



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

Diseño e implementación de una red neuronal artificial para
la mejora de la resolución de imágenes de RMN cerebral

Trabajo Fin de Grado

Grado en Ingeniería Biomédica

AUTOR/A: Pérez Martínez, María

Tutor/a: Manjón Herrera, José Vicente

CURSO ACADÉMICO: 2022/2023

Agradecimientos

Quiero aprovechar la ocasión para agradecer el apoyo y ayuda de mis padres y familia desde el primer momento en el que decidí irme de casa para seguir estudiando hasta estos últimos días de mi titulación. Asimismo, tampoco puedo olvidar en este momento a todos mis amigos que han hecho de estos cuatro años una experiencia inolvidable aprendiendo entre todos a vivir y sacar las cosas hacia delante a pesar de las dificultades. De mis amigos del pueblo tampoco me puedo olvidar pues ellos son mi segunda familia y uno de mis principales apoyos a pesar de vernos menos de lo que me gustaría. Gracias a todos, en especial este último año donde he necesitado más apoyo. Por otro lado, también agradecer al personal del laboratorio ITACA por darme la oportunidad de realizar el trabajo y sobre todo a mi tutor José Vicente por ayudarme, guiarme y enseñarme durante estos meses de trabajo.

Resumen

La investigación sobre imagen biomédica ha crecido en los últimos años y sigue creciendo en la actualidad. En este trabajo se presenta el diseño de un nuevo modelo de red neuronal que permite la mejora de la resolución de imágenes de RMN cerebral, con el objetivo de obtener imágenes de alta resolución a partir de imágenes de entrada de baja resolución.

En primer lugar, en este trabajo se ha realizado una búsqueda bibliográfica de RMN y el estado del arte de las diferentes técnicas de procesado de imagen, necesario para que cualquier modelo de red neuronal que recibe imágenes obtenga buenos resultados y el estado del arte de los modelos que se usan para realizar superresolución o mejora de la resolución en la actualidad, en este caso SynthSR.

Para continuar se estableció la arquitectura de la red que iba a ser usada con el fin de obtener el principal objetivo, el modelo empleado fue una red U-Net 3D la cual cuenta con capas de convolución y de modificación en la dimensionalidad, teniendo forma de 'U' como indica su nombre. Antes de entrenarla se prepararon las imágenes de baja y alta resolución. Una vez conseguidas estas imágenes se procede a entrenar y tras esto a evaluar la red/redes.

Para evaluar la red se usan diferentes métricas de evaluación como el PSNR, SSIM o la correlación y además se comparó con el estado del arte (SynthSR y SynthSeg) para observar el impacto de la reconstrucción en la calidad de la segmentación.

En la actualidad, la investigación sobre las diferentes técnicas de tratamiento de imagen médica, en este caso de mejora de la resolución de RMN cerebral, es de gran importancia ya que con estas se puede hacer un diagnóstico precoz y correcto de diferentes patologías que puede mejorar la calidad de vida de una persona.

Palabras clave: RMN, red neuronal artificial, red convolucional, *Deep learning*, resolución, superresolución.

Abstract

Biomedical imaging research has grown in recent years and continues to grow today. In this work we present the design of a new neural network model that allows the improvement of the resolution of brain MRI images, with the aim of obtaining high resolution images from low resolution input images.

Firstly, in this work, a bibliographic search of NMR and the state of the art of the different image processing techniques, necessary for any neural network model that receives images to obtain good results and the state of the art of the models that are currently used to perform super-resolution or improvement of the resolution, in this case SynthSR, has been carried out.

In order to continue, the architecture of the network that was going to be used in order to obtain the main objective was established. The model used was a 3D U-Net network, which has layers of convolution and modification in the dimensionality, having a 'U' shape as its name indicates. Before training it, low and high resolution images were prepared. Once these images have been obtained, the network/networks are trained and then evaluated.

To evaluate the network, different evaluation metrics such as PSNR, SSIM or correlation are used and also compared with the state of the art (SynthSR and SynthSeg) to observe the impact of the reconstruction on the quality of the segmentation.

Nowadays, research on different medical imaging techniques, in this case on improving the resolution of brain MRI, is of great importance as these can be used to make an early and correct diagnosis of different pathologies that can improve the quality of life of a person.

Keywords: MRI, artificial neural network, convolutional network, deep learning, resolution, super-resolution.

Índice de figuras

Figura 1.1. Esquema del proceso de adquisición de una imagen de RM. Fuente: Serreta & Alberich-Bayarri, 2018	19
Figura 1.2. Protones en estado Down (derecha) y estado UP (izquierda). Fuente: Serreta & Alberich-Bayarri, 2018.....	19
Figura 1.3. Precesión de un protón de H, movimiento generado por el campo magnético. Fuente: Serreta & Alberich-Bayarri, 2018.	20
Figura 1.4. Modalidades de la IRM. Potenciación T1 (izquierda), T2 (centro) y densidad protónica (derecha). Fuente: Vilarroig, 2019	21
Figura 1.5. Ejemplo de registro 2D.	22
Figura 1.6. Ejemplo registro MNI152.	23
Figura 1.7. Relación entre Deep Learning, Machine Learning e inteligencia artificial. Fuente: (Gómez Tejada, 2020)	24
Figura 1.8. Esquema de una neurona biológica. Fuente: (Azpilcueta, 2020).	25
Figura 1.9. Esquema analogía entre neuronas biológicas y redes neuronales. Fuente: (Future Space, 2021)	26
Figura 1.10. Modelo de cinco neuronas conectadas, con los pesos, bias y funciones de activación. Fuente: (InteractiveChaos, 2018)	26
Figura 1.11. Capas de las redes neuronales. Fuente: (Carmona Benítez, 2014)	27
Figura 1.12. Ejemplo de overfitting, underfitting y resultado correcto. Fuente: (Rivera Phillips, 2019).	30
Figura 1.13. Esquema de filtrado de imagen en una capa de convolución Fuente: (Silvestre Llopis, 2019).	32
Figura 1.14. Arquitectura Autoencoder. Fuente: (Silvestre Llopis, 2019).....	33
Figura 1.15. Modelo U-Net. Fuente: (Jayawardene, 2020/2021).....	34
Figura 1.16. Ejemplos de la aplicación del algoritmo SynthSR. Fuente: (Iglesias et al., 2021)....	36
Figura 1.17. Ejemplo de uso de SynthSeg, a la izquierda se encuentra la imagen de entrada y a la derecha su segmentación. Fuente: (Billot et al., 2023).....	37
Figura 3.1. La fila superior muestra una imagen simulada de baja resolución donde se pueden observar los artefactos de interpolación. La fila inferior muestra la imagen de alta resolución correspondiente (objetivo).	41
Figura 3.2. Flujo de trabajo.	42
Figura 3.3. Imagen de una red U-Net. Fuente: (López-Lozano, 2021)	44
Figura 3.4. Red DPN aplicada a la segmentación del cerebelo.	45
Figura 3.5. Ejemplo del funcionamiento de data augmentation. Fuente: (Raj, 2018)	45
Figura 4.1. Gráfica valores PSNR, correlación y SSIM, en naranja los resultados obtenidos de las 7 redes entrenadas cada una para una resolución, en azul los resultados de una única red entrenada para todas las resoluciones.	51
Figura 4.2. Gráfica valores de índice DICE.....	54
Figura 4.3. De arriba abajo. Imagen de baja resolución (5 mm) a mejorar. Imagen de superresolución obtenida por SynthSR. Imagen de superresolución obtenida por nuestro método.....	55
Figura 4.4. De arriba abajo. Otro ejemplo de una Imagen de baja resolución (7 mm) a mejorar. Imagen de superresolución obtenida por SynthSR. Imagen de superresolución obtenida por nuestro método.	56

Figura 4.5. De arriba abajo, etiquetas de segmentación de la imagen original HR. Segmentación con SynthSeg de una imagen de 5mm de resolución. Segmentación con SynthSeg haciendo uso del método generado en el trabajo para mejorar la resolución a partir de la misma imagen de resolución de 5 mm.57

Figura 4.6. De arriba abajo, etiquetas de segmentación de la imagen original HR. Segmentación con SynthSeg de una imagen de 7mm de resolución. Segmentación con SynthSeg haciendo uso del método generado en el trabajo para mejorar la resolución a partir de la misma imagen de resolución de 7 mm.58

Índice de tablas

Tabla 1.1. Principales funciones de activación.....	27
Tabla 1.2. Ejemplos de funciones de coste para el entrenamiento de RNA.....	29
Tabla 3.1. Características del equipo usado para los experimentos.	40
Tabla 4.1. Resultados de la correlación de las diferentes arquitecturas.	50
Tabla 4.2. Resultados modificación arquitectura UNET3D.	50
Tabla 4.3. Resultados de una única red entrenada para todas las resoluciones.	51
Tabla 4.4 Resultados para las siete redes entrenadas por separado.	51
Tabla 4.5. Tabla de comparación de resultados entre SynthSR y el método generado en el trabajo.	52
Tabla 4.6. Resultados con la red entrenada con todos las BD.	53
Tabla 4.7. Resultados con la red entrenada con todos las BD más índice DICE.....	53
Tabla 4.8. Comparación de resultados entre SynthSeg y el método generado.....	54
Tabla 7.1. Objetivos de desarrollo sostenible de la agenda de 2030.	62

Parte II

Tabla 2.1. Presupuesto de mano de obra.	68
Tabla 2.2. Presupuesto de software.	69
Tabla 2.3. Presupuesto hardware.....	69
Tabla 3.1. Presupuesto de ejecución material.	70
Tabla 3.2. Presupuesto ejecución por contrata.	70

Lista de acrónimos

TAC, Tomografía Axial Computarizada

PET, *Positron Emission Tomography*

IRM, Imagen de Resonancia Magnética

RM, Resonancia Magnética

RF, Radiofrecuencia

TI, Tiempo de Inversión

H, Hidrógeno

BD, Base de Datos

NC, *Normalized Correlation*

LS, *Least Squares*

MI, *Mutual Information*

SR, *Superresolution*

HR, *High Resolution*

LR, *Low Resolution*

AP, Aprendizaje Profundo

DL, *Deep Learning*

AA, Aprendizaje Automático

ML, *Machine Learning*

IA, Inteligencia Artificial

RNA, Red Neuronal Artificial

ANN, *Artificial Neural Network*

SN, Sistema Nervioso

MSE, *Mean Squared Error*

MAE, *Mean Absoluted Error*

WMAE, *Weighted Mean Absolute Error*

ITK, *Insight Toolkit*

ANTs, *Advanced Normalization Tools*

SISR, *Single Image Superresolution*

PSNR, *Peak Signal-to-Noise Ratio*

SSIM, *Structural Similarity Index*

CNN, *Convolutional Neural Network*

GANs, *Generative Adversarial Networks*

RMN, *Resonancia Magnética Nuclear*

Índice general

- Parte I. MEMORIA
- Parte II. PRESUPUESTOS

Parte I

MEMORIA

Índice

1. INTRODUCCIÓN	18
1.1. RESONANCIA MAGNÉTICA	18
1.1.1. Fundamentos RM.	18
1.1.2. Potenciones.	20
1.2. TÉCNICAS DE PREPROCESADO DE IMAGEN MÉDICA	21
1.2.1. Registro.	21
1.2.2. Superresolución.	23
1.2.3. Segmentación.	24
1.3. <i>DEEP LEARNING</i> Y REDES NEURONALES ARTIFICIALES.	24
1.3.1. Fundamentos generales del <i>Deep Learning</i> .	24
1.3.2. Fundamentos de las Redes Neuronales Artificiales.	25
1.3.3. Entrenamiento y aprendizaje de redes.	28
1.3.4. Evaluación	29
1.3.5. Conceptos de DL	30
1.4. ESTADO DEL ARTE	30
1.4.1. Herramientas de registro.	30
1.4.2. Técnicas de superresolución.	31
1.4.3. Redes neuronales profundas.	32
1.4.4. SynthSR.	34
1.4.5. SynthSEG	36
2. OBJETIVOS E HIPOTESIS	38
3. MATERIALES Y MÉTODOS	39
3.1. MATERIALES	39
3.1.1. Imágenes	39
3.1.2. Hardware	40
3.1.3. Software	40
3.2. METODOLOGÍA	41
3.2.1. Preparación de los datos.	42
3.2.2. Modelo de la red.	43
3.2.3. Data augmentation	45

3.3.	ENTRENAMIENTO DE LA RED	46
3.3.1.	Training.....	46
3.3.2.	Test.....	47
4.	RESULTADOS	50
4.1.	RESULTADOS DEL <i>TRAINING</i>	50
4.2.	RESULTADOS DEL <i>TEST</i>	50
5.	DISCUSIÓN Y CONCLUSIONES	59
5.1.	DISCUSIÓN.....	59
5.2.	CONCLUSIONES	60
6.	LÍNEAS FUTURAS	61
7.	OBJETIVOS DE DESARROLLO SOSTENIBLE	62
8.	BIBLIOGRAFÍA	63

1. INTRODUCCIÓN

La imagen médica es una herramienta crucial en medicina e investigación biomédica. La imagen médica es el conjunto de técnicas y procesos que se usan con el objetivo de generar imágenes del cuerpo humano o partes de este con propósitos clínicos o científicos por lo que ha revolucionado la forma de diagnosticar patologías y planificar y guiar intervenciones quirúrgicas. Dentro de estas técnicas encontramos imágenes TAC y PET ambas con radiaciones ionizantes, imágenes de rayos X, imágenes por ecografía y una de las modalidades más usadas en la actualidad que es la Imagen por Resonancia Magnética sobre la cual se va a desarrollar el presente TFG.

1.1. RESONANCIA MAGNÉTICA

La imagen por resonancia magnética (IRM) es un método tomográfico de emisión cuyas principales ventajas sobre otros métodos de imagen son: la posibilidad de obtener cortes o planos primarios en cualquier dirección del espacio, una elevada resolución de contraste, ausencia de efectos nocivos al no usar radiaciones ionizantes y la elevada versatilidad para el manejo de contraste (Lafuente Martínez & Hernández Moreno, 2006).

En la actualidad esta técnica tiene una amplia relevancia, permite obtener una gran cantidad de información anatómica y funcional de manera 'in vivo', lo que la convierte en una herramienta que sirve para el diagnóstico de numerosas enfermedades con importante impacto social y económico como el cáncer, enfermedades cardiovasculares, neurológicas y del sistema músculo esquelético. Es una modalidad que sigue en expansión tecnológica, sus aplicaciones son cada vez mayores y se encuentran en constante evolución (Díaz, 2014).

