
Práctica 1 (Python):

Introducción al procesamiento de imágenes

1. Objetivos

- Familiarizar al estudiante con las librerías de Python para el procesamiento de imágenes.
- Aprender que un volumen es sólo una colección de cortes.
- Procesar todas las imágenes de un directorio.
- Aprender a leer imágenes DICOM con la librería pydicom.
- Reconstruir volúmenes a partir de los cortes.
- Observar la anisotropía de los volúmenes obtenidos de TACs

2. Parámetros de entrega

Sicua

- Adjuntar un único archivo (a menos que se especifique lo contrario) con los códigos de la siguiente forma: *main_CódigoEstudiante#1_CódigoEstudiante#2.m*.
- Adjuntar **TODO** el código que se entregó en el item anterior, en formato *.txt*. Llámelo de igual manera: *CódigoEstudiante#1_CódigoEstudiante#2.txt*. Esto con el fin de poder evaluar en Sicua automáticamente cualquier intento de copia o similitud entre los algoritmos. Cualquier intento de copia será tratado de acuerdo al reglamento de la Universidad. Aquel grupo que no incluya este item en su entrega tendrá una nota de 0 en el laboratorio.
- Adjuntar un único archivo con el informe en PDF con el mismo nombre del primer item: *CódigoEstudiante#1_CódigoEstudiante#2.pdf*. El informe debe presentarse con el formato CVPR (para más información del informe ver sección de Parámetros de calificación).
- En el caso de que no tenga compañero, **DEBE** utilizar un cero como el código de su compañero, i.e. *main_CódigoEstudiante_0.m* para el archivo de matlab, *CódigoEstudiante_0.pdf* para el informe y *CódigoEstudiante_0.txt* para el archivo de texto.

- **NO** se permiten archivos comprimidos tales como *zip*, *rar*, *7z*, *tar*, *gz*, *xz*, *iso*, *cbz*, etc. Aquel grupo que envíe su informe como un archivo comprimido no tendrá calificación.
- No se recibirá ningún archivo por algún medio diferente a Sicua plus.

3. Algunas reglas

- La asistencia a la sección de laboratorio inscrita es **obligatoria**. Acorde con el Reglamento General de Estudiantes de Pregrado de la Universidad de los Andes, la inasistencia a más del 20 % de las clases resultará en la reprobación de la materia.
- Tendrán una semana para realizar cada práctica (Informe, código, etc.). La guía se entregará el viernes anterior a la sesión y el plazo de entrega será hasta las 11:59 pm del domingo de la semana de la sesión. El vínculo para el envío del laboratorio dejará de estar disponible luego de esta hora.
- Los informes deben realizarse **únicamente** con la pareja. Esto quiere decir que aunque es válido discutir los problemas con sus compañeros, la solución y el código, deben ser de su completa autoría. Está prohibido copiar literalmente el algoritmo y/o procedimiento desarrollado por otro grupo. Si llega a obtener un código de internet, asociado al problema a resolver, este debe estar debidamente referenciado y usted debe entenderlo por completo.

4. Parámetros de calificación

Resultados

1. Todos los códigos deben mantener orden y coherencia en la ejecución de comandos, es decir, cada vez que se muestre una figura, el programa debe esperar para que se presione una tecla, para así continuar con la siguiente y así sucesivamente (para esto utilice en Matlab `pause`) o en Python `input ("Press Enter to continue...")`. Tenga en cuenta que si se quieren contrastar dos imágenes use `subplot` o `imshowpair`.
2. Toda figura debe llevar su título y descripción en el informe.
3. El código debe estar debidamente comentado.
4. NUNCA utilice rutas absolutas para leer o guardar archivos. Este es el error más común en la ejecución de los códigos.
5. Para generar rutas utilice `fullfile` en Matlab o `os.path.join` en Python ya que puede que corramos los laboratorios usando Linux o Windows y los separadores de archivos cambian dependiendo del sistema operativo.
6. Asuma que dentro de la carpeta de ejecución del código se encuentran los archivos necesarios para el laboratorio.

Ejemplo: dentro del código principal el estudiante quiere leer la imagen `im.png` que está dentro de una carpeta de imágenes en la misma ruta que el `main`.

