($2^{16} - 1$) [2]. La ventaja de usar con más bits es que se puede representar una mayor cantidad de colores diferentes. Sin embargo, la desventaja es que el almacenamiento por píxel va a ser mayor.
6. **¿Qué representan los niveles de gris en las imágenes de los canales de color?** Entre más cercano sea a blanco el nivel de gris, quiere decir que ese canal tiene mayor iluminación (brillo) en la imagen original [1].
7. **¿Cuál es la diferencia entre los formatos disponibles para guardar las imágenes ('jpeg', 'tiff', 'png' y 'dicom')?** JPEG es un formato de imagen comprimida que omite información que no puede ser detectada por el ojo humano. TIFF es un formato altamente extensible, que permite anadir información por medio del uso de etiquetas. PNG es un formato comprimido pero sin pérdida de información. Una imagen en

formato PNG por lo general es mas grande que una en JPEG y mas pequeña que la TIFF. Finalmente, DICOM es un formato diseñado específicamente para la industria médica. Se parece a TIFF pues presenta extensión por medio de etiquetas. Sin embargo, estas extensiones son más que todo información asociada a la imagen (i.e. nombre y fecha de nacimiento del paciente, médico a cargo, entre otros) mas que características adicionales [3].

8. **¿Qué es el histograma de una imagen?** El histograma de una imagen es una gráfica en la que cada barra representa el número de píxeles presentes con cada intensidad de color [4]. La intensidad suele ir de 0 a 255 si es una imagen a color, o de 0 a 1 para una imagen en escala de gris.

9. **¿Cómo funciona rgb2gray?** Esta función convierte cada pixel con componentes RGB en un solo componente de escala de gris mediante una suma ponderada de los componentes Rojo (R), Azul (B) y Verde (G) originales [5]. Los pesos de la suma ponderada son así:

$$GS = 0.2125R + 0.7154G + 0.0721B$$

10. **¿Observa alguna diferencia entre el histograma de la imagen a color vectorizada y el de la imagen en escala de grises?** Si, como se aprecia en la figura 2, las gráficas no tienen el mismo comportamiento.

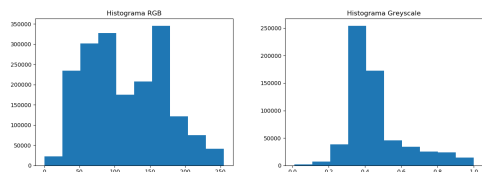


Figure 2. Comparación de los histogramas para RGB y escala de gris

¿Tiene sentido el histograma de la imagen a color obtenido al mezclar la información de los tres canales de color en un vector? Esto tiene sentido teniendo en cuenta que no todos los componentes de RGB tienen el mismo peso al hacer la conversión a escala de gris. El verde tiene mucho más peso que los otros dos y por eso, el comportamiento del histograma de escala de gris va a ser más parecido al del canal verde que a los tres canales juntos.

3. Segmentación por medio del histograma de una imagen

1. **¿Cómo es posible que la imagen tenga canales de color pero se vean en escala de grises?** Es posible

representar una imagen de escala de grises utilizando los 3 canales del modelo RGB. Esto es debido a que si usas una misma intensidad lumínica en los 3 canales se generará una tonalidad de gris.

2. **¿Qué pasa cuando el número de bins es máximo? ¿cuál es el efecto de disminuir el número de bins?** Los bins hacen referencia a la cantidad de columnas del histograma, por lo tanto el máximo es el máximo valor que puede tomar un píxel (255), cuando este es máximo habrá una columna por cada valor que puede tomar un píxel. Al disminuir los bins, estamos haciendo que haya columnas que no representen un único valor, sino, un rango de valores, por lo cual se pierde exactitud de la distribución de las intensidades.

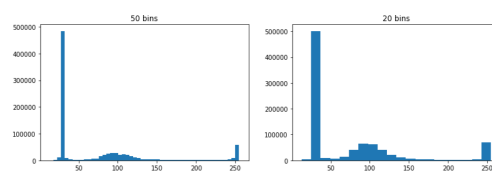


Figure 3. Comparación de mascarar con distinta cantidad de bins

En la figura 3 se puede observar que el histograma con 50 bins explica mejor el comportamiento de la imagen con respecto al de 20 bins

3. **¿Cómo cambia la máscara de acuerdo a los umbrales que se usan?** Dependiendo de que umbrales escojamos estaríamos permitiendo que más intensidades cumplan con las condiciones necesarias para pertenecer a la máscara. Esto causa que los píxeles con valor 1 en la imagen sean más que con umbrales más bajos. Esto se puede observar en la figura 4 donde se puede observar que la máscara con umbrales más separados permite que muchos más píxeles sean de color blanco.

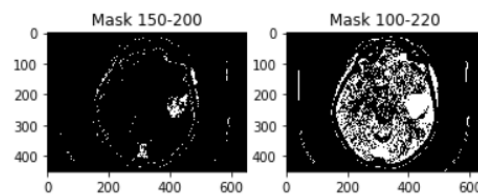


Figure 4. Comparación de mascarar con distintos umbrales

4. **¿Por qué es posible segmentar los pulmones utilizando el histograma? ¿Sería más difícil hacerlo para algún tejido? Si hay ruido en la máscara (elementos que no hacen parte del ACV) ¿a qué se debe?** Es posible porque el histograma nos muestra cuáles son las intensidades más y menos presentes en

la imagen. Por lo cual, podemos acotar cuales son los píxeles que nos interesa que hagan parte de la imagen y quitar los que no. Seria mas difícil hacer este proceso cuando dos tejidos producen intensidades muy parecidas el la imagen lo que causaría que haya ruido en la imagen como se observa en ambas mascarar de la figura 3.

4. Problema Biomédico: Imágenes de tomografías

1. **¿Porqué vmin y vmax permite que las tomografías se vean mejor?** Esto se debe a que la función imshow se auto ajusta a el valor máxima y mínimo de la data. Entonces si el mínimo y el máximo están muy cercanos el contraste entre estos no se notaran. Por eso, si obligamos que el mínimo y el máximo estén muy alejados aumentara el contraste[6]
2. **¿En qué eje observa más separación de cortes?** En el eje Z se observan más separación entre cortes, esto se debe a que existen mayor cantidad de cortes en el eje Y que en el eje Z

Realizado en L^AT_EX

References

- [1] 9.3 Sistemas de representación. Zator Systems. Tomado de: https://www.zator.com/Hardware/H9_3.htm
- [2] Understanding Data Types in Python. Tomado de: <https://jakevdp.github.io/PythonDataScienceHandbook/02.01-understanding-data-types.html>
- [3] Tan L. (2006). Image file formats. Biomedical imaging and intervention journal, 2(1), e6.
- [4] Atienza, V. (s.f.) El histograma de una imagen digital. Universidad Politécnica de Valencia. Departamento de Informática de Sistemas y Computadores.
- [5] RGB to grayscale. Tomado de: https://scikit-image.org/docs/dev/auto_examples/color_exposure/plot_rgb_to_gray.html
- [6] matplotlib.pyplot.imshow. Tomado de: https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.imshow.html