1.1.1. Fundamentos RM.

La imagen de Resonancia magnética es una técnica de imagen que se basa en el fenómeno físico de la Resonancia Magnética (RM), es decir se basa en la aplicación de un campo magnético y señales de radiofrecuencia (RF) no ionizantes sobre el paciente (Díaz, 2014).

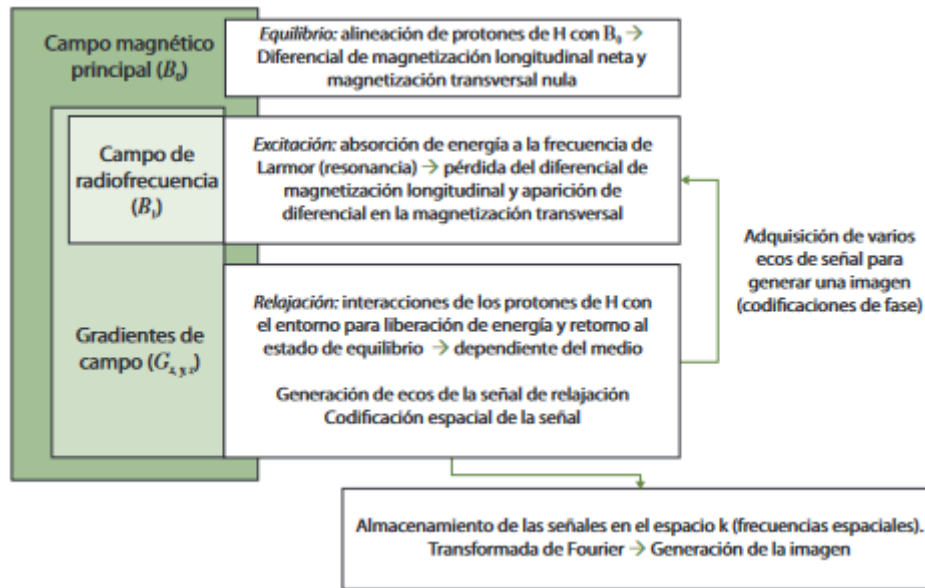


Figura 1.1. Esquema del proceso de adquisición de una imagen de RM. Fuente: (Serreta & Alberich-Bayarri, 2018)

La maquinaria de resonancia está formada por un imán, que se encarga de generar el campo magnético B_0 y por una serie de bobinas de gradiente de campo magnético que emiten los pulsos de radiofrecuencia necesarios. La combinación de ambos elementos produce la excitación de los núcleos de H^1 . El motivo del uso de este núcleo para realizar la RM se debe a que el fenómeno de resonancia se basa en las propiedades magnéticas de este por lo que solo aquellos que tengan un número impar de protones o neutrones serán sensibles, ya que su momento magnético neto será positivo (Serreta & Alberich-Bayarri, 2018).

Como ya se ha mencionado se encuentra el campo magnético B_0 , este se mantiene en funcionamiento de manera continua y tiene una inducción magnética muy elevada (suele estar entre 1.5T y 3T en los equipos clínicos más habituales). Debido a esta inducción los átomos de H se disponen en dos niveles energéticos que pueden observarse en la Figura 1.2: 'Estado UP', el protón se alinea en la misma dirección y sentido que el campo B_0 y 'Estado DOWN', el protón se alinea en sentido contrario y misma dirección al campo B_0 (Serreta & Alberich-Bayarri, 2018). Los núcleos van a contar con un vector de magnetización M (en dirección del campo magnético) por estar bajo la influencia de un campo magnético, a este vector M se le denomina magnetización longitudinal (Serreta & Alberich-Bayarri, 2018).

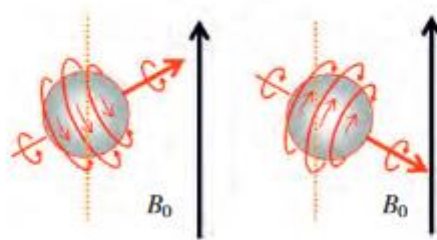


Figura 1.2. Protones en estado Down (derecha) y estado UP (izquierda). Fuente: (Serreta & Alberich-Bayarri, 2018).

A continuación, encontramos las bobinas, estas se encargan de generar un campo magnético de radiofrecuencia denominado B_1 , encargado de generar y originar la excitación. Esta excitación va a provocar dos efectos, el primero es proporcionar energía a los protones y el

segundo hace que los protones entren en sincronismo y comiencen a precesar en fase como en la Figura 1.3. Esta precesión juega un papel fundamental en el fenómeno de resonancia (Serreta & Alberich-Bayarri, 2018).

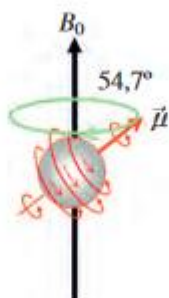


Figura 1.3. Precesión de un protón de H, movimiento generado por el campo magnético. Fuente: (Serreta & Alberich-Bayarri, 2018).

Tras esto se realiza la interrupción del pulso de radiofrecuencia y comienza el proceso de relajación, en el que los protones vuelven a su menor nivel de energía, liberando de esta manera energía y a su vez detectada en forma de corriente eléctrica, la cual va a ser captada por una antena que es capaz de cuantificar los tiempos en los que la magnetización transversal desaparece (tiempo relajación T2) y la magnitud longitudinal se restablece completamente (tiempo relajación T1). Como vemos este proceso de recuperar el estado inicial es la base de la obtención de contrastes diferentes en las imágenes de RM.

Con RMN podemos obtener las imágenes de las zonas sobre las que se quiere llevar a cabo el estudio ya que esta técnica permite hacer la selección de un corte en específico. Por otro lado, si se quiere ver un volumen habrá que excitar dos o más planos de corte para completar la estructura al completo.

1.1.2. Potenciones.

Encontramos una gran variabilidad de contrastes en las imágenes de resonancia magnética, esto se debe a que el contraste obtenido va a depender de la información que queramos extraer.

Para obtener el contraste es necesario hacer uso de pulsos de radiofrecuencia, cuyo propósito es perturbar ese exceso de protones que se encuentran en un menor nivel de energía y que conforman la magnetización longitudinal (Alcañas Martínez, 2015).

Una vez generado el pulso de radiofrecuencia específico y aplicado a los tejidos, se va a generar los efectos que se han comentado en el apartado anterior, estos son la absorción de energía de los portones que hace que la magnetización longitudinal y la sincronidad de estos disminuya generando una nueva magnetización transversal. Por lo que disminuye la magnetización longitudinal y aumenta la transversal, esta última da información de contraste de los tejidos.

Tras el pulso de RF para obtener información de contraste se interrumpe y como se ha explicado con anterioridad comienza la fase de relajación y encontramos dos tipos, la relajación transversal y la longitudinal.

Al tiempo de relajación longitudinal se le denomina T1 y es el tiempo que tarda la componente longitudinal del vector M en llegar al 63% del valor inicial de magnetización. Al tiempo de relajación transversal se le denomina T2 y es el tiempo que tarda la componente transversal del vector M en decaer al 37% respecto de su valor mientras el pulso se estaba emitiendo. Estos tiempos de relajación son únicos para cada tejido y nos van a proporcionar información de la composición. Algunos ejemplos de estas potenciaciones podemos verlos en la figura 1.4.

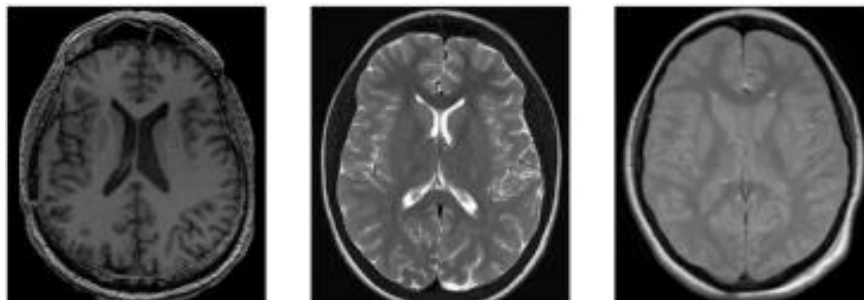


Figura 1.4. Modalidades de la IRM. Potenciación T1 (izquierda), T2 (centro) y densidad protónica (derecha). *Fuente:* (Vilarroig, 2019)

Por lo que lograr un determinado contraste en un tejido va a estar estrechamente relacionado con la potenciación de las secuencias de resonancia magnética utilizadas. En estas secuencias se pueden ir variando determinados parámetros en función de lo que se busque. Las secuencias básicas que se usan son el spin echo y la secuencia de gradiente de echo, con ambas se pueden obtener imágenes potenciadas en T1, T2 o densidad de protones. En este trabajo se va a hacer uso de imágenes T1 en las cuales la grasa aparece hiperintensa y los líquidos hipointensos.

1.2. TÉCNICAS DE PREPROCESADO DE IMAGEN MÉDICA.

Con el fin de obtener un buen análisis de las imágenes obtenidas se debe llevar a cabo una serie de técnicas que nos permitan hacer las imágenes comparables y que sea posible trabajar con ellas. A este conjunto de técnicas se les denomina técnicas de preprocesado, y es un paso fundamental para el éxito del análisis.

El preprocesado puede estar compuesto de varios procesos con el objetivo de mejorar la calidad de las imágenes o normalizar patrones. En el caso de la aplicación médica un buen preprocesado puede significar la detección temprana y eficaz de la enfermedad.

1.2.1. Registro.

El registro se puede definir como una técnica de transformación geométrica que alinea los puntos de un objeto con los correspondientes del mismo u otro objeto con el objetivo de situar todas las imágenes en el mismo espacio de referencia. Por lo que se puede decir que es una técnica de normalización espacial de las imágenes. En este proyecto es importante tener las imágenes registradas en un espacio estándar para poder llevar a cabo un buen análisis.

Las técnicas de registro se pueden clasificar según su modalidad como monomodal o multimodal o según el sujeto como intrasujeto, intersujeto o atlas. Como se ha comentado el registro hace uso de transformaciones geométricas estas pueden ser:

- *Lineales*, dentro de esta encontramos tres subtipos que dependen de los grados de libertad:
 1. Rígida, permite 6 grados de libertad es decir tres translaciones y tres rotaciones.
 2. Global, permite 9 grados de libertad, en este caso a lo anterior le añadimos otras tres escalados.
 3. Afín, este permite 12 grados de libertad, de manera análoga se le añade a lo anterior tres shear.
- *No lineales (warping)*, permite miles de grados de libertad. En estas transformaciones las ecuaciones relativas a las coordenadas de la imagen son no-lineales, estas transformaciones de grado superior permiten cualquier cambio geométrico entre dos imágenes, el problema es que tienen un coste computacional muy elevado.

Para llevar a cabo un buen registro se pueden optimizar diferentes funciones de coste que tienen como objetivo medir la similitud de la imagen de referencia y la imagen a transformar, las cuales pueden ser, correlación normalizada (NC), mínimos cuadrados (LS), *Mutual Information* (MI), Normalized Mutual Information. Cada método tiene sus ventajas con respecto al resto, para realizar los registros es importante establecer una buena estrategia para obtener un desarrollo óptimo y eficiente. Figura 1.5 muestra un ejemplo de registro.

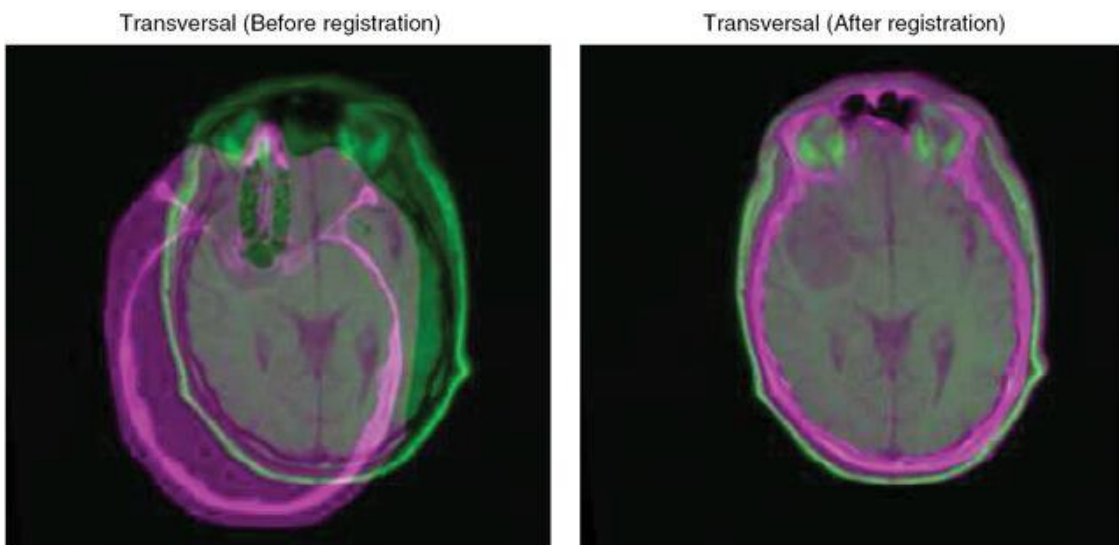


Figura 1.5. Ejemplo de registro 2D.

Un tipo de registro muy utilizado en imagen médica es el registro de las imágenes del paciente desde su sistema de coordenadas original (llamado sistema nativo) a un sistema de coordenadas estandarizado. El más usado en neuroimagen es el sistema MNI152 (Fonov et al., 2011). Este registro es el que se ha usado en este TFG ya que como se verá más adelante el objetivo fundamental es mejorar la calidad de las imágenes en el espacio MNI153. La figura 1.6 muestra un ejemplo de este registro.

