



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Departamento de sistemas informáticos y
computación

Segmentación 3D de imagen médica con MONAI

Trabajo prácticas Redes neuronales artificiales

Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas
e Imagen Digital

Autor

Francisco Javier Gil-Terrón Rodríguez

2021 - 2022

Tabla de contenidos

1. Introducción	3
1.1 Motivación y objetivos.....	3
1.2 Monai y formato de imagen NIfTI.....	3
2. Desarrollo.....	5
2.1 Conjunto de datos.....	5
2.2 Topología de red	6
2.3 Configuración	6
2.4 Resultados.....	7
3. Evaluación mejor modelo	9

1. Introducción

1.1 Motivación y objetivos

El presente trabajo versará, en esencia, sobre segmentación tridimensional de órganos a partir de imagen médica, y más específicamente, segmentación en 3D de bazo a partir de archivos NIfTI.

La elección de esta categoría en concreto, segmentación de bazo, y no otro tipo de segmentación automática como podría ser reconocimiento automático de tumores, se ha hecho debido a que esta opción es más didáctica, reproducible y ofrece una mayor libertad a la hora de experimentar.

Además, cabe destacar que el objetivo que se persigue con este trabajo no es el de alcanzar los resultados que hay actualmente en el estado del arte (que rondan el 95% de precisión en esta tarea concretamente), sino explorar un ámbito nuevo de aplicación para redes neuronales convolucionales junto a un formato de datos que no había visto hasta ahora, como es el de NIfTI.

Incluso ofreciendo resultados relativamente malos, es más fácil para un profesional encargado de la anotación semántica (radiólogo generalmente en este ámbito) partir de una solución pre-segmentada y corregirla, que segmentarla a mano, por lo que en ciertas aplicaciones podría ser permisible un 'amplio' margen de error.

1.2 Monai y formato de imagen NIfTI

[Monai](#) es un Framework open-source basado en PyTorch y especializado en el campo del Deep learning en imagen médica, tanto en tareas de clasificación como de segmentación.

En cuanto al conjunto de datos que se va a emplear, como ya se ha citado, estará formado por archivos NIfTI. Pese a que el estándar en imagen médica sea el formato DICOM, Monai trabaja con archivos de este tipo, que se caracterizan por estar compuestos por secuencias de imágenes 2D, que adicionalmente pueden ser vistas desde tres planos o perspectivas: axial, sagital y coronal; además de otra metainformación. La cantidad de planos o imágenes 2D que contienen estas secuencias depende del órgano o sección con la que se esté trabajando, pero suelen ser del orden de entre 50 y 300.

A continuación, se muestra un ejemplo de NIfTI (en este caso, de una muestra cerebral), donde se visualiza uno de estos planos desde las tres perspectivas que se han citado:

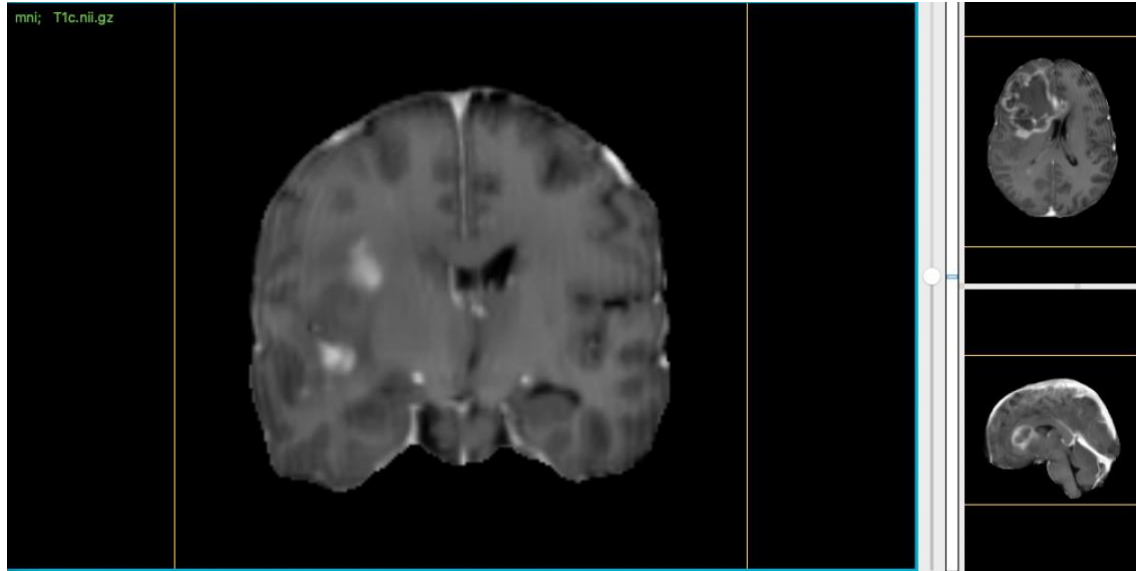


Figura 1 – Ejemplo de plano contenido en archivo NIfTI

Así pues, un único archivo NIfTI podría descomponerse en decenas o incluso centenares de imágenes 2D, lo que se debe tener en cuenta a la hora de valorar el tamaño del conjunto de datos que se vaya a emplear.