6.1. Forma incorrecta:

En Matlab:

```
imread('C:/Estudiante/docs/ElLab/ims/im.png').
```

En Python:

```
skimage.io.imread('C:/Estudiante/docs/ElLab/ims/im.png')
```

6.2. Forma correcta:

En Matlab:

```
imread(fullfile('ims','im.png')).
```

En Python:

```
skimage.io.imread(os.path.join('ims','im.png'))
```

Informe

Todos los laboratorios deben realizarse en formato CVPR. Para aquellos que desean realizarlo en L^AT_EX, pueden obtener una plantilla del formato en el siguiente link. Cabe resaltar que los informes del laboratorio no deben contener ninguna sección de artículo científico, esto significa que no deben incluir ninguna división como resultados, abstract o conclusiones. Por consiguiente, deben responder únicamente a las preguntas del informe. Recuerden incluir imágenes de sus resultados y documentarlas debidamente. Por último, el informe tiene una longitud máxima de 2 páginas. Se pueden incluir imágenes como anexos pero las imágenes principales deben ser parte del informe.

Bonos

Cada grupo ganará puntos que le suben la nota por cada una de las siguientes características que se cumpla:

1. Usar Python, sea ya en *Notebooks* de *Jupyter*, o en scripts convencionales de Python.
2. Desarrollar el informe en L^AT_EX. (Aquellas personas que lo desarrolle en L^AT_EX, deben escribir al final del informe "**Realizado en L^AT_EX**". De lo contrario no se contará el bono. Los grupos que intenten reproducir la frase en un informe realizado en Word tendrán 0 en la nota de dicho laboratorio. Para poder escribir el logo utilice el comando "\LaTeX" en su informe de latex.

Estos puntos se asignarán de acuerdo al criterio de los profesores.

5. Procedimiento

Importe las siguientes librerías: `numpy`, `requests`, `skimage` y `matplotlib`. Para mayor orden es conveniente importar todo al comienzo del script. Los paquetes pueden importarse usando abreviaciones para simplificar el código por ejemplo, `import numpy as np`. Además, uno puede importar únicamente una función de un paquete por ejemplo, `from skimage.color import rgb2gray`. Para este laboratorio utilizarémos `io` de `skimage`, `rgb2gray` de `skimage.color` y `pyplot` de `matplotlib` (este usualmente se importa así: `import matplotlib.pyplot as plt`). A continuación está un ejemplo del código para importar las librerías:

```
# package imports
import numpy as np
import requests # for saving images from the internet
import skimage.io as io
from skimage.color import rgb2gray
import matplotlib.pyplot as plt
```

5.1. Histograma de una imagen (30 %)

1. Descargue una imagen a color de internet utilizando el comando `requests.get()` de esta forma:

```
r = requests.get(image_url)
with open(image_name, "wb") as f:
    f.write(r.content)
```

2. Cargue la imagen en Python usando el comando `skimage.io.imread` y asígnela a una variable. Para visualizar la imagen use `plt.imshow` acompañado de `plt.show()`. ¿Qué pasa si usa el comando `plt.plot` para visualizar la imagen.
3. ¿Cómo se representa una imagen en un computador? ¿Qué diferencias existen entre una imagen a color y una en escalas de grises? ¿De qué clase son estas estructuras (utilice la función `type`)? ¿Hay otra clases posibles para representarlos? ¿Cuáles son las ventajas y desventajas de usar las clases `uint8`, `uint16`, `double` y `single`?
4. Use la imagen a color (color: $M \times N \times 3$) y visualice cada uno de los canales (junto con la imagen original) en un `plt.subplot` de 2×2 . Para ver cada canal en escala de gris utilice la opción `cmap='gray'` de `plt.imshow`. Guarde la imagen generada por el `subplot` mediante `plt.savefig`. ¿Qué representan los niveles de gris en las imágenes de los canales de color? ¿cuál es la diferencia entre los formatos disponibles para guardar las imágenes ('jpeg', 'tiff', 'png' y 'dicom')?
5. Obtenga el histograma de la imagen a color convirtiéndola en un vector y usando `plt.hist` (si tiene problemas con el número de dimensiones de la imagen, use la función `flatten()` de `numpy`). También queremos obtener el histograma de la imagen en blanco y negro, para esto primero convierta la imagen a escala de grises con la función `rgb2gray` y calcule el histograma de esta con `hist` (asegúrese que el rango del histograma corresponde al rango de valores de la

imagen y que la imagen sea un vector). ¿Qué es el histograma de una imagen? ¿cómo funciona `rgb2gray`? ¿observa alguna diferencia entre el histograma de la imagen a color vectorizada y el de la imagen en escala de grises? ¿tiene sentido el histograma de la imagen a color obtenido al mezclar la información de los tres canales de color en un vector?