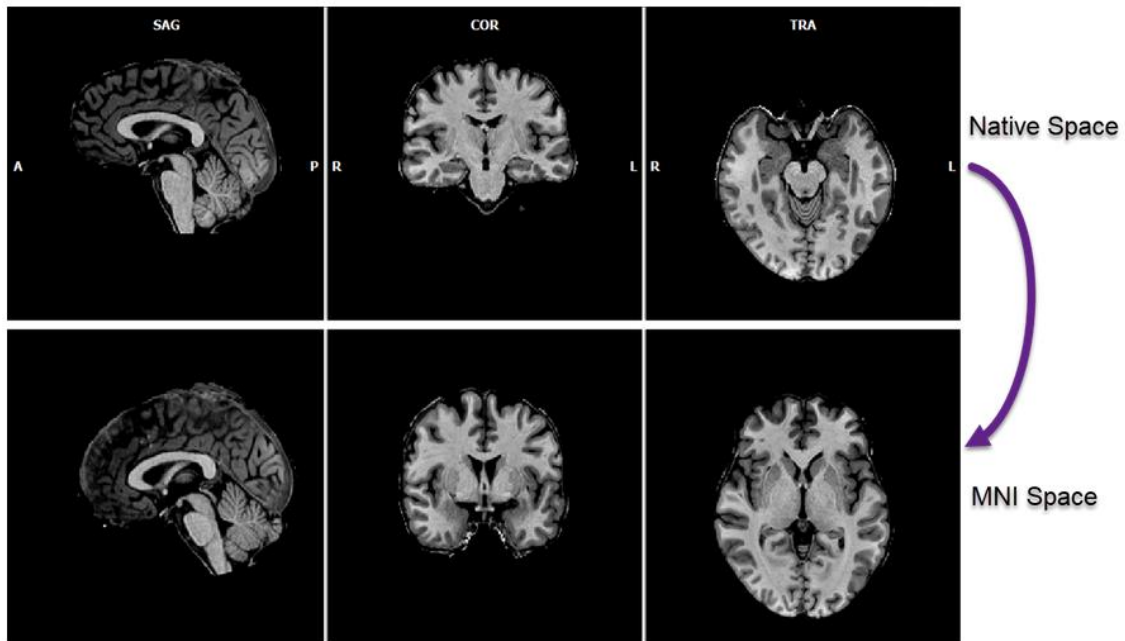


Figura 1.6. Ejemplo registro MNI152.

1.2.2. Superresolución.

Debemos saber que las imágenes médicas tienen una resolución finita, esto se produce debido a que hay limitaciones de adquisición que son generadas por el equipo de resonancia magnética del que se haga uso y también por el confort del paciente. La resolución de la imagen nos indica el nivel de detalle que puede ser extraído de la misma (Marzoa Tanco, 2019).

De esta manera, las técnicas de superresolución (SR) se encargan de inferir los datos de alta resolución (HR) subyacentes a partir de los datos de baja resolución (LR) usando modelos generativos, es decir sobre una imagen de baja resolución le aplicamos una serie de restricciones y obtenemos una estimación de su versión de alta resolución.

Por tanto, se puede definir superresolución como un conjunto de técnicas y algoritmos que se diseñan con el objetivo de aumentar la resolución espacial de una imagen (Marzoa Tanco, 2019). En la actualidad los algoritmos de SR se pueden separar en dos grandes familias:

1. Técnicas multi-imagen: se obtienen las imágenes de HR haciendo uso de una secuencia de imágenes de baja resolución y generalmente se resuelven dos problemas, el de estimación de registro y el de estimar la imagen HR. Se trata de fusionar y extraer la información común de las imágenes (Marzoa Tanco, 2019).
2. Técnicas monoimagen: la imagen obtenida mediante SR viene de la información proporcionada de una única imagen de LR. Este es el más usado ya que no requiere de la adquisición de múltiples imágenes (Marzoa Tanco, 2019).

Las bases para llevar a cabo la reconstrucción de una imagen de alta resolución (HR) a partir de una imagen inicial de baja resolución (LR) son la existencia de información subyacente de alta resolución que, siguiendo una serie de restricciones con un modelo matemático, da lugar a la información HR de la imagen inicial.

1.2.3. Segmentación.

Se puede definir la segmentación como “la extracción de un objeto de interés de una imagen arbitraria” (Duda y Hart, 1973). Es una técnica de tratamiento de imágenes que permite extraer información de las sub-partes de las que está compuesta esta imagen.

En el campo de análisis de imagen médica la segmentación es una de las tareas de mayor importancia. En imágenes de resonancia magnética cerebrales, la segmentación se utiliza para realizar un análisis cuantitativo de las regiones que forman el cerebro y para visualizar las estructuras anatómicas de este (Herrera, 2006). Esta técnica permite, por ejemplo, delimitar zonas patológicas y planificar con antelación una cirugía.

1.3. DEEP LEARNING Y REDES NEURONALES ARTIFICIALES.

1.3.1. Fundamentos generales del *Deep Learning*.

Antes de introducir las redes neuronales se deben introducir los términos relacionados. Aprendizaje profundo (AP) o Deep Learning (DL), aprendizaje automático (AA) o Machine Learning (ML) e inteligencia artificial.

En primer lugar, la inteligencia artificial (IA) se puede definir como cualquier técnica que permita a una máquina imitar la inteligencia humana. A continuación, tenemos el Machine Learning (ML) que consiste en una serie de algoritmos computacionales que pueden aprender de los datos y que mejoran automáticamente con la experiencia. Por último, tenemos el Deep Learning (DL), en este caso también son algoritmos computacionales que son capaces de aprender cantidad de niveles de características o representaciones de elevada abstracción (García, 2020). En la siguiente Figura 1.5 podemos observar cómo se relacionan todos los términos explicados.

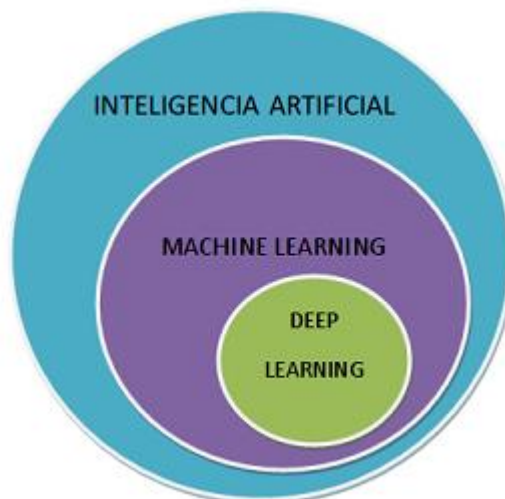


Figura 1.7. Relación entre Deep Learning, Machine Learning e inteligencia artificial. Fuente: (Gómez Tejada, 2020)

Este aprendizaje de características va a seguir una serie de métodos que va a permitir que una máquina sea capaz de aprender. Según la forma de adquirir estas características se pueden diferenciar dos técnicas de aprendizaje automático (Rouhiainen, 2018).

1. Aprendizaje no supervisado, son los algoritmos que no van a usar datos etiquetados (sin la solución) para que se produzca el aprendizaje. El objetivo es obtener patrones ocultos o agrupaciones de los propios datos. Este método no requiere de la intervención humana (Rouhiainen, 2018).
2. Aprendizaje supervisado, son de algoritmos que aprenden a partir de datos etiquetados, es decir en base a ejemplos dato-resolución, en este caso sí se va a requerir la intervención humana para generar la retroalimentación (Rouhiainen, 2018).

Como ya se ha mencionado el Deep Learning es un subcampo del aprendizaje automático y para realizar este aprendizaje se hace uso de redes neuronales artificiales que se desarrollan en el siguiente apartado.

1.3.2. Fundamentos de las Redes Neuronales Artificiales.

Las redes neuronales artificiales (RNA) o *artificial neural networks* (ANNs), son algoritmos que están basados en el funcionamiento de las redes neuronales biológicas del Sistema Nervioso (SN).

El desempeño que hacen las neuronas en el cerebro se explica gracias al movimiento Conexionista cuyo principio central es que los fenómenos mentales pueden ser descritos por redes de unidades sencillas que se interconectan (Ros García, 2019).

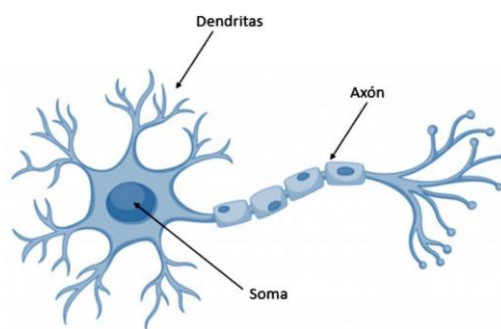


Figura 1.8. Esquema de una neurona biológica. Fuente: (Azpilcueta, 2020).

La neurona (figura 1.8) es la unidad del SN y en especial del cerebro, se encarga de procesar y combinar señales y enviarlas al resto de neuronas conectadas entre sí.

Las RNA siguen el siguiente esquema de funcionamiento (figura 1.9), la unidad principal es la neurona o perceptrón simple. Como se puede observar en la figura cada nodo es una neurona artificial y las flechas representan las conexiones que hay entre estas, desde la salida de una hasta la entrada de la siguiente (Ros García, 2019).

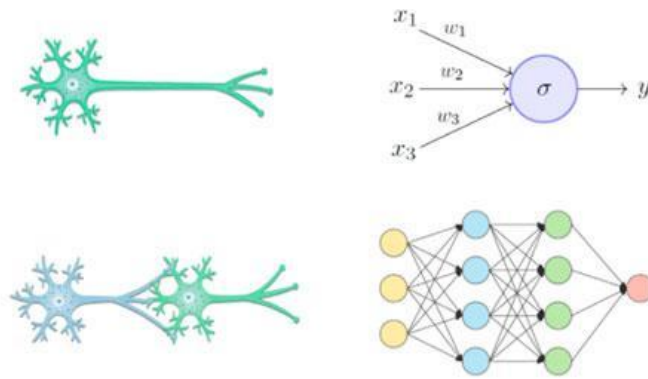


Figura 1.9. Esquema analogía entre neuronas biológicas y redes neuronales. Fuente: (Future Space, 2021)

Al igual que ocurre en el SN en el cual la señal de salida va a depender de las conexiones sinápticas entre neuronas, en las RNA el valor de salida también va a venir determinado por el valor de entrada de las diferentes neuronas conectadas multiplicado por un peso, y de un bias (Figura 1.10). Además, en la salida de cada neurona de la red se le aplica una transformación no lineal conocido como función de activación. Por lo que el modelo general sería el expuesto en la ecuación 1.

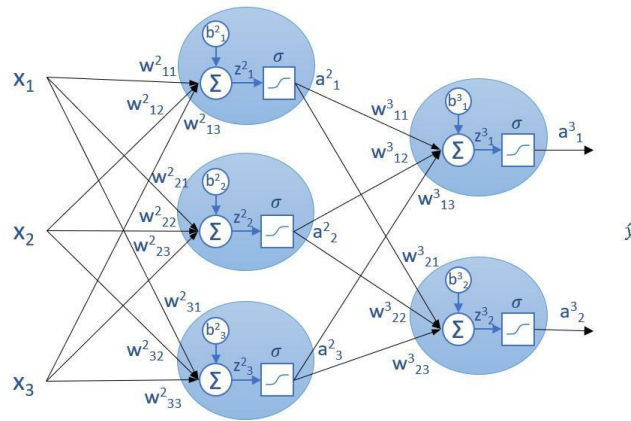


Figura 1.10. Modelo de cinco neuronas conectadas, con los pesos, bias y funciones de activación. Fuente: (InteractiveChaos, 2018)

$$y_j = f\left(\sum_{i=1}^N w_{ij}x_i + b_j\right) \quad (1)$$

En cuanto a funciones de activación, se pueden usar diferentes tipos en base al estudio, dentro de las más conocidas y usadas encontramos las presentes en la tabla 1.1.

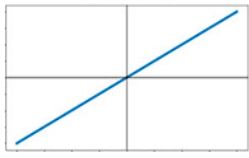
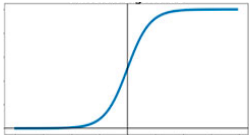
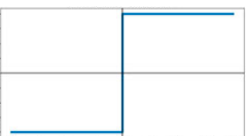
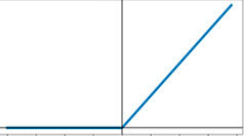
Nombre	Función	Rango	Forma
Lineal	$f(x) = x$	$[-\infty, +\infty]$	
Sigmoide	$f(x) = \frac{1}{1 + e^{-x}}$	$[0, +\infty]$	
Escalón	$f(x) = \text{sign}(x)$ $f(x) = H(x)$	$\{-1,1\}$ $\{0,+1\}$	
ReLU	$f(x) = (0, x)$	$[0, +\infty]$	
Softmax	$f(z)_i = \frac{e^{-z_i}}{\sum_{k=1}^K e^{-z_k}}; z \in R^K; i = 1, \dots, K$	$[0,1]^K$	

Tabla 1.1. Principales funciones de activación.

Las neuronas se suelen organizar formando capas. Dependiendo del número de neuronas que hay en cada capa, del número de capas, de la función de activación y de las conexiones se pueden obtener diferentes tipos de arquitecturas en las RNA. En función del problema a resolver se generará la arquitectura óptima.

A las RNA que están formadas por un conjunto de capas que están unidas por pesos y la salida de una capa es la entrada de la siguiente se les denomina redes multicapa. Estas son capaces de aprender funcionalidades que las redes de una sola capa no pueden (Ros García, 2019). Se puede ver la representación gráfica en la figura 1.11.

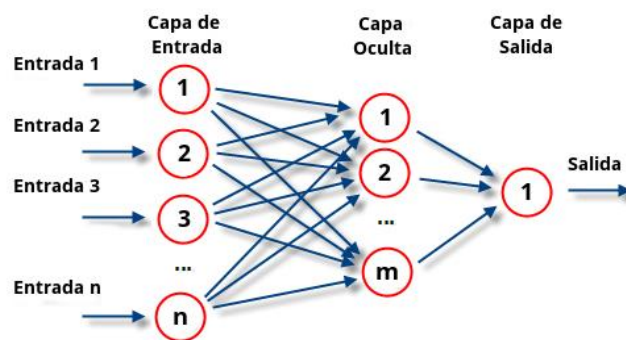


Figura 1.11. Capas de las redes neuronales. Fuente: (Carmona Benítez, 2014)

En esta figura se pueden distinguir tres tipos de capas:

- Capa de entrada, en esta se encuentran las neuronas que reciben los datos iniciales, son las primeras neuronas por lo que no tienen ninguna conexión anterior.
- Capas ocultas, estas son las que hay entre la capa de salida y la de entrada, por lo que van a estar conectadas con las capas adyacentes. Las redes de Deep Learning van a tener múltiples capas ocultas.
- Capas de salida, en estas encontramos las neuronas que nos van a dar el resultado final.

Una de las principales ventajas de las redes neuronales es que aprenden de la experiencia a través de ejemplos de los que extraen características principales. Generalizan en ejemplos nuevos lo que han aprendido de los anteriores (Ros García, 2019).