2. Desarrollo

2.1 Conjunto de datos

El dataset que se ha empleado esta formado por un total de 41 archivos 3D (NIfTIs), 32 serán empleados para entrenamiento y los 9 restantes para test. Pueden parecer pocos, pero serán suficientes para esta pequeña demostración teniendo en cuenta que la segmentación de un órgano completo no es tan compleja como lo puede ser la segmentación de fragmentos de tejidos necróticos y tumorales. De hecho, en la misma UPV se han creado modelos para la segmentación automática de gliomas con tamaños del orden de 100 muestras.

En la imagen que sigue, se muestra un ejemplo de plano o imagen 2D contenido en uno de los archivos NIfTI que se han empleado, junto a su etiqueta, en este caso, la correcta segmentación del bazo.

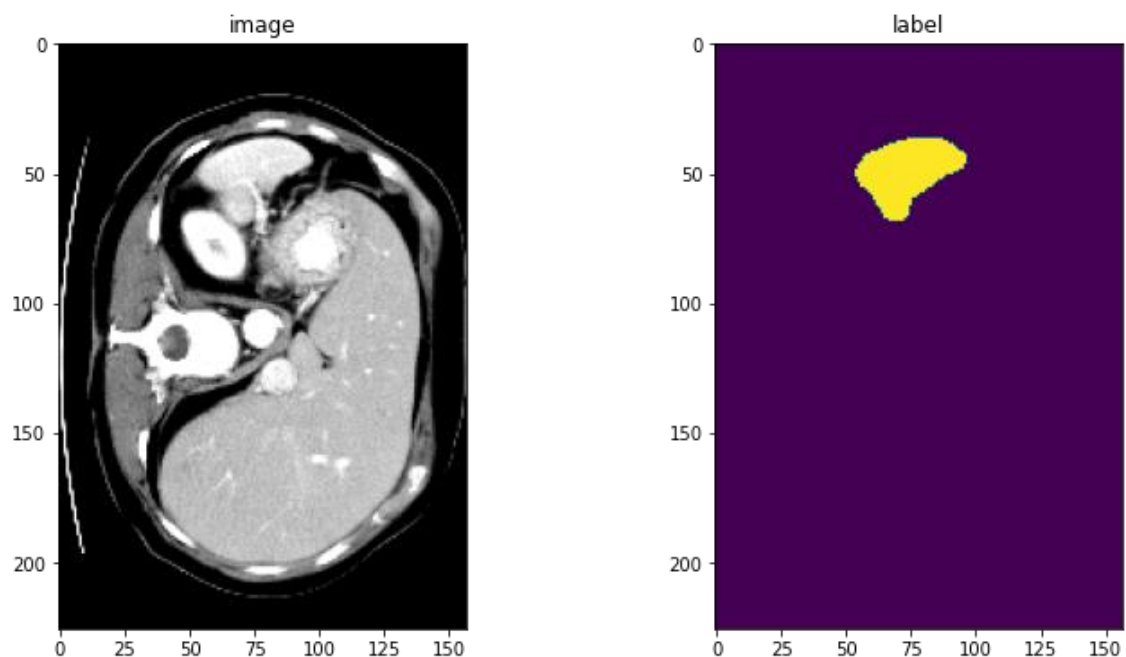


Figura 2 – Ejemplo de muestra

2.2 Topología de red

El tipo de red que se va a emplear para el proyecto es una Unet, topología ampliamente usada en el ámbito de la segmentación automática (al menos en el campo de la imagen médica).

La red ha sido creada con un constructor de Monai y se ha simplificado con respecto a la red original propuesta para esta tarea, es decir, se han eliminado capas para hacerla más accesible y como se verá en el capítulo de resultados, las prestaciones obtenidas son bastante aceptables.

La estructura se compondrá principalmente por bloques de una convolución seguida de Batch Normalization, Dropout y función de activación PReLU. Mientras que la versión original se componía de 18 de estos bloques, en la que se ha usado para el proyecto se ha reducido el tamaño a 12 bloques. Adicionalmente se utilizan dos unidades residuales para evitar la pérdida de gradiente.