5.2. Segmentación por medio del histograma de una imagen (30 %)

Los histogramas se pueden utilizar para separar objetos de interés dentro de una imagen. En la Figura 1 se encuentra una imagen con su respectivo histograma.

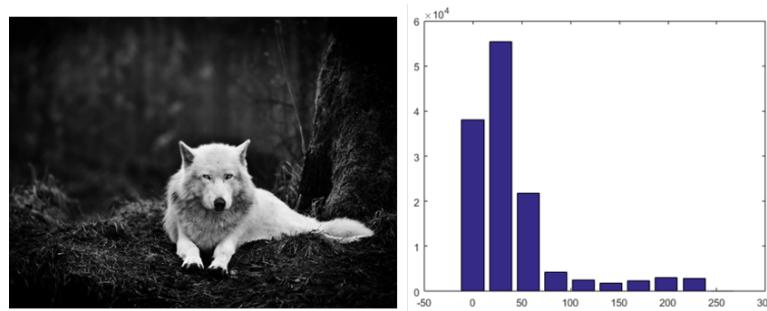


Figura 1: Imagen en niveles de gris y su respectivo histograma con un número de *bins* diferente a 256.

El histograma proporciona la información adicional necesaria para crear una máscara del objeto, es decir, una imagen binaria (de 1s y 0s) que asigna una etiqueta positiva (1) únicamente a los píxeles del objeto. Haciendo uso de la máscara, es posible aislar el objeto del resto de la imagen. En la Figura 2, se muestra como en la imagen de la izquierda la máscara (región seleccionada en amarillo) se aísla en la imagen de la derecha.



Figura 2: Máscara de la imagen en amarillo y extracción de la región de interés utilizando la máscara.

Una de las causas de un accidente cerebrovascular (ACV) es la hemorragia cerebral ocasionada por la ruptura de un vaso sanguíneo en el cerebro. El objetivo, como se muestra en la Figura 3, es segmentar la hemorragia cerebral en la imagen de tomografía (región amarilla), haciendo uso del histograma de la imagen.



Figura 3: (Izquierda) Imagen del accidente cerebrovascular de un paciente causado por una hemorragia intracerebral. (Centro) máscara de segmentación de la hemorragia. (Derecha) segmentación del ACV.

1. Descargue la imagen original de tomografía haciendo click en la figura 3 y guárdela como *acv.jpg*.
2. Al cargar la imagen en Python, si tiene las dimensiones de una imagen a color, quédese con el primer canal ¿cómo es posible que la imagen tenga canales de color pero se vean en escala de grises?.
3. Obtenga el histograma de la imagen con *plt.hist*. Tenga en cuenta que un parámetro importante para variar en el momento de construir el histograma es el número de *bins*. ¿Qué pasa cuando el número de *bins* es máximo? ¿cuál es el efecto de disminuir el número de *bins*?
4. Para crear la máscara seleccione dos umbrales (intensidad de un *bin*). Asígnele una etiqueta positiva (1) a los píxeles cuya intensidad sea superior al umbral menor e inferior al umbral mayor. A los píxeles que no cumplan las condiciones anteriores asígneles una etiqueta negativa (0). Muestre la máscara junto a la imagen original en una figura. ¿Cómo cambia la máscara de acuerdo a los umbrales que se usan?
5. Utilice la máscara para extraer el ACV de la imagen. Muestre la imagen resultante en una figura. ¿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?

5.3. Problema Biomédico: Imágenes de tomografías (40 %)

Usted quiere analizar los resultados de la última tomografía que se le realizó a un paciente, pero descubre que el hospital le ha entregado los datos en desorden (el nombre de la imagen no corresponde al número del corte) y mezclados con tomografías de otros dos pacientes. Su objetivo es leer los cortes y crear tres volúmenes (uno por cada tomografía realizada a un paciente) con los cortes organizados como se muestra en la figura 4.