1.3.3. Entrenamiento y aprendizaje de redes.

Como se ha explicado, en un aprendizaje supervisado se va a producir un entrenamiento con ejemplos. Para que se realice de manera correcta se deben dividir los datos empleados en un conjunto de entrenamiento, un conjunto de validación y un conjunto de test. Los datos de entrenamiento son los que se van a usar para ajustar los pesos y bias, el modelo generado va a estar ajustado a estos datos. Durante este proceso, se usan los datos de validación para tener una estimación del proceso de entrenamiento y evitar el problema de *overfitting* (García, 2020). Para terminar, se hace uso del conjunto de test, en este conjunto se debe asegurar que únicamente haya datos con los que la red no haya sido entrenada, sirven para evaluar la red de forma independiente una vez terminado el proceso de entrenamiento.

En primer lugar, se realiza una inicialización de los parámetros, puede realizarse de manera aleatoria o a partir de los pesos de una red que ha sido entrenada con anterioridad.

Realizada la inicialización, comienza la propagación hacia delante o *feed forward*, en esta etapa las entradas iniciales pasan a través de las capas profundas hasta llegar a la capa de salida. Una vez ha finalizado se realiza la comparación entre los resultados obtenidos y los datos reales que serían el resultado deseado. Para llevar a cabo esta comparación se utilizan funciones de coste o *loss functions* (García, 2020).

De esta manera se obtiene el error cometido por la red y el modelo calculará las modificaciones necesarias para minimizar la función de error. Algunas de estas funciones las podemos ver en la tabla 1.2. Dependiendo de la aplicación se usará un tipo de función de error u otro.

Nombre	Expresión	Aplicación
<i>Mean Absolute Error (MAE)</i> (Norma L1)	$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i $	Regresión
<i>Mean Squared Error (MSE)</i> (Norma L2)	$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i ^2$	Regresión
<i>Binary Cross Entropy (BCE)</i>	$BCE(y, \hat{y}) = (y)(-\log \log \hat{y}) + (1 - y)(-\hat{y})$	Clasificación binaria

Tabla 1.2. Ejemplos de funciones de coste para el entrenamiento de RNA.

Una vez se obtienen los resultados de las funciones de error, comienza la retropropagación del error o *backpropagation*, a partir del error se modifican los valores de las capas de la red con algoritmos de retropropagación basados en el gradiente o la derivada (García, 2020). El algoritmo más conocido es el de *gradient descent*. De esta manera el algoritmo va minimizando la función de coste en cada iteración en dirección opuesta al gradiente.

En la actualidad se hace uso de algoritmos de optimización por lotes o *batches*, debido a que los cálculos de los parámetros para un conjunto de entrenamiento muy amplio pueden ser lentos y de gran coste computacional, una vez determinados los lotes se hacen los cálculos de cada iteración. A cada una de estas iteraciones se les denomina *epochs*, en cada una la red ajusta los pesos del *batch* y cuando termina se hace la predicción sobre el conjunto de validación. Si disminuye el error de la función de coste entonces los pesos calculados son guardados, si por otro lado no disminuye se mantienen los pesos anteriores.

1.3.4. Evaluación

La evaluación comienza cuando se ha finalizado el entrenamiento de la red. Para llevarla a cabo se van a usar los datos que se encuentran en el conjunto de test y que por lo tanto son desconocidos para la red. Lo que sucede es que la red procesa estos datos con los pesos que ha aprendido durante el entrenamiento y obtiene unos resultados que hay que cuantificar para ver si la red realiza su tarea de manera correcta.

Para evaluar este funcionamiento hay una serie de métricas de evaluación en función de la tarea a realizar por la red. A continuación, se presentan algunas de ellas. Para modelos de clasificación se emplean, sensibilidad y especificidad, análisis de la curva ROC o precisión, mientras que si son modelos de regresión se puede usar Mean Squared Error (MSE), Root Mean Squared Error (RMSE) o Mean Absolute Error (MAE).

En nuestro caso de mejora de la resolución, las métricas que se emplean son PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index Measure) o el coeficiente de correlación de Pearson (CC).

1.3.5. Conceptos de DL

Los modelos que se generan de redes neuronales son modelos matemáticos simples que pueden estar asociados con un algoritmo de aprendizaje. Para llegar a comprenderlas de manera correcta se definen a continuación una serie de conceptos utilizados en la composición de los modelos.

- **Overfitting o sobreajuste**, es un problema que se da durante el entrenamiento generado debido a que la red se entrena de manera excesiva sobre el mismo conjunto de datos. Esto hace que la red sea capaz de minimizar el error con los datos de entrenamiento, pero este se ve aumentado en los datos no entrenados o de validación. Debido a esto la red pierde capacidad a la hora de predecir datos nuevos.
- **Underfitting**, es el problema contrario al *overfitting*, en este caso la red no es entrenada lo suficiente con el dataset de entrenamiento, de esta manera no aprende por lo que no podrá hacer buenas predicciones, esto puede deberse a que no hay suficientes datos sobre los que entrenarse o por un número insuficiente de epochs.

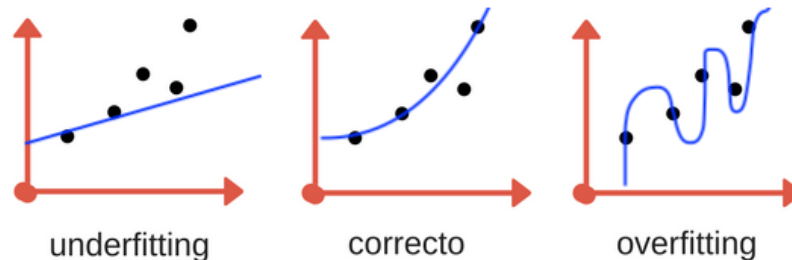


Figura 1.12. Ejemplo de *overfitting*, *underfitting* y resultado correcto. Fuente: (Rivera Phillips, 2019).

- **Batch normalization**, en primer lugar, un *batch* es el número de ejemplos que se introducen en una red para que entrene cada vez. Esta capa se encarga de normalizar los valores de las neuronas de entrada calculando primero la media y la varianza y tras esto normalizando los datos restando la media y dividiendo por la desviación. Esta capa acelera de forma notable el proceso de aprendizaje.
- **Fine-tuning**, consiste en bloquear los parámetros de una red que ya ha sido entrenada, generalmente se bloquean los parámetros de las primeras capas, de esta forma solo algunas capas de la red se pueden modificar. Esto sirve para adaptar una red que ya ha sido entrenada para una tarea similar, sin necesidad de entrenarla desde el principio. También se puede usar sin desbloquear los pesos para ser usado como una inicialización alternativa a la aleatoria.

1.4. ESTADO DEL ARTE

En este apartado se va a realizar una recopilación de las herramientas o software que están disponibles y se utilizan en la actualidad para llevar a cabo las diferentes técnicas que se han explicado con anterioridad sobre imagen médica.

1.4.1. Herramientas de registro.

A continuación, se exponen algunos de los distintos softwares que disponen los investigadores y médicos para realizar el registro de imágenes en la actualidad.

- ITK (Insight Toolkit), es una herramienta de código abierto usada para el registro y segmentación de imágenes médicas. Permite realizar transformaciones de manera manual o automática, las transformaciones ofrecidas son de tipo afín, rígida y no rígida.
- ANTs (Advanced Normalization Tools), es un software de código abierto considerado el estado del arte en el registro de imágenes médicas y está basado en ITK. (Avants et al., 2008)

1.4.2. Técnicas de superresolución.

A continuación, se procede a hacer una revisión de tecnologías de superresolución que están siendo usadas en la actualidad, centrado en la superresolución sobre una sola imagen o SISR (Single Image Superresolution). En este caso se supone que la imagen de LR es una imagen submuestreada y con ruido de la imagen de alta resolución.

Hay gran cantidad de métodos propuestos para realizar superresolución sobre una imagen, que no hay que confundir con los métodos basados en interpolación, estos han sido ampliamente utilizados a lo largo de los años en la investigación debido a su bajo costo y baja complejidad computacional. Estos métodos tienden a difuminar detalles de alta frecuencia y pueden producir escalas muy grandes (Hernández, 2020).

Un enfoque clásico, es el uso de algoritmos basados en ejemplos, en estos se generan imágenes de HR a partir de imágenes de LR haciendo uso de la auto similitud entre estas (Hernández, 2020). Para poder realizar este método se requiere de una base de datos o diccionarios en los cuales estén presentes una serie de pares de parches, es decir porciones de la imagen tanto de alta y baja resolución. El objetivo de estos es obtener relaciones de mapeo que permitan estimar las nuevas imágenes de HR a partir de las de LR. Generalmente para establecer estas relaciones de mapeo se usan métodos basados en vecindades, en métodos que usan la escasez o basados en regresión. La desventaja de estos métodos es que es necesario realizar una búsqueda de un amplio conjunto de datos (Hernández, 2020).

En la actualidad, los modelos que más éxito tienen son los basados en redes neuronales convolucionales, será en este enfoque en el que se centrará el presente trabajo. Estos, a partir de un mapeo no lineal entre parches de LR y HR reconstruyen la imagen HR que se está buscando. En estos casos, las redes también tienen que ser entrenadas con un amplio volumen de datos. La ventaja que supone usar este método es que proporciona un rendimiento superior en calidad de restauración permitiendo así una mejor relación señal ruido (PSNR, Peak-Signal to Noise) y en similitud estructural (SSIM, Structural Similarity Index Measure), métricas utilizadas para ver la calidad del método de superresolución (Hernández, 2020). En la actualidad, existen multitud de algoritmos basados en redes convolucionales, como el llamado SynthSR (Iglesias et al., 2021) que realiza la superresolución de imágenes y que será explicado en el apartado 1.4.4.

1.4.3. Redes neuronales profundas.

En este apartado se comentarán las principales arquitecturas de redes neuronales usadas para hacer superresolución en imagen médica y su funcionamiento.

- **Redes Neuronales Convolucionales o *Convolutional Neural Networks (CNN)***: son un tipo de red neuronal artificial que va a procesar sus capas imitando al córtex visual del cerebro para intentar identificar distintas características de la entrada (Artola Moreno, 2019). Generalmente estas se usan para hacer la extracción de características de imagen.

Estas redes reciben su nombre por hacer uso de capas de convolución en su arquitectura, esta operación de convolución matemáticamente va a venir definida de la siguiente manera:

$$Y[i, j, k] = \sum_{k=-r}^r \sum_{k=-r}^r \sum_{k=-r}^r K[u, v, w]X[i - u, j - v, k - w] \quad (2)$$

$$Y = K \otimes X \quad (3)$$

En las ecuaciones 2 y 3, Y es el resultado obtenido de realizar la operación de convolución entre X que es la imagen y K el cual es un filtro o kernel. Estos Kernels, van a ir ajustando sus valores de manera que obtenga el mayor peso las características de mayor importancia volviéndose de esta manera dominantes sobre el resto. El valor de estos kernels se va a ir actualizando de manera periódica en cada iteración. Para poder ver gráficamente como es la convolución tenemos la siguiente figura 1.13.

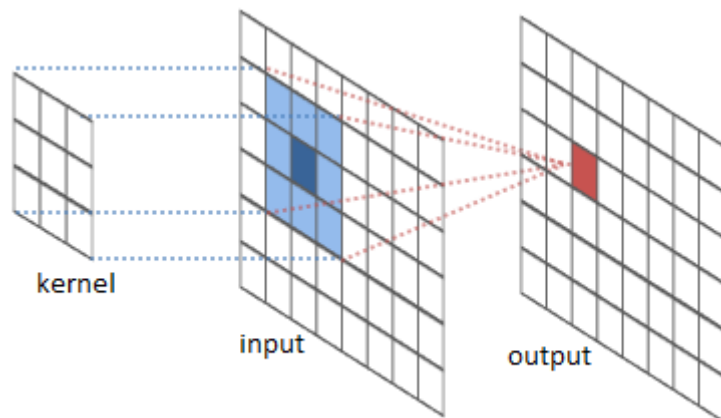


Figura 1.13. Esquema de filtrado de imagen en una capa de convolución Fuente: (Silvestre Llopis, 2019).

Para obtener una buena convolución siempre se debe tener en cuenta la dimensionalidad de los Kernels, hay una relación directa entre el tamaño de kernel y el nivel de detección de características relevantes (Soto, 2021). Con un tamaño de kernel grande, la detección de características será pobre mientras que si es al contrario, se

encontrarán más características. Debemos de saber que el tamaño de kernel usado también va a suponer un mayor o menor coste computacional, por lo que se debe encontrar un equilibrio. Generalmente los filtros usados son de 3x3 o de 5x5 en imágenes 2D (Soto, 2021).

La diferencia entre la forma general explicada anteriormente de ANN y CNN es que los pesos de la red de la última arquitectura van a ser los valores de los Kernels. Por lo que podemos particularizar la expresión de la ANN (ecuación 1) a CNN como:

$$Y_j = f(b_j + \sum_i^N K_{ij} \otimes X_i) \quad (4)$$

Siendo Y la matriz de salida, X cada una de las N neuronas o matrices de entrada, K el Kernel asociado a la conexión entre las neuronas, b es el bias y f la función de activación.

En este tipo de arquitectura, encontramos que una vez finalizado todo el proceso de convolución, se va a obtener una matriz de convolución llamada mapa de características en la cual se ven representadas las diferentes características de la imagen. Encontramos también las capas de reducción o *pooling*, se van a encargar de minimizar el tamaño de la matriz, por lo que las capas más profundas encontraremos toda la información importante resumida.

Por último, a la salida habrá neuronas de clasificación a las cuales llegan las características relevantes que se extraen y se encargarán de realizar una clasificación de estas características.

- **Autoencoder:** Los autoencoders son un tipo de red convolucional que se suelen utilizar para condensar la información relevante de una imagen. Consta de un encoder y un decoder. El encoder reduce la dimensión de las características mediante la aplicación sucesiva de capas de convolución y pooling. El decoder realizar el proceso inverso. Este tipo de redes tiene como entrada y salida la misma imagen y se suele usar como método de reducción de dimensionalidad usando la salida del encoder como una extracción de características de menor dimensión que la imagen de entrada. La figura 1.14 muestra un ejemplo de esta arquitectura.

