Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

# Computer Vision

*Práctica 3. Segmentación*

Autores:
Víctor Gallardo Sánchez (801159)
Nerea Gallego Sánchez (801950)

22 de abril de 2023

# Índice

# 1. Introducción

Se presenta la memoria del trabajo realizado para la práctica L3. En ella se va a exponer las tareas realizadas y los conceptos aprendidos durante la realización de la misma.

# 2. Visualize the imaging dataset

En este apartado se pide visualizar algunas imágenes para obtener una idea inicial sobre cómo son los datos.
Para ello se ha usado la función *subplots* de la librería *matplotlib* y se han mostrado las ímagenes de los datos en cuadrado tal y como se puede ver en la imagen 1

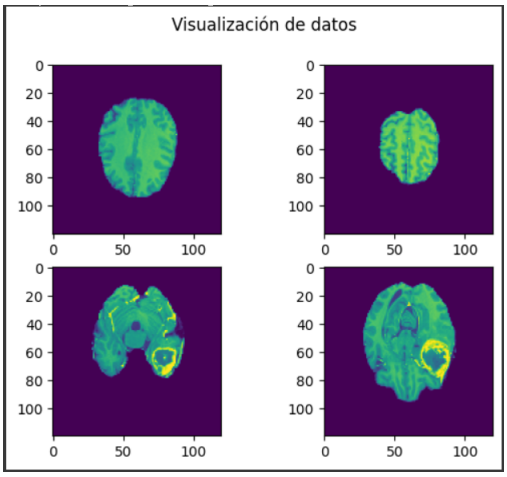


Figura 1: Ejemplo de visualización de los datos

# 3. Implement a dataset class

En este apartado se pide completar la clase *BrainImageSet*. Primero es necesario inicializar todos argumentos de la clase vacíos. A continuación se inicializan las imágenes y las etiquetas correspondientes. También se pide rellenar el resto de funciones de la clase. Una de las funciones devuelve el tamaño de datos que contiene. A continuación, la siguiente imagen devuelve la imagen con un determinado ínidice y normaliza su intensidad. Por último, la funtión *get_random_batch* devuelve un conjunto de imágenes aleatorias de un tamaño dado.

# 4. Build a U-net architecture

Se ha implementado esta arquitectura siguiendo [1]. Para ello se completó cada capa con las operaciones correspondientes.
Un aspecto a destacar es que las flechas descendientes de la arquitectura se intentaron hacer con max-Pool 2x2 pero dio problemas. Es por eso que se realizó con conv2d con el parámetro stride = 2. Esto corresponde a realizar MaxPool 2x2 y la siguiente conv2D.

# 5. Train the segmentation model

Para entrenar el modelo descrito en el apartado anterior fue necesario añadir el optimizador de tipo *Adam*. Por último, en cada iteración se han añadido el criterio de loss, la backpropagation y añadir un step al optimizador.

# 6. Visualise the segmentation results

Se han decidido mostrar 5 resultados de la segmentación y se muestran en las imágenes 2, 3, 4, 5 y 6.

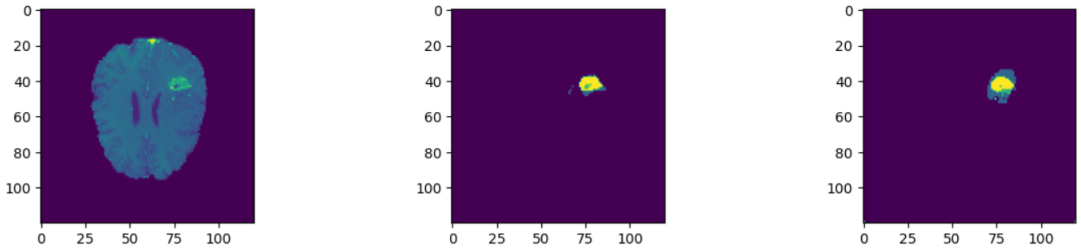


Figura 2: Segmentación 1

Como se puede observar cada imagen contiene 3 sub-imágenes. Estas corresponden a la imagen real, la segmentación dada por la label y la segmentación obtenida por el modelo. Cabe destacar que son 5 imágenes aleatorias que se obtienen con la función get_random_batch.