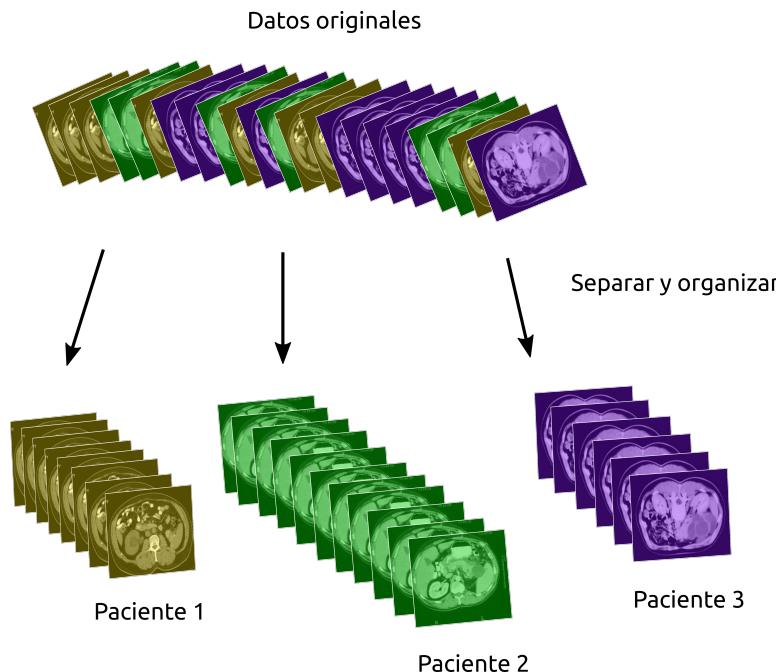


Figura 4: Descripción gráfica del problema biomédico

1. Descargue de [este link](#) la carpeta `mixed_slices` en la misma carpeta de ejecución del código. No es necesario que haga este paso a través de código.
2. Obtenga una lista de los archivos en la carpeta `mixed_slices` con la función `glob` de la librería `glob`, es decir, `glob.glob(os.path.join('mixed_slices', '*.dcm'))`. Esta función devuelve una lista con los archivos y, por tanto, se puede iterar la lista para obtenerlos uno a uno. Para cargar un archivo en particular, use la función `dcmread` de la librería `pydicom`, es decir, `pydicom.dcmread(os.path.join('mixed_slices', 'tac_1.dcm'))`
Cada archivo corresponde a un corte de la tomografía; el número en el nombre del archivo no necesariamente es el número de corte (tenga cuidado con esto).
3. Al cargar un corte en una variable
(`x = pydicom.dcmread(os.path.join('mixed_slices', 'tac_1.dcm'))`), puede ver los metadatos (atributos) del corte al imprimir `x`. Para acceder a un atributo en particular, e.g., el atributo `atributo1`, puede llamarlo de la siguiente manera: `x.atributo1`.
4. Cree tres variables de 3 dimensiones [*filas, columnas, numero de cortes de la tomografía*] `vol1`, `vol2` y `vol3` en las que guardará todos los cortes. El paciente 8 tiene 123 cortes, el 10 tiene 110, y el 26 tiene 114. Use los metadatos para obtener valores del número de filas y columnas de cada imagen (las tres tomografías coinciden en estos valores).
Puede inicializar las variables en ceros con la función `zeros` de la librería `numpy`.
5. En un ciclo, lea cada corte con `pydicom.dcmread` y ubíquelo en el volumen correspondiente y en la posición de corte indicados por los metadatos. En los metadatos hay información para obtener tanto el número del corte como el nombre del paciente.

6. Para cada tomografía visualice cada corte en el eje z usando un ciclo y las funciones `ion()`, `show()`, `imshow()`, `draw()`, `pause()` y `clf()` de la librería `matplotlib.pyplot`. Por ejemplo, para una serie de gráficas:

```
import numpy as np
import matplotlib.pyplot as plt

plt.ion()
plt.show()
for idx, num in enumerate(np.linspace(0.1, 1, 100)):
    plt.plot([1, 2, 3], [num, np.exp(num), np.log(num)])
    plt.title(f'Iteration {idx} - number {num}')
    plt.draw()
    plt.pause(0.001)
    plt.clf()
```

7. Si no logra visualizar ningún corte correctamente, limite la escala de gris con los parámetros `vmin` y `vmax` de la función `imshow` (de `matplotlib`) entre, por ejemplo, 0 y 2000. Analice por qué esto mejora la visualización
8. Visualice los cortes de la tomografía más reciente del paciente en el eje y. ¿En qué eje observa más separación de cortes?