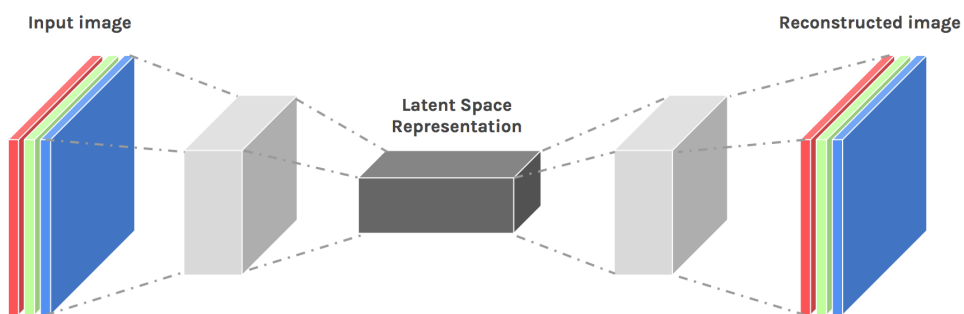


Figura 1.14. Arquitectura Autoencoder. Fuente: (Silvestre Llopis, 2019)

- **UNET:** Este tipo de arquitectura desarrolladas por Olaf Ronneberger, Philipp Fischer y Thomas Brox (Ronneberger et al., 2015) es una generalización del autoencoder donde se usan conexiones residuales en cada nivel de resolución con el fin de minimizar la pérdida de resolución de las predicciones como se puede observar en la Figura 1.15. La UNET está formada por:

1. Codificador (*Encoder*), esta es la parte que se encarga de comprimir los datos de entrada es decir se codifica la información desde un espacio original a un espacio reducido con reducción de dimensionalidad o *downsampling*.
2. *Bottleneck feature*, es la capa que divide el *encoder* y *decoder*. En esta encontramos la representación comprimida de la entrada, con dimensiones menores, pero con la información más relevante.
3. Decodificador (*Decoder*), esta parte está a cargo de reconstruir los datos objetivo a partir de la representación comprimida. Es decir, se encarga de llevar la información del espacio latente al original, es la inversa del *decoder*, en el proceso encontramos tanto capas de convolución como capas de *upsampling*, este aumento de la dimensionalidad se puede realizar con *Zero Padding* o con interpolación.
4. Capas de conexión: Para terminar con esta arquitectura se debe añadir que, para no sufrir una pérdida de información durante los procesos de codificación y decodificación, se pueden incluir conexiones entre los tramos. Es decir, las matrices convolucionadas que hay en el tramo del codificador se concatenan con el conjunto remuestreado después de hacer *upsampling* (Vilarroig, 2019).

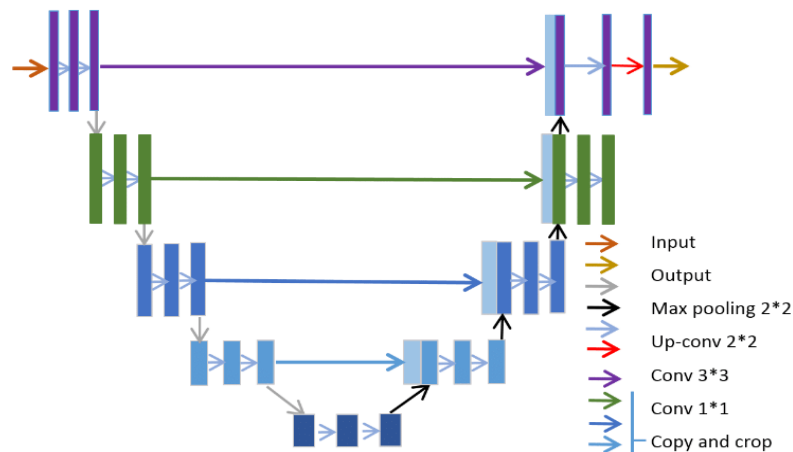


Figura 1.15. Modelo U-Net. Fuente: (Jayawardene, 2020/2021).

1.4.4. SynthSR

En este apartado se presenta el método de SynthSR, este es un algoritmo que se basa en el uso de redes neuronales convolucionales para realizar superresolución en imagen médica (Iglesias et al., 2021). Este es el algoritmo más cercano que está relacionado con el presente trabajo.

Este método puede usarse sobre las diferentes potencias de resonancia y devolverá una imagen de alta resolución potenciada en T1.

En el artículo del método, se defiende una solución al problema de las diferencias sustanciales que hay en los protocolos de adquisición de RM tanto entre centros, como también de orientación, resolución y contraste. De manera que la solución se basa en utilizar imágenes generadas sintéticamente para realizar el entrenamiento de la red Convolutiva.

El control de este proceso generativo de las imágenes permite realizar el entrenamiento de la CNN tanto para superresolución como para síntesis o ambas a la vez para cualquier combinación de contraste, resolución u orientación, esto permite una gran aplicabilidad.

En primer lugar, para poder entrenar este método se realizó un generador de datos sintéticos con el objetivo de generar esos datos de cualquier tipo de resolución y contraste. Para poder hacerlo, contaban con un dataset de imágenes de HR de cerebro con el mismo contraste con sus correspondientes segmentaciones y clases relacionadas con las estructuras del cerebro de las cuales tener una referencia para después entrenar la red para la síntesis. Tras esto se realiza un “data Augmentation” que se realiza aplicando transformaciones espaciales.

Una vez obtenidos todos los datos de manera correcta se generan las intensidades de las HR sintéticas haciendo uso de muestreo de un modelo de mezcla gaussiana o *Gaussian Mixture Model* (GMM). Por último, se generan las intensidades de las imágenes de LR sintéticas y corrompidas haciendo uso de la variabilidad de los marcos de coordenadas, teniendo en cuenta la resolución con espaciado y grosor de cortes (volumen parcial) y generando errores de registro y artefactos de remuestreo. A partir de este momento se comienza a realizar el entrenamiento de la red ya que se tienen todos los datos adecuados.

Para realizar el aprendizaje de la red usaron estos datos sintéticos y en el caso de hacer únicamente superresolución se busca recuperar el volumen de HR sintético, para hacerlo la red resuelve un problema de optimización sencillo en el que predice el residuo en lugar del volumen directamente. La CNN utilizada es entrenada con un optimizador llamado Adam (Kingma & Ba, 2017) y busca minimizar la norma L1 del error. Decidieron hacer uso de la norma L1 ya que esta generaba resultados visuales más realistas comparados con el uso de la norma L2.

Por último, la arquitectura de la CNN usada en este método es una U-net 3D con 5 niveles. Cada uno de los niveles consiste en dos capas cada una comprende convoluciones de con Kernels de 3x3x3 y una función de activación no lineal denominada ELU. La primera capa tiene 24 Kernels, este número se va a ver duplicado en cada *max-pooling* y reducido a la mitad en cada upsampling. La última capa usa una activación lineal para la estimación.

Para validar los resultados y la aplicabilidad del método hacen uso de métricas de similitud de imagen (PSNR y SSIM) y también haciendo un análisis del rendimiento de herramientas comunes de neuroimagen. La figura 1.16 muestra varios ejemplos de la aplicación de este algoritmo.

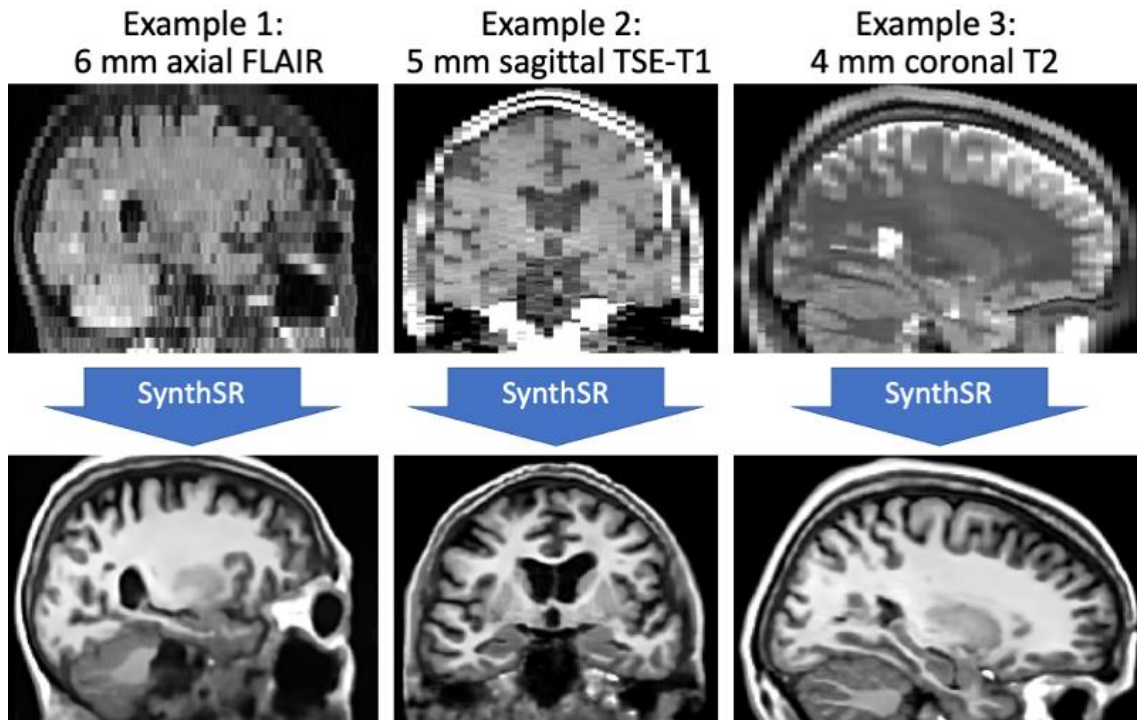


Figura 1.16. Ejemplos de la aplicación del algoritmo SynthSR. Fuente: (Iglesias et al., 2021)

1.4.5. SynthSEG

A continuación, se presenta SynthSeg, la primera herramienta de DL para segmentación de imágenes de cerebro de cualquier contraste y resolución (Billot et al., 2023). Como sabemos la segmentación es una técnica importante en el tratamiento de imágenes debido a que permite análisis volumétricos y de forma que pueden ser usados para generar diagnósticos mejores y con antelación.

En el artículo se presenta esta herramienta como la primera red neuronal capaz de segmentar imágenes cerebrales de cualquier resolución y contraste sin tener que reentrenarla ni hacer *fine tuning* (Billot et al., 2023). Solucionando de esta manera los tiempos de espera de entrenamiento de la red.

En primer lugar, para entrenar la red se crean imágenes sintéticas como las muestras de entrenamiento a partir de un modelo generativo, con parámetros aleatorios para que asegure que se generan imágenes con diferente resolución y contraste. Una vez se tienen las imágenes se generan los mapas de etiquetas los cuales se han obtenido tanto manualmente como automáticamente. Una vez hecho esto se genera una imagen sintética inicial en HR muestreando en GMM.

Para conseguir que el método sea más robusto se genera tanto *spatial augmentation* como *intensity augmentation* y un campo de inhomogeneidad. Por otro lado, para conseguir la robustez para las diferentes resoluciones modelan el espaciado y el grosor entre los diferentes cortes.

La arquitectura de red que se usa para la generación del método es una U-Net 3D que consta de cinco niveles separados cada uno por una capa de *batch normalization* con *max pooling* o *upsampling*. Cada nivel tiene dos capas de convolución con kernel de 3x3x3. Se hace uso de la función de activación ELU (Clevert et al., 2016), la primera capa tiene 24 Kernels, este número

se va a ver duplicado en cada *max-pooling* y reducido a la mitad en cada *upsampling*. La última capa usa una activación lineal para la estimación (softmax).

En el momento de la evaluación, la imagen que se pasa de entrada se va a remuestrear con el objetivo de que la salida de la red sea una imagen de HR con intensidades normalizadas, por otro lado, se obtienen mapas de predicciones suaves para cada etiqueta (Billot et al., 2023). Como salida se obtienen las imágenes en HR anteriormente dichas y sus segmentaciones.

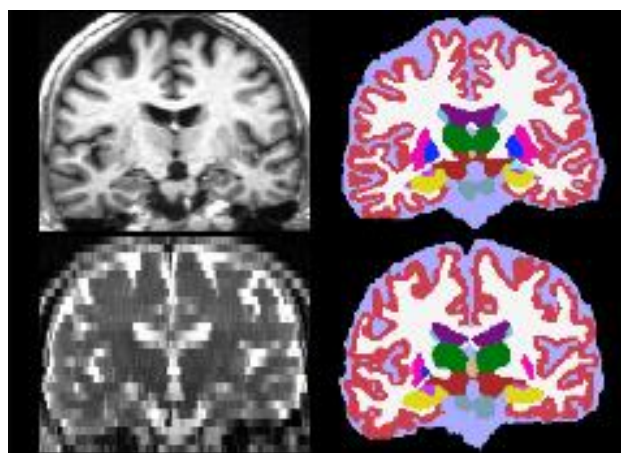


Figura 1.17. Ejemplo de uso de SynthSeg, a la izquierda se encuentra la imagen de entrada y a la derecha su segmentación. Fuente: (Billot et al., 2023)

2. OBJETIVOS E HIPOTESIS

Objetivo principal

El objetivo principal de este presente Trabajo Final de Grado es el diseño y desarrollo de un algoritmo basado en redes neuronales convolucionales para mejorar la calidad de imágenes de RM baja resolución.

Hipótesis

Tras analizar los métodos del estado del arte se plantea una posible mejora de los mismos en base a las siguientes hipótesis:

1. Resolver solo el problema de la mejora resolución dejando el problema de síntesis de modalidad aparte.
2. Dividir el problema en sub-problemas para simplificar cada uno de ellos y mejorar la efectividad.
3. Evaluar nuevas funciones de coste asociadas al objetivo final de análisis específico.

Objetivos secundarios

Para evaluar la validez de las hipótesis planteadas se plantan una serie de objetivos secundarios:

- Familiarización con la herramienta actual de superresolución de imágenes SynthSR.
- Familiarización con la herramienta actual de segmentación para múltiples resoluciones SynthSEG.
- Estudio de las diferentes formas de generar una red neuronal, técnicas y arquitecturas usadas y estudio de los parámetros a tener en cuenta y su efecto en el aprendizaje de la red.
- Entrenar y evaluar distintas versiones redes neuronales. Con especial énfasis en la comparación de la solución basada en una sola red o el uso de múltiples redes para múltiples resoluciones.
- Evaluar una nueva función de coste asociada a la tarea de segmentación para reconstruir las imágenes con mayor calidad.
- Identificación de posibles problemas y limitaciones observados durante el desarrollo del trabajo y propuesta de mejoras que permitan la solución de estos.