Hay que tener en cuenta que tanto las convoluciones como los Batch Normalization son 3D.

Cabe destacar que, puesto que Monai está basado en PyTorch, se podría crear perfectamente la red con PyTorch nativo y con unas pequeñas modificaciones hacerlo funcionar con Monai.

2.3 Configuración

Por lo que respecta a los hiperparámetros que determinarán el aprendizaje, inicialmente se limitará a 200 epochs el entrenamiento (ajustado a lo máximo que permite Google Colab en un día).

Además, se utilizará un batch size de 2 debido a la limitación computacional de la que se dispone, aunque en casos prácticos de esta misma tarea emplean valores del orden de 4-8.

En cuanto al criterio con el que se medirá la precisión del modelo, puesto que se trata de una tarea de segmentación y no de clasificación, se utilizará el coeficiente de Dice, estadístico que mide la superposición de dos superficies y se utiliza para evaluar modelos de segmentación semántica, como es el actual.

En lo referente a Data Augmentation, únicamente se empleará el equivalente a Random Crop que proporciona Monai para imagen en 3D. Se han probado

otras técnicas de Data Augmentation que se emplearon durante las prácticas, como Horizontal Flip y traslaciones, pero no han tenido un impacto demasiado relevante en los resultados finales, por lo que se han eliminado del modelo final.

El resto de las transformaciones que se utilizan tienen la finalidad de preparar los datos y metadatos para ser procesados correctamente, como, por ejemplo, alinear los datos para tratarlos como elementos 3D, eliminar la parte exterior de las imágenes que no se corresponde con órganos o tejidos, y pasar los datos a tensores.

2.4 Resultados

Con lo anteriormente citado, los resultados que se obtuvieron en el entrenamiento se pueden ver reflejados en las siguientes gráficas, donde se puede ver el loss y precisión (o Dice) en cada iteración.

Como era de esperar debido al pequeño valor del batch size, los resultados pueden no ser consistentes a corto plazo produciendo valores distantes entre epochs consecutivos, pero a largo plazo la tendencia es que el modelo mejore con las iteraciones.

El algoritmo de optimización que se recomienda para esta tarea, y con el que se llega al accuracy de 95% es Adam con factor de aprendizaje de $1e-4$, pero para llegar a este resultado son necesarias al menos 500 epochs y limitando a 200 se obtienen resultados bastante malos tal como se puede ver en la siguiente gráfica:

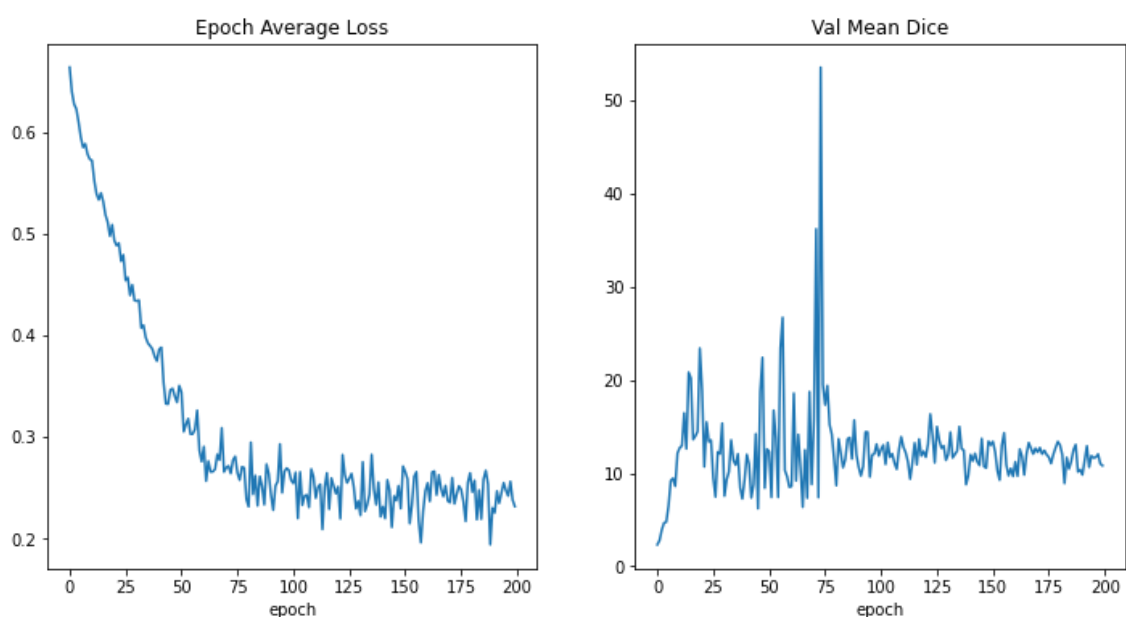


Figura 3 – Loss y Dice obtenidos con Adam

Para tratar de compensar la limitación computacional se trabajará con otro algoritmo optimizador que converja antes, aunque obtenga peores resultados a largo plazo, por lo que se ha empleado SGD con factor de aprendizaje 0.1 y scheduler en 3 etapas con gamma 0.1 (también se probó learning rate scheduler ReduceLROnPlateau con paciencia 5 y factor 0.5, pero se obtuvieron peores resultados). A continuación, se muestran los resultados obtenidos:

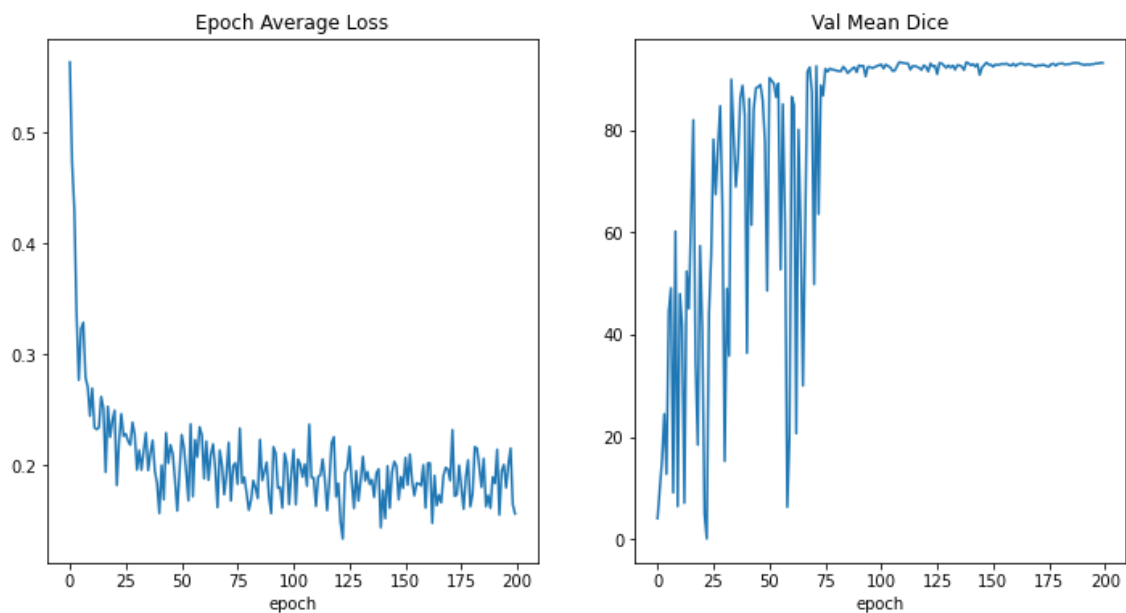


Figura 4 – Loss y Dice obtenidos con SGD

Como se puede ver en las gráficas, los resultados se alejan de alcanzar el 95% que se lograba en el estado del arte, no obstante, son bastante aceptables teniendo en cuenta que se ha simplificado considerablemente la red y se han limitado drásticamente el número de epochs para el entrenamiento del modelo.

3. Evaluación mejor modelo

Finalmente, con el mejor modelo obtenido en el capítulo anterior, se mostrarán algunos de las segmentaciones que produce junto a la etiqueta o segmentación esperada para ver gráficamente los resultados.

De manera resumida se recuerda que la configuración con la que se ha alcanzado estos resultados es una Unet con 10 bloques convolucionales y 2 unidades residuales, optimización con SGD con factor de aprendizaje 0.1 y scheduler en 3 etapas con factor 0.1.

Así pues, a continuación, se muestran tres muestras aleatorias que ha segmentado dicho modelo:

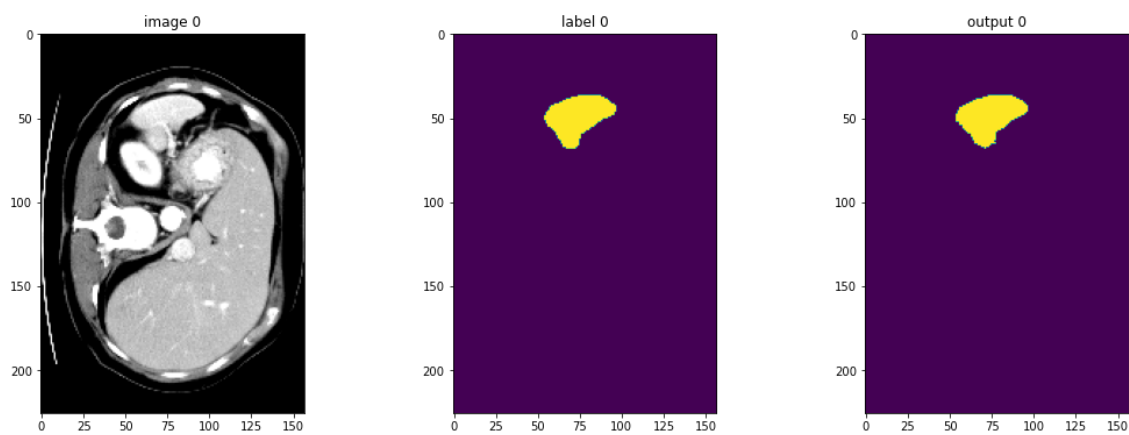


Figura 5 – Ejemplo 1 de segmentación de nuestro modelo

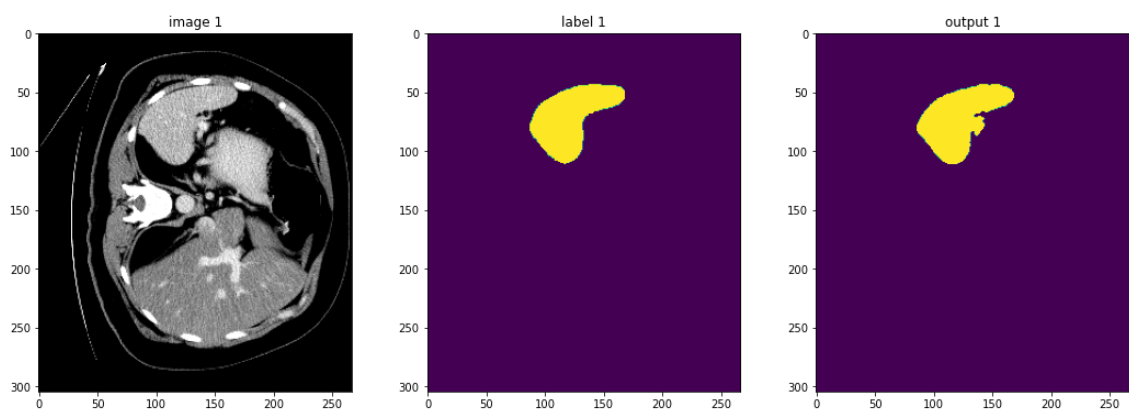


Figura 6 – Ejemplo 2 de segmentación de nuestro modelo

Segmentación 3D de imagen médica con MONAI

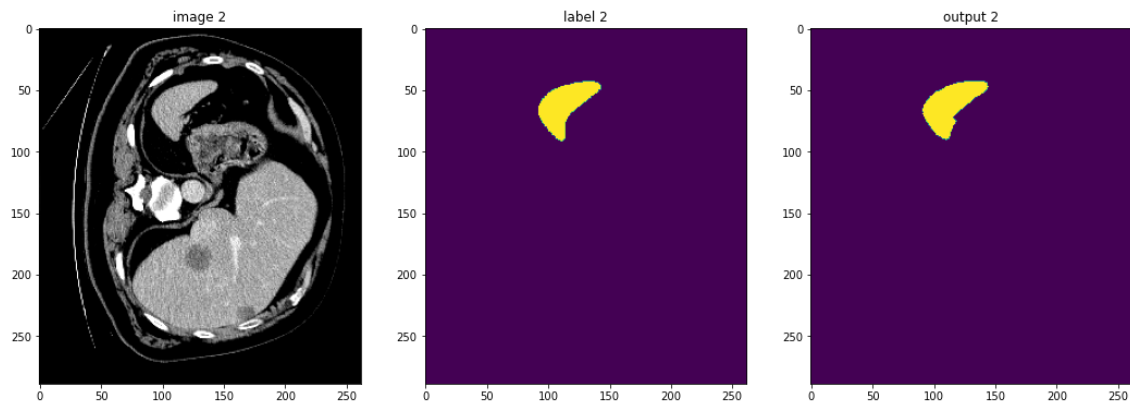


Figura 7 – Ejemplo 3 de segmentación de nuestro modelo

A la vista de las imágenes anteriores, los resultados distan de ser perfectos, pero se aproximan bastante a los resultados esperados y los errores que produce serían fácilmente corregibles por un profesional familiarizado con este tipo de imágenes, que es el objetivo que se persigue en tareas de segmentación de imagen médica actuales.