# 7. Evaluate the segmentation results

Para evaluar los resultados se ha utilizado la medida Dice Similarity Coefficient (DSC). Se han tomado 200 muestras aleatorias del conjunto de imágenes de test. Se ha tomado esta decisión porque google colab no permitía cargar todas las imágenes de test en memoria. Se ha calculado por cada imágen las etiquetas más significativas (1 - edema, 2 - non-enhancing tumour y 3 - enhancing tumour) y se ha calculado por un lado la intersección de píxeles que coinciden en la imagen de labels y en la segmentada por el modelo y por otro lado la longitud de píxeles de las distintas etiquetas para ambas imágenes (tanto label como la segmentada por el modelo). Se ha hecho la suma de las intersecciones con distintas etiquetas y se ha dividido por la suma de las longitudes con distintas etiquetas, obteniendo así la medida DSC por cada imagen.

A continuación se pide elegir tres imágenes con distintos valores de DSC representativos y compararla con un método de segmentación tradicional.
Se han probado distintos métodos de segmentación tradicional y finalmente se ha optado por utilizar segmentación por threshold.
La conclusión que se pueden obtener viendo ambos métodos es que segmenta bastante mejor el método de deep learning que el método tradicional. Este método de deep-learning consigue extraer mejor los resultados que se buscan, se adapta mejor ya que es un modelo que se va entrenando con gran cantidad de datos para que realice la segmentación lo mejor posible. Sin embargo, el método tradicional se basa únicamente en un treshold sencillo, por lo que muchas veces segmenta muchas más partes de la imagen de las que se buscan.
En las imágenes 7, 8 y 9 se pueden observar ejemplos de labels (imagen de la izquierda), segmentación con deep-learning (imagen del medio) y segmentación con threshold (imagen de la derecha).