3. MATERIALES Y MÉTODOS

Se comenzará describiendo cuáles han sido los materiales utilizados para poder llevar a cabo la investigación de manera correcta y los métodos que se han seguido para obtener los resultados más óptimos posibles.

3.1. MATERIALES

Para realizar el siguiente proyecto se han usado diferentes herramientas tanto software como hardware y datos extraídos de una base de datos determinada que serán explicados en los siguientes apartados.

3.1.1. Imágenes

Las imágenes utilizadas provienen de la base de datos *Human Connectome Project* (<https://www.humanconnectome.org/>). El *Human Connectome Project* (HCP) es un proyecto científico a gran escala cuyo objetivo es mapear la conectividad del cerebro humano utilizando técnicas de neuroimagen avanzadas. El proyecto, lanzado en 2009, ha recopilado imágenes de resonancia magnética de más de 1,200 voluntarios. En este TFG se han empleado 75 imágenes potenciadas en T1 con unas dimensiones de 260x311x260 vóxeles elegidos arbitrariamente.

A partir de estas 75 imágenes se hace la división entre los datos de test y los de training. Para el test se reservó 5 mientras que para el training 70 imágenes. A cada una de estas se le aplicaron diferentes transformaciones como se ha explicado con anterioridad para poder obtener las imágenes de baja resolución a partir de las cuales se hace el entrenamiento, validación y evaluación.

Por otro lado, se va a usar una segunda base de datos de 3000 archivos que son las imágenes de alta resolución potenciadas en T1 denominada 'QC3000'. De este se hará de nuevo una separación de los datos, en la carpeta de *test* ubicamos 315 imágenes mientras que en *training* tenemos las restantes. A partir de estas se generarán las imágenes de baja resolución de este dataset.

Por último, para generar las imágenes de baja resolución se usaron 45.000 imágenes de la base de datos volBrain. VolBrain es un sistema online de análisis de neuroimagen (www.volBrain.net) que cuenta con una base de datos privada procedente de los usuarios del sistema que han decidido compartir sus imágenes con los creadores del sistema. Esta base de datos consta de cerca de 400.000 imágenes de todo el planeta y representa una de las más diversas en procedencia y variabilidad. Estas imágenes, se usaron para generar datos de baja resolución representativas de la realidad actual.

3.1.2. Hardware

Para llevar a cabo el trabajo se ha usado un equipo PC de sobremesa para la realización de los scripts en los distintos lenguajes de programación y realizar los experimentos.

Especificaciones	Características
Sistema operativo	AMD Ryzen 7 1700 Eight-Core Processor de 3.00 GHz
Procesador	Windows 10 Enterprise
Memoria RAM	16,0 GB
GPU	Nvidia GTX 1080 Ti (11 GB)

Tabla 3.1. Características del equipo usado para los experimentos.

Para realizar los entrenamientos y los test de las redes generadas se ha usado una GPU con 11 GB de memoria. Es importante conocer las características de la máquina usada para realizar el proyecto ya que la velocidad, el rendimiento y los tiempos de ejecución van a depender de estas.

3.1.3. Software

El desarrollo del código del presente proyecto se ha llevado a cabo con dos lenguajes de programación, Python y MATLAB. Por un lado, Python es un lenguaje de programación de alto nivel creado por Guido Rossum, este destaca por tener una sintaxis clara y legible, por su versatilidad, portabilidad, interactividad y la existencia de bibliotecas que facilitan el desarrollo de aplicaciones. Por otro lado, MATLAB es un entorno de programación ampliamente usado en el procesamiento y análisis de datos creado por MathWorks y destaca por su capacidad de hacer cálculos numéricos y manipulación de matrices eficiente. Al igual que Python, es un entorno interactivo, con una amplia biblioteca de funciones, permite la visualización de datos y se puede integrar con otros lenguajes y herramientas.

Para realizar la visualización de las imágenes se ha empleado la aplicación de ITK-SNAP (Yushkevich & Gerig, 2017), la cual es una aplicación gratuita. Esta permite la visualización y segmentación automática de imágenes médicas.

Cada uno de estos lenguajes ha sido usado para diferentes tareas, en primer lugar, con MATLAB se han preprocesado los datos, se ha llevado a cabo el registro de las imágenes desde un espacio nativo al espacio MNI152, se han generado las diferentes imágenes de LR necesarias a partir de las imágenes HR originales y se han clasificado estas con sus diferentes resoluciones.

Por otro lado, Python se ha usado para desarrollar diferentes scripts entre los que encontramos los modelos de las diferentes arquitecturas de las redes neuronales con las que se han llevado a cabo experimentos, un script de lectura de datos en el cual se leen los datos de validación y de entrenamiento y se hace *“data augmentation”* y por último un script de test en el que se usan los datos del conjunto de test y sobre los que se hace una evaluación con diferentes métricas.

Para poder llevar a cabo lo mencionado, se han empleado una serie de librerías de python: TensorFlow y Keras, Numpy, Matplotlib, Nibabel, Os, Glob, Scipy entre otros. Todos los scripts se han ejecutado en la plataforma de Anaconda.

3.2. METODOLOGÍA

Para entender este TFG hay que tener en cuenta que las imágenes de RM se obtienen en múltiples orientaciones y resoluciones por todo el mundo. Por lo tanto, para mejorar su resolución es mucho más eficiente trabajar en un espacio estandarizado (como el MNI152). El sistema propuesto en este TFG, trabaja en dicho espacio, y tiene como objetivo transformar imágenes de baja resolución (LR) y su versión correspondiente de alta resolución (HR). La figura 3.1. muestra un ejemplo del input y output del sistema propuesto.

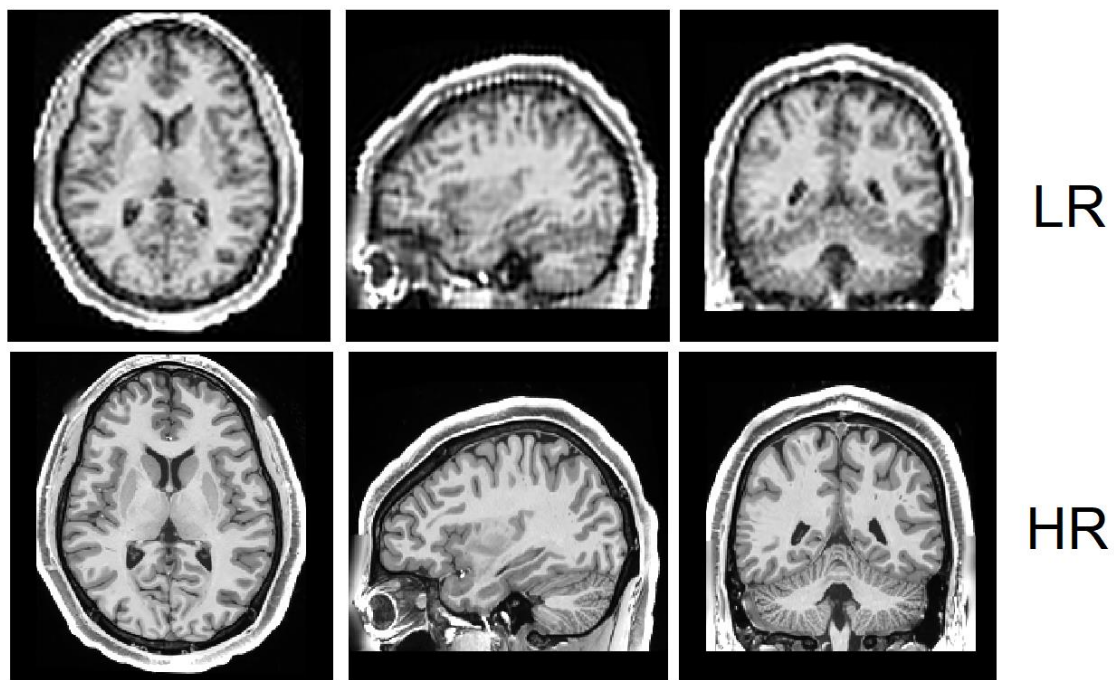


Figura 3.1. La fila superior muestra una imagen simulada de baja resolución donde se pueden observar los artefactos de interpolación. La fila inferior muestra la imagen de alta resolución correspondiente (objetivo).

Para llevar a cabo el objetivo propuesto, se ha seguido el flujo de trabajo que vemos en la figura 3.2. En primer lugar, se va a realizar una preparación y procesado de los datos que se van a introducir en la red para que esta sea entrenada y después validada. Una vez se tienen los datos se genera el modelo o la arquitectura de la red que va a resolver el problema y para aumentar el número de casos se va a usar un generador de datos o *data augmentation*. Una vez preparados los datos comienza el entrenamiento de la red generada y la evaluación de la misma.

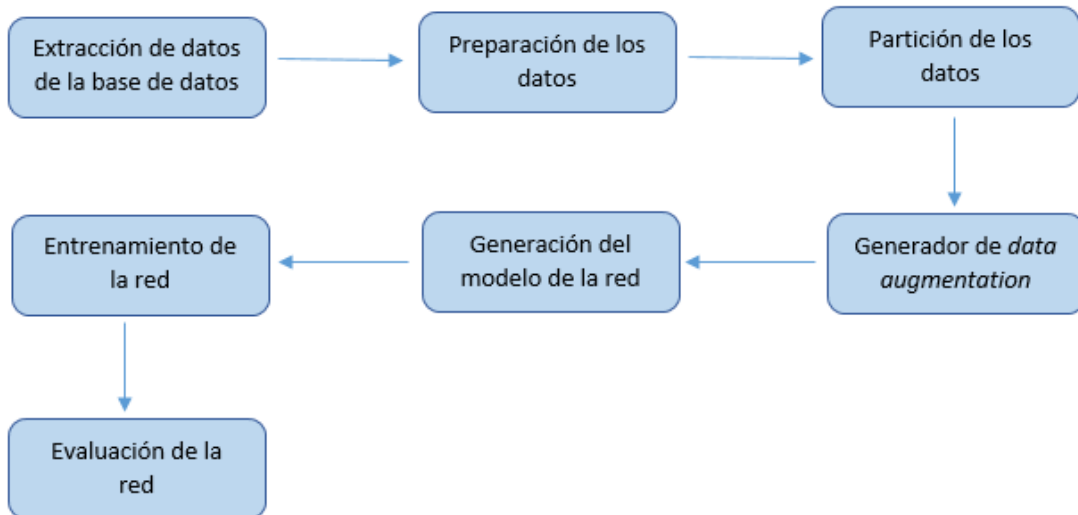


Figura 3.2. Flujo de trabajo.

3.2.1. Preparación de los datos.

Para preparar los datos, todas las imágenes se registraron al espacio MNI152. Esto se realizó usando el software ANTS y una imagen de referencia en dicho espacio (atlas T1 mni152). Este paso, genera imágenes en el espacio MNI de tamaño 181x217x181 vóxeles y un tamaño de voxel de 1 mm³ mediante la aplicación de una transformación afín de 12 grados de libertad.

Una vez se tienen las imágenes de HR de referencia registradas al espacio MNI se va a realizar de nuevo el uso del software ANTs para generar imágenes LR a partir de dichas imágenes. Para simular las imágenes de LR se usaron las imágenes de la base de datos de volBrain. Esta base de datos, contiene imágenes de las más diversas resoluciones (2x2x2 mm, 1x1x5, 1.2x1.2x2, etc) y orientaciones (coronal, axial, sagital, oblicuo).

El proceso para simular imágenes de LR consistió en registrar las imágenes HR al espacio nativo de las de imágenes de volBrain (con la consiguiente pérdida de resolución) y después aplicar la transformación inversa para volver al espacio MNI. En este proceso, la imagen resultante pierde calidad y además simula artefactos realistas ya que se usaron imágenes reales de referencia.

Dada la diversidad de resoluciones, se agruparon las imágenes generadas por resolución (calculada como $res=resx*resy*resz$, por ejemplo, $1.2x1.2x2=2.88$ mm³). Se generaron 7 grupos (<1, 2, 3, 4, 5, 6 y >7 mm).

Se realizó una selección aleatoria de la imagen de referencia LR y la transformación que se va a aplicar y la imagen de HR a la cual se le va a realizar el proceso. Como salida se generan las rutas y los nombres de archivo para guardar las imágenes que han sido transformadas.

Tras realizar este procesamiento automatizado de las imágenes con ANTs se realiza una evaluación de la calidad de las transformaciones haciendo uso de la métrica de coeficiente de correlación. Esto se realiza para asegurarnos que las imágenes han sido generadas de manera correcta ya que en algunas de las imágenes que se obtenían se podía observar cortes debido a

una mala transformación. Esto es necesario, ya que, si se hace uso de estas imágenes mal generadas durante el entrenamiento de la red, esta, no aprenderá de forma correcta y no se obtendrán resultados robustos. Para cada una de las resoluciones se estableció un umbral de correlación para evitar este problema si el coeficiente está por debajo del umbral entonces el archivo transformado es eliminado, de lo contrario la imagen generada se guarda en el directorio especificado.

Una vez que ya se cuenta con los datos tanto de HR como LR hay que hacer un procesado de los mismos ya que como se sabe el preprocesado de las imágenes antes de usarlas para obtener resultados es de vital importancia. Para realizar esta preparación de los datos se ha usado un script de Python en el cual se aplican algunas de estas técnicas. A continuación, se exponen los métodos usados:

1. En primer lugar, se va a realizar un tratamiento de los datos con el objetivo de manipular los valores NaN, es decir, aquellos valores que son *Not a Number* y que pueden generar problemas a la hora de realizar el entrenamiento. Para llevarlo a cabo se busca tanto en imágenes de LR como de HR cualquier valor NaN y si se encuentra se reemplaza con un cero. Se usa para evitar píxeles o elementos que no tienen un valor numérico válido.
2. Normalización de las imágenes LR y HR. Esta normalización consistió en dividir cada imagen por su media. Al realizarlo, aseguramos que las imágenes estén a una escala de intensidad similar, es decir, se elimina el sesgo de intensidad. Esto supone un entrenamiento más sencillo debido a que se evitan diferencias excesivas en la escala de intensidades de las imágenes.

Tras todo el proceso de generación de datos de entrenamiento el total de pares de datos LR-HR es de 20.289 casos (unos 3000 casos por resolución, desde 1 mm³ a 7 mm³).

3.2.2. Modelo de la red

El modelo que se ha usado para llevar a cabo el trabajo ha sido una arquitectura de CNN específicamente una versión de U-Net 3D (Figura 3.3) con algunas modificaciones. Se ha generado en base a los mejores resultados obtenidos que se pueden ver en el apartado 4 del presente TFG.

En la capa inicial, tenemos la capa de entrada de la imagen, esta es el input de la red y son las imágenes de LR de los sujetos, estas cuentan con unas dimensiones de anchura, profundidad y longitud determinadas (181 x 217 x 181) debido a que son un volumen. Como ya sabemos, a lo largo de la red se van a ir generando cada vez capas más profundas que van a ir extrayendo los mapas de características más importantes que van a identificar los patrones de las imágenes, el nivel de detalle de esto va a venir relacionado con la profundidad de la red. Tras la primera convolución la red ya no trabaja con la imagen inicial sino con los mapas de características que se han obtenido de esta y así sucesivamente.

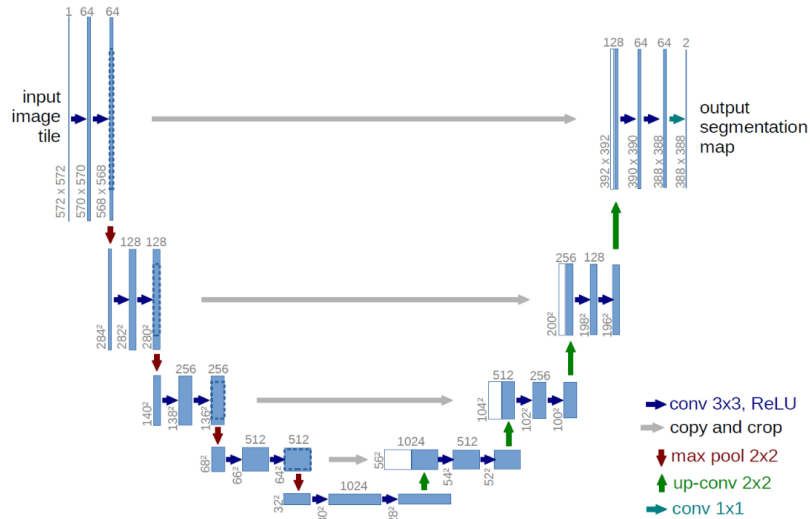


Figura 3.3. Imagen de una red U-Net. Fuente: (López-Lozano, 2021)

Una vez introducidas las imágenes comienzan las capas del codificador (*encoder*), este se va a componer de cuatro bloques convolucionales cada uno de ellos seguidos de una capa de *pooling*. En cada uno de los bloques se le aplica una convolución 3D con un kernel de tamaño 3x3x3 y una función de activación ReLU. Tras la capa convolucional se agrega una capa de normalización (*Batch Normalization*). A parte, en las capas 2, 3 y 4 se le aplica una capa de *pooling* para reducir la dimensionalidad espacial de la imagen de entrada. Tras la primera capa de *pooling* se van a duplicar el número de filtros que se van a ir aplicando, de esta manera los mapas de características comienzan a aumentar hasta alcanzar la última capa donde el número de filtros se multiplica por 8.

Tras el codificador (*encoder*) y la capa latente (*bottleneck feature*) comienzan las capas de decodificación (*decoder*), esta se compone por bloques de *upsampling* seguidos de convoluciones 3D. Cada bloque de *upsampling* se va a usar para aumentar de nuevo las dimensiones espaciales de la entrada con el objetivo de recuperar las dimensiones iniciales.

Tras esto se aplican concatenaciones entre las capas de *downsampling* y las de *upsampling*, después de cada concatenación se aplica una capa de nuevo de *Batch Normalization* y una convolución 3D con una función de activación ReLU.

Por último, la salida del decodificador pasa por una última capa de convolución en la que se va a obtener una única salida final de un solo canal. Esta salida se suma a la imagen de entrada para reconstruir la imagen final. Es decir, se ha usado una variante residual de la UNET3D donde la red se encarga de calcular las altas frecuencias que hay que sumar a la imagen original para obtener la imagen de HR.

También se usó una red de segmentación para evaluar la calidad de las imágenes reconstruidas. En concreto, se usó una red DPN (Deep Piramidal Network) (ver figura 3.4).

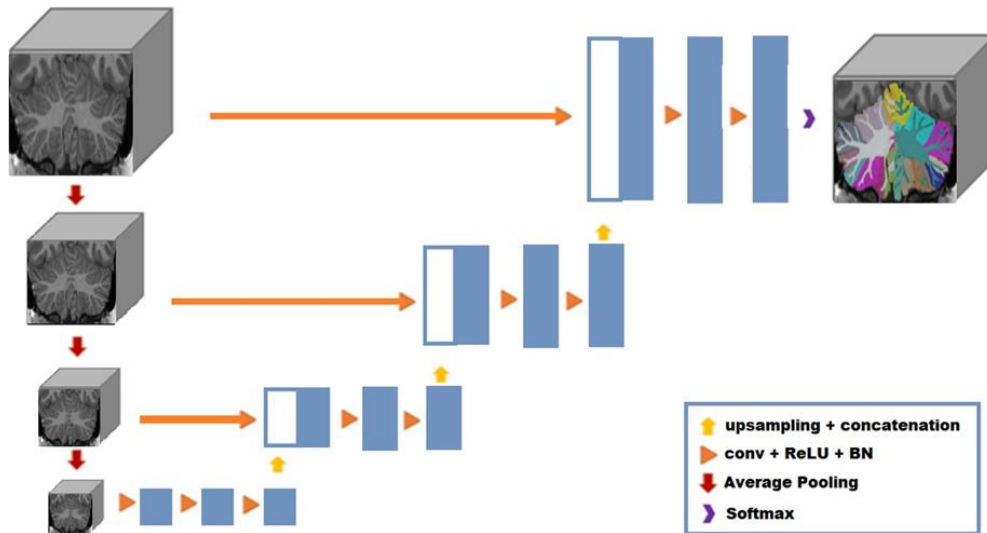


Figura 3.4. Red DPN aplicada a la segmentación del cerebro. Fuente: (Morell S., 2023)

3.2.3. Data augmentation

El “data augmentation” sirve para aumentar la cantidad y la diversidad de los datos de entrenamiento disponibles como se puede ver en la Figura 3.5. De esta manera al introducir estas variaciones en los datos se aumenta la capacidad de generalización del modelo y mejora el rendimiento, aumentando de esta manera la robustez.

Se realizan una serie de transformaciones aleatorias sobre las imágenes de entrada generadas. En nuestro caso, se aplica una o no (50% de probabilidad), una transformación en espejo de forma que el hemisferio izquierdo se pone a la derecha y viceversa aprovechando la simetría natural del cerebro. También, se genera un número aleatorio en el rango de [0.9, 1.1], que se multiplica tanto por la imagen de LR como la de HR para conseguir una modificación aleatoria en la escala del brillo aumentando la variabilidad en la intensidad de la imagen. También, se introduce de forma aleatorio un emborronamiento usando un filtro uniforme.

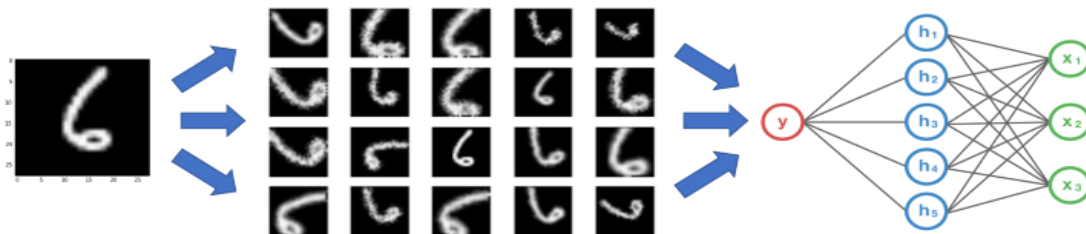


Figura 3.5. Ejemplo del funcionamiento de data augmentation. Fuente: (Raj, 2018)

3.3. ENTRENAMIENTO DE LA RED

El enfoque general seguido para la mejora de la resolución de la imagen es la creación de 7 redes neuronales U-Nets como se ha expuesto anteriormente. Una para cada una de las 7 resoluciones de imágenes usadas.

3.3.1. Training

La metodología seguida para hacer el training ha sido hacer pruebas con diferentes arquitecturas de redes neuronales buscando cuál es la que obtiene los mejores resultados los cuales veremos en el siguiente apartado. Todas las redes propuestas han sido redes neuronales convolucionales las cuales presentan algunos cambios entre sí con el fin de mejorar los resultados finales.

Se ha usado un enfoque secuencial para entrenar las 7 redes. Es decir, primero se entrena la red de resolución de 1 mm^3 usando el dataset correspondiente. Cuando acaba, se utilizan los pesos de esta red para inicializar los pesos de la red de 2 mm^3 usando un enfoque de *fine tuning*. Este proceso se itera hasta la resolución de 7 mm^3 .

Se han usado distintas funciones de pérdida. La tasa de error obtenida se usa para optimizar el modelo en cada iteración. Se recuerda que el conjunto de validación y el conjunto de test se deben diferenciar de manera correcta, pues el primero nos evita el *overfitting* y el segundo se usa una vez el entrenamiento ha terminado para evaluar lo aprendido. El optimizador que ha sido elegido para realizar este TFG ha sido uno de los que más están siendo usados en la actualidad, es decir, el optimizador Adamax (Kingma & Ba, 2017). De esta manera un optimizador es el encargado de adaptar los pesos para que sean cada vez mejores. El funcionamiento básico que lleva a cabo es el cálculo del gradiente de la función de coste por cada peso de la red.

En el proceso de entrenamiento también contamos con una red neuronal convolucional denominada DPN que ya había sido entrenada con anterioridad para realizar la segmentación de imágenes de resonancia magnética de cerebros. Esta red, la usaremos como no entrenable, es decir, los pesos de este modelo no se van a ir actualizando durante el entrenamiento, pues esta la queremos para hacer la segmentación de las imágenes y calcular una función de *loss*.

La idea subyacente es que cuanto menor sea el error de la red que se encarga de la superresolución mejor será la segmentación y de aquí sacamos la métrica de error del coeficiente DICE que se explicará a continuación. En definitiva, los pesos de la Unet 3D se van a ir actualizando mientras que los de la DPN se van a mantener fijos. Para ello se concatenaron las dos redes y se evaluaron conjuntamente las métricas.

En cuanto a las funciones de error se han contemplado son las siguientes:

- *Mixed loss*, en esta se van a combinar diferentes métricas entre las que encontramos la correlación, el *mean absolute error* de los gradientes locales, el SSIM y *mean absolute error* de las intensidades de la imagen. El objetivo de usar esta función de pérdidas es minimizar las diferencias entre predicciones y valores reales.

- Error absoluto ponderado, esta calcula el error absoluto ponderado por el gradiente local, dando más peso a las áreas con alto gradiente.
- Coeficiente de DICE, se calcula el complemento del coeficiente de DICE entre predicciones y etiquetas, este usa la similitud entre los dos conjuntos. El objetivo es maximizar la segmentación precisa entre las predicciones y etiquetas verdaderas.
- *Feature loss*, mide diferencias entre las características de las predicciones y las imágenes verdaderas, hace uso de una convolución aleatoria en la cual se extraen las características y luego se calcula la media de la diferencia entre estas características. El objetivo es minimizar esta diferencia.

En nuestro caso, no se hizo necesario generar un conjunto de datos de validación ya que por cada resolución teníamos casi 3000 imágenes diferentes para hacer el training, es decir en total de más 20000 imágenes, por lo que el *overfitting* es improbable.

Para realizar el entrenamiento se usaron 2000 *epochs*. Por cada *epoch* se procesaban 20 casos. Una vez finalizado el primer ciclo los pesos se guardan, a partir de este momento en cada ciclo se comprueba si se ha producido una reducción del valor de error y si es cierto entonces se sustituyen los pesos guardados y hay una actualización del modelo.

3.3.2. Test

Una vez finalizada la etapa de entrenamiento comienza la fase del *test* para realizar una evaluación del aprendizaje de la red y poder ver si la generación de imágenes de alta resolución a partir de baja resolución ha sido satisfactoria. Se ha generado un script de 'test' en el cual se hace una evaluación de cada una de las siete diferentes resoluciones con las que contamos.

Como sabemos, en el *test* no se pueden usar datos que se hayan usado durante el entrenamiento. Por ello, necesitamos datos nuevos que introducir para ver cómo resuelve el problema. Del conjunto de imágenes se reservaron para el *test* en exclusivo 35 casos de las distintas resoluciones, es decir 5 de cada resolución.

Las diferentes métricas de evaluación que se han trabajado en este TFG son las siguientes:

- PSNR (Peak to Signal Noise Ratio), es un término que se usa para definir la relación entre la energía máxima posible de una señal y el ruido que afecta a la calidad de su representación fidedigna. En este caso cuanto mayor sea la relación, es decir cuanto mayor sea el PSNR es que mejor se ha reconstruido la imagen degradada para que coincida con la imagen original y mejor es el algoritmo reconstructivo (National Instruments, 2023). La expresión matemática sería la ecuación 5.

$$PSNR = 20 \left(\frac{MAX_f}{\sqrt{MSE}} \right) \quad (5)$$

Donde como vemos es una expresión que utiliza *Mean Squared Error* o error cuadrático medio que se define como en la ecuación 6.

$$MSE = \frac{1}{mn} \sum_0^{m-1} \sum_0^{n-1} ||f(i, j) - g(i, j)||^2 \quad (6)$$

- SSIM (Structural Similarity Index), esta métrica se usa para medir la similitud entre dos imágenes dadas, en nuestro caso las imágenes que va a predecir nuestra red a partir de las imágenes de entrada y las originales. Va a cuantificar las diferencias entre ambas como el sistema visual humano va a tener en cuenta tres características de las imágenes: luminancia, contraste y estructura combinándolas todas en su expresión final (Datta, 2021). Tiene un rango de [0, 1], cuanto más cerca de la unidad se encuentre veremos una mayor similitud entre ambas imágenes (Datta, 2021). Su expresión es como en la ecuación 7.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C1)(2\sigma_{xy} + C2)}{(\mu_x^2 + \mu_y^2 + C1)(\sigma_x^2 + \sigma_y^2 + C2)} \quad (7)$$

- Correlación, es una métrica que se trata de una medida de regresión para determinar la relación de índole lineal que se produce entre dos variables. Permite que se establezca una relación cuantitativa lineal entre dos variables en nuestro caso imágenes. Un valor de correlación de 1 nos indica una correlación positiva entre las imágenes, por lo que estas están altamente relacionadas mientras que si es cercano a -1 nos indica que las imágenes son inversamente proporcionales (DELSOL, 2021). (ver ecuación 8)

$$CC(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (8)$$

- Índice Dice, nos va a dar a conocer el índice de acierto en la segmentación, es una medida de coincidencia en las etiquetas entre la imagen de segmentación ofrecida como entrada, y la imagen de salida ofrecida por la red, a nivel de vóxel. Cuanto más cercano al 1 más alta será esta coincidencia, su expresión se encuentra en la ecuación 9.

$$DICE(x, y) = \frac{2|x \cap y|}{|x| + |y|} \quad (9)$$

Tras la aplicación del *test*, cada uno de los casos que se han introducido como input en la red ya entrenada tendrán sus correspondientes imágenes de superresolución y sus segmentaciones.