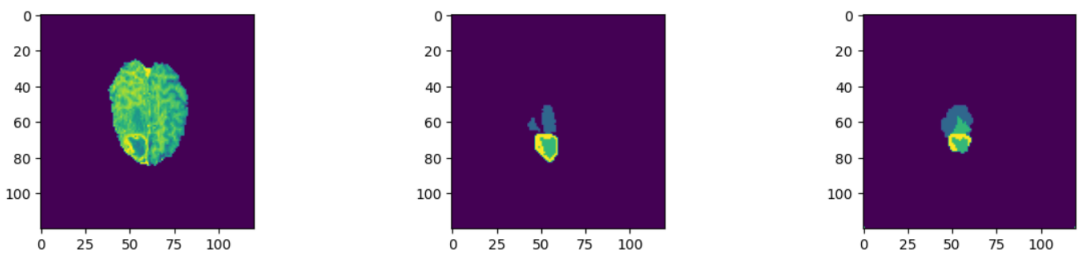


Figura 3: Segmentación 2

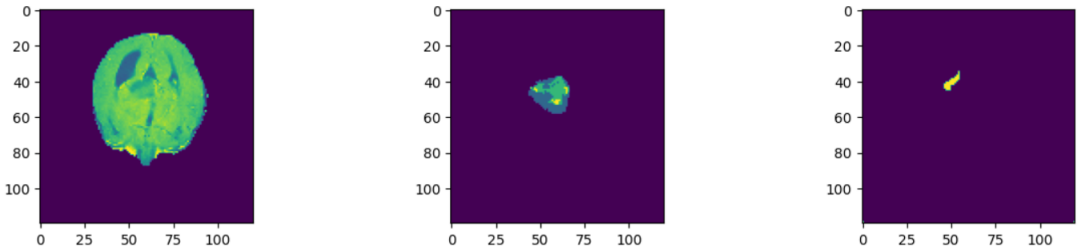

Figura 4: Segmentación 3

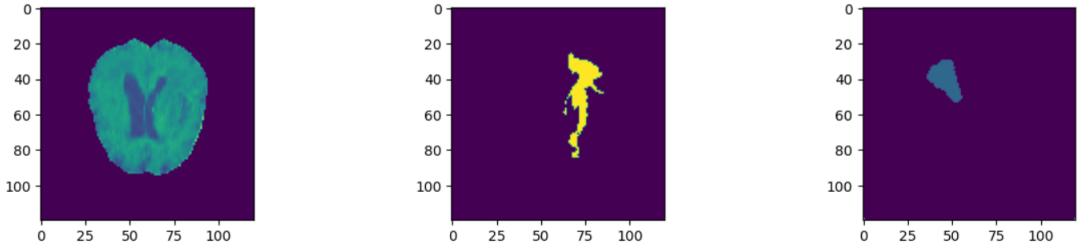

Figura 5: Segmentación 4

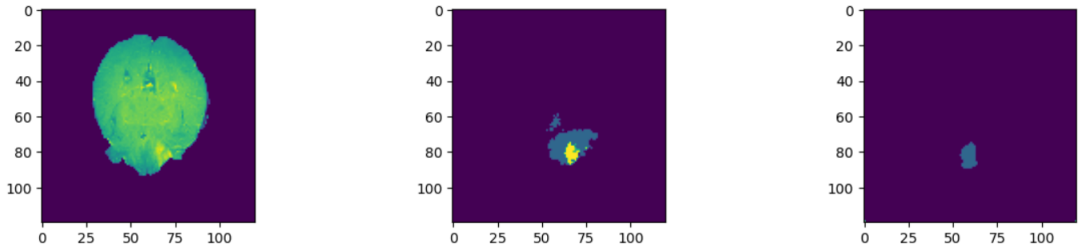

Figura 6: Segmentación 5

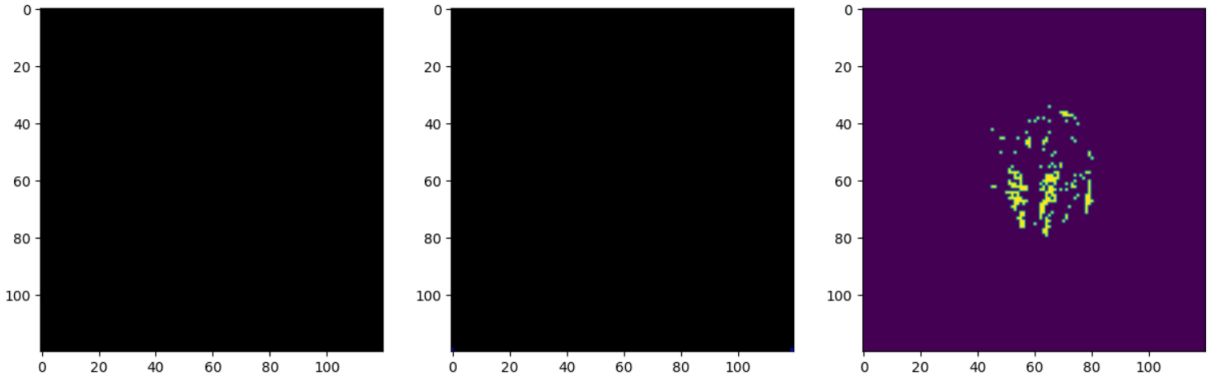

Figura 7: Segmentación tradicional vs deep learning 1

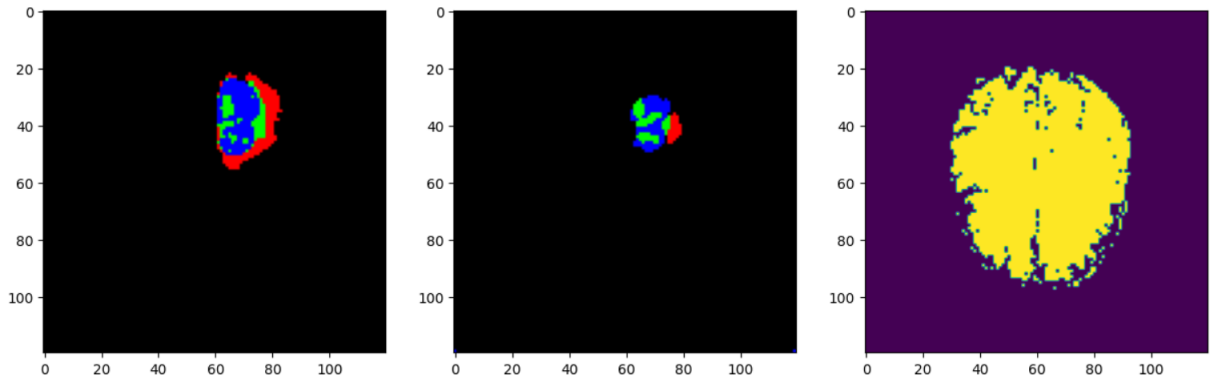

Figura 8: Segmentación tradicional vs deep learning 2

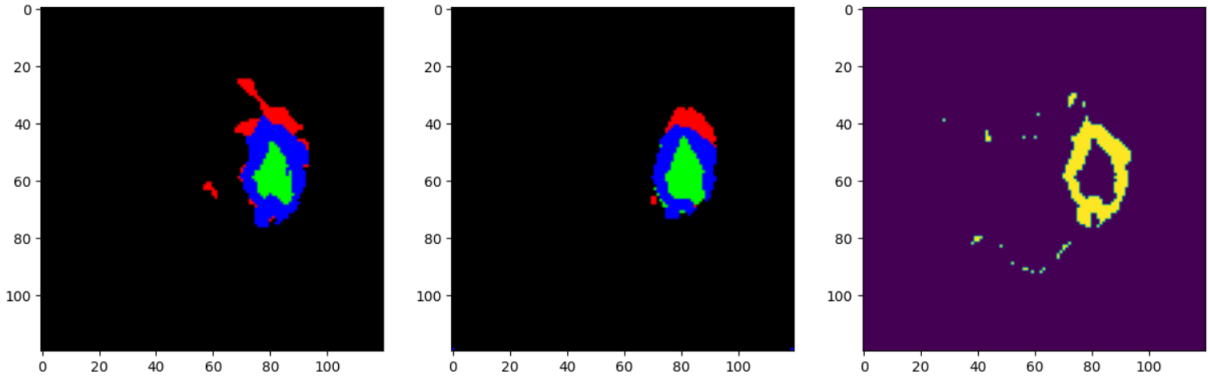

Figura 9: Segmentación tradicional vs deep learning 3

## 8. Esfuerzos dedicados

**Víctor**

| Día | Hora | Tareas |
|---|---|---|
| 27/03/2023 | 3 | Sesión de prácticas L3 |
| 04/04/2023 | 1,5 | fixing unet |
| 08/04/2023 | 1,5 | fixing unet |
| 11/04/2023 | 1,2 | fixing |
| 15/04/2023 | 1,5 | evaluation |
| 22/04/2023 | 1,00 | memoria |
| Total | 9,7 horas | |

**Nerea**

| Día | Hora | Tareas |
|---|---|---|
| 27/03/2023 | 3 | Sesión de prácticas L3 |
| 04/04/2023 | 1,5 | fixing unet |
| 05/04/2023 | 0,5 | fixing unet |
| 11/04/2023 | 1,20 | evaluating unet |
| 11/04/2023 | 0,62 | fixing unet |
| 14/04/2023 | 0,8 | evaluating unet |
| 15/04/2023 | 1,5 | evaluating unet |
| 15/04/2023 | 0,28 | evaluating unet |
| 16/04/2023 | 0,33 | memoria |
| 22/04/2023 | 1,00 | memoria |
| Total | 10,73 horas | |

## Referencias

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F.

Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.