4. RESULTADOS

A continuación, se exponen los resultados obtenidos con los diferentes experimentos.

4.1. RESULTADOS DEL TRAINING

En primer lugar, se realizaron diferentes experimentos cortos para obtener los resultados de las diferentes arquitecturas de redes neuronales convolucionales que se pensó que podrían resolver de manera correcta nuestro problema comparándolas, utilizando la métrica de correlación. Estos se obtienen con un solo caso de la resolución más baja, es decir de resolución 7 mm³ pues esta es la más difícil de resolver. Estos resultados (ver tabla 4.1) son los que ayudaron a tomar la decisión de la arquitectura correcta y explicada en los apartados anteriores. Se compararon la red DPN, una Red SR3D que una red de superresolución lineal donde no se usan capas de pooling, la red UNET con dos canales de entrada donde el canal auxiliar era una imagen de ejemplo de HR y la red UNET descrita. Como se puede observar, el mejor resultado se obtuvo con la red UNET3D propuesta.

	DPN	SR_3D	UNET3D_prior	UNET3D
Correlación	0,9077	0,8388	0,9326	0,9386

Tabla 4.1. Resultados de la correlación de las diferentes arquitecturas.

Una vez elegida la arquitectura, se realizaron una serie de modificaciones para ver si se puede obtener un mejor resultado, manteniendo en todo momento el número de filtros en de 8 en la primera capa. Estas modificaciones implican añadir ruido o el uso de una UNET3D no residual. Las modificaciones no mejoraron los resultados como se puede ver en la tabla 4.2.

	Correlación
UNET3D	0,9386
+ NOISE (0,1)	0,9335
+ NOISE (0,0.5)	0,9292
+ NOISE (0,0.25)	0,9136
UNET3D (no residual)	0,9322

Tabla 4.2. Resultados modificación arquitectura UNET3D.

4.2. RESULTADOS DEL TEST

Una vez obtenidos los mejores pesos durante el *training*, se evaluó con datos de test. Es decir, con imágenes que no ha visto antes para ver la capacidad de generalización de las distintas redes.

En primer lugar, se evaluó un modelo el cual no distinguía entre las diferentes resoluciones de las imágenes con las que contábamos. Es decir, esta red se entrenaba de golpe con imágenes con resoluciones muy diferentes, lo que se traduce en una búsqueda de pesos óptimos más compleja. Para obtener el resultado de esta obtenemos las métricas de evaluación PSNR, SSIM y correlación (explicadas con anterioridad en 3.3.2) para los datos de *test* de la base de datos

HCP tras un entrenamiento de 1000 *epochs*. Los resultados obtenidos con una única red son los expuestos en la tabla 4.3. Como se puede ver, los resultados decrecen en calidad conforme se reduce la resolución de las imágenes como era de esperar.

	Res 1	Res 2	Res 3	Res 4	Res 5	Res 6	Res 7	Media
PSNR	42,8099	40,7905	33,7578	32.3124	29,2554	28,6088	28,3211	33,9794
SSIM	0,9744	0,9595	0,8817	0,8056	0,7220	0,7096	0,7458	0,8283
Correlación	0,9949	0,9931	0,9670	0,9353	0,9077	0,8944	0,9041	0,9452

Tabla 4.3. Resultados de una única red entrenada para todas las resoluciones.

Para evaluar la hipótesis de que 7 redes especializadas funcionarían mejor que una general, se comparó los resultados de la tabla 4.3 con los resultados de las 7 redes entrenadas (una para cada resolución). De esta manera el entrenamiento de las redes es más sencillo y óptimo como se puede ver en los resultados de la tabla 4.4. Para obtener un resultado más visual se expone la figura 4.1 en la que vemos las gráficas de los resultados de las métricas de evaluación (PSNR, SSIM y correlación) obtenidas por la red única y por las siete redes diferentes en base a las resoluciones. A partir de estos resultados se puede ver que el enfoque multi-red es mejor que el uso de una sola red en casi todas la métricas y resoluciones.

	Res 1	Res 2	Res 3	Res 4	Res 5	Res 6	Res 7	Media
PSNR	45,9191	42,3212	36,1305	34,7586	33,1195	32,5883	32,6416	36,7823
SSIM	0,9883	0,9691	0,9122	0,8577	0,8371	0,8124	0,7903	0,8489
Correlación	0,9974	0,9953	0,9808	0,9692	0,9612	0,9562	0,9558	0,9731

Tabla 4.4 Resultados para las siete redes entrenadas por separado.



Figura 4.1. Gráfica valores PSNR, correlación y SSIM, en naranja los resultados obtenidos de las 7 redes entrenadas cada una para una resolución, en azul los resultados de una única red entrenada para todas las resoluciones.

Para comparar los resultados de nuestro método con algún método del estado del arte para la mejora de resolución de imágenes, se usó SynthSR como referencia. Se realizó la superresolución de imágenes LR tanto con nuestro método como con el método de SynthSR.

Para asegurarnos de una correcta comparación usamos datos de *test* (315 casos) de una base de datos diferente a la que se ha usado para realizar el entrenamiento, el motivo de esto es que se comparen las redes de manera que ninguna haya visto las imágenes con las que va a trabajar antes. Las métricas utilizadas para esta comparación son las mismas con las que se ha estado trabajando durante los experimentos, obteniendo los resultados expuestos en la tabla 4.5. Como se puede observar los resultados del método propuesto es mejor que SynthSR para todas las métricas.

		SynthSR	Nuestro método
PSNR	Res 1	20,4805	38,4631
	Res 2	20,4225	36,0547
	Res 3	20,0837	29,1124
	Res 4	19,9159	27,8956
	Res 5	19,7993	27,1153
	Res 6	19,7062	26,5342
	Res 7	19,3287	26,7494
	Media	19,9624	30,2749
SSIM	Res 1	0,4828	0,8945
	Res 2	0,4734	0,9206
	Res 3	0,4333	0,8217
	Res 4	0,4025	0,7800
	Res 5	0,3839	0,7629
	Res 6	0,3761	0,7141
	Res 7	0,3615	0,7522
	Media	0,4162	0,8066
Correlación	Res 1	0,7225	0,9937
	Res 2	0,7630	0,9880
	Res 3	0,6930	0,9630
	Res 4	0,6799	0,9504
	Res 5	0,6703	0,9455
	Res 6	0,7630	0,9365
	Res 7	0,6221	0,9352
	Media	0,7020	0,9589

Tabla 4.5. Tabla de comparación de resultados entre SynthSR y el método generado en el trabajo.

Una vez obtenidos los resultados sobre el *dataset* que aún no había visto la red (QC3000), se procedió a entrenar esta misma red con los nuevos datos de esta BD para que cuente con datos más variados de los que aprender y pueda hacer mejores estimaciones de los pesos, con el objetivo de obtener un mejor modelo. Tras el entrenamiento se realizó la evaluación que se puede ver en la tabla 4.6. donde se aprecia la mejora de los resultados.

	Res 1	Res 2	Res 3	Res 4	Res 5	Res 6	Res 7	Media
PSNR	45,6565	42,3027	36,2471	34,5989	33,2109	32,4124	32,6594	36,7268
SSIM	0,9867	0,9602	0,9096	0,8461	0,8378	0,7796	0,7404	0,8658
Correlación	0,9970	0,9953	0,9814	0,9687	0,9621	0,9546	0,9562	0,9736

Tabla 4.6. Resultados con la red entrenada con todos las BD.

Finalmente, a estos resultados obtenidos con el método realizado para este TFG también se añadió la métrica basada en el índice de DICE en base a que el objetivo de mejorar la resolución de las imágenes además de aumentar la calidad visualmente es que puedan ser segmentadas de manera correcta por una red entrenada específicamente para la segmentación. Para incluir esta métrica se generó una tercera y última red, partiendo de la U-Net 3D a la que se le añade un discriminador en este caso una red de segmentación congelada dando así una nueva función de error que es el índice DICE.

En este caso se van a calcular dos índices DICE, el primero o DICE 0 es el que se genera usando las imágenes LR y el segundo DICE es el resultado sobre la imagen reconstruida. Se ve claramente como en las resoluciones más bajas el índice aumenta lo que indica que la reconstrucción con esta red es prometedora. Los resultados obtenidos de esta los encontramos en la tabla 4.7, como vemos la mayoría de las métricas sobre todo a mayor resolución mejoran.

	Res 1	Res 2	Res 3	Res 4	Res 5	Res 6	Res 7	Media
PSNR	43,7731	41,3512	36,2760	34,8919	33,2558	32,8369	33,2334	36,5769
SSIM	0,9871	0,9560	0,9135	0,8682	0,8365	0,8308	0,8321	0,8891
Correlación	0,9955	0,9940	0,9817	0,9698	0,9627	0,9588	0,9615	0,9748
DICE 0	0,9325	0,9240	0,8577	0,8158	0,7564	0,7547	0,7389	0,8257
DICE	0,9317	0,9244	0,8798	0,8543	0,8217	0,8196	0,7891	0,8601

Tabla 4.7. Resultados con la red entrenada con todos las BD más índice DICE.

Los resultados de segmentación obtenidos en este caso también se compararon con un método segmentación robusto a la resolución que se usa en la actualidad denominado SynthSeg (Billot et al., 2023). Para realizar esta comparación, usaron imágenes con la resolución mejorada, estas imágenes ya mejoradas se usaron como entrada del método de SynthSeg el cual nos va a generar la segmentación. Por otro lado, SynthSeg tiene la capacidad de hacer la segmentación a partir de imágenes de baja resolución, por lo que le pasamos como imágenes de entrada las mismas imágenes de *test* de LR con las que ha trabajado nuestra red. Por último, para obtener una buena comparación de resultados, vamos a hacer uso de la herramienta SynthSR para mejorar la resolución de las imágenes LR con otro método y posteriormente hacer la segmentación de estas imágenes como se ha hecho con las imágenes mejoradas por nuestro modelo. Los resultados obtenidos son de índice DICE el cual nos da el nivel de calidad de segmentación y los podemos ver en la tabla 4.8 y figura 4.2.

		LR	7 redes	SynthSR	1 red
DICE	Res 1	0,9904	0,9907	0,9049	0,9880
	Res 2	0,9726	0,9734	0,9002	0,9718
	Res 3	0,9237	0,9267	0,8791	0,9230
	Res 4	0,8719	0,9021	0,8533	0,8713
	Res 5	0,7678	0,8692	0,8096	0,7631
	Res 6	0,7282	0,8568	0,7965	0,7242
	Res 7	0,7128	0,8208	0,7648	0,7066
	Media	0,8524	0,9057	0,8441	0,8497

Tabla 4.8. Comparación de resultados entre SynthSeg y el método generado.

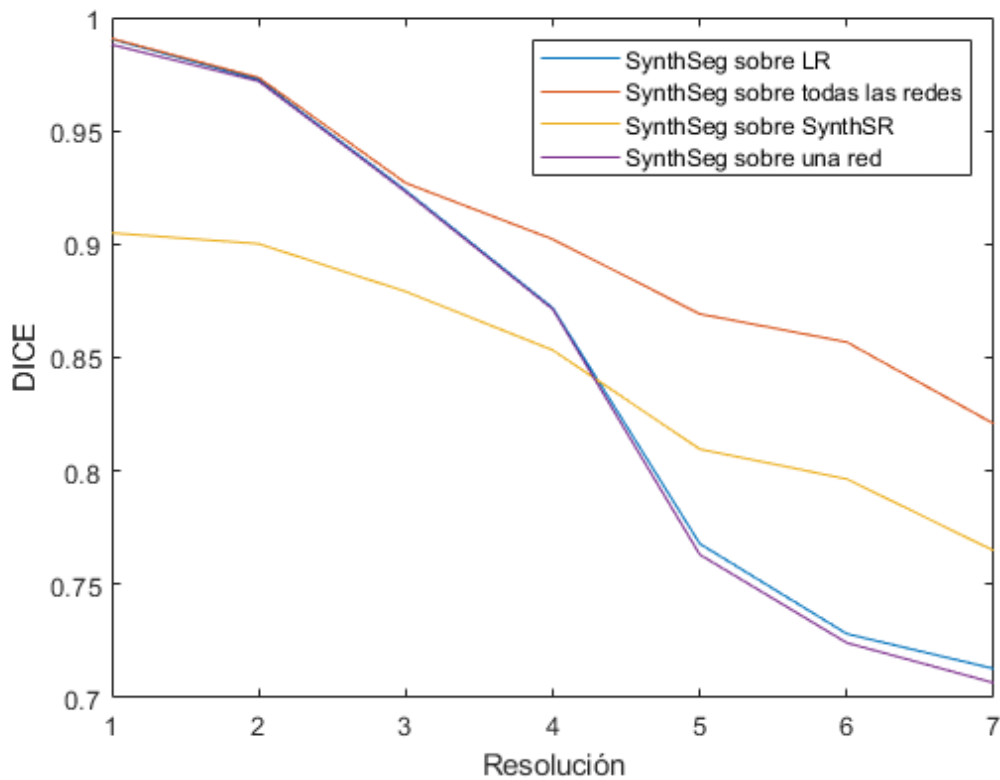


Figura 4.2. Gráfica valores de índice DICE.

Como se puede observar se ve la tendencia de disminución del índice DICE conforme la resolución va disminuyendo en todos métodos. Sin embargo, se observa como la línea roja, la cual es el método generado en el trabajo consigue mantener estos valores del índice más elevados que el resto de métodos, lo que supone una mejor segmentación.

Además, del uso de métricas de evaluación podemos hacer una comparación visual y cualitativa de las diferencias entre ambos métodos. Para poder observarlo de forma correcta se van a mostrar las imágenes de alta resolución que se han obtenido a partir de las resoluciones más bajas con las que se ha trabajado tanto con el método generado en el trabajo como con SynthSR (figura 4.3 y 4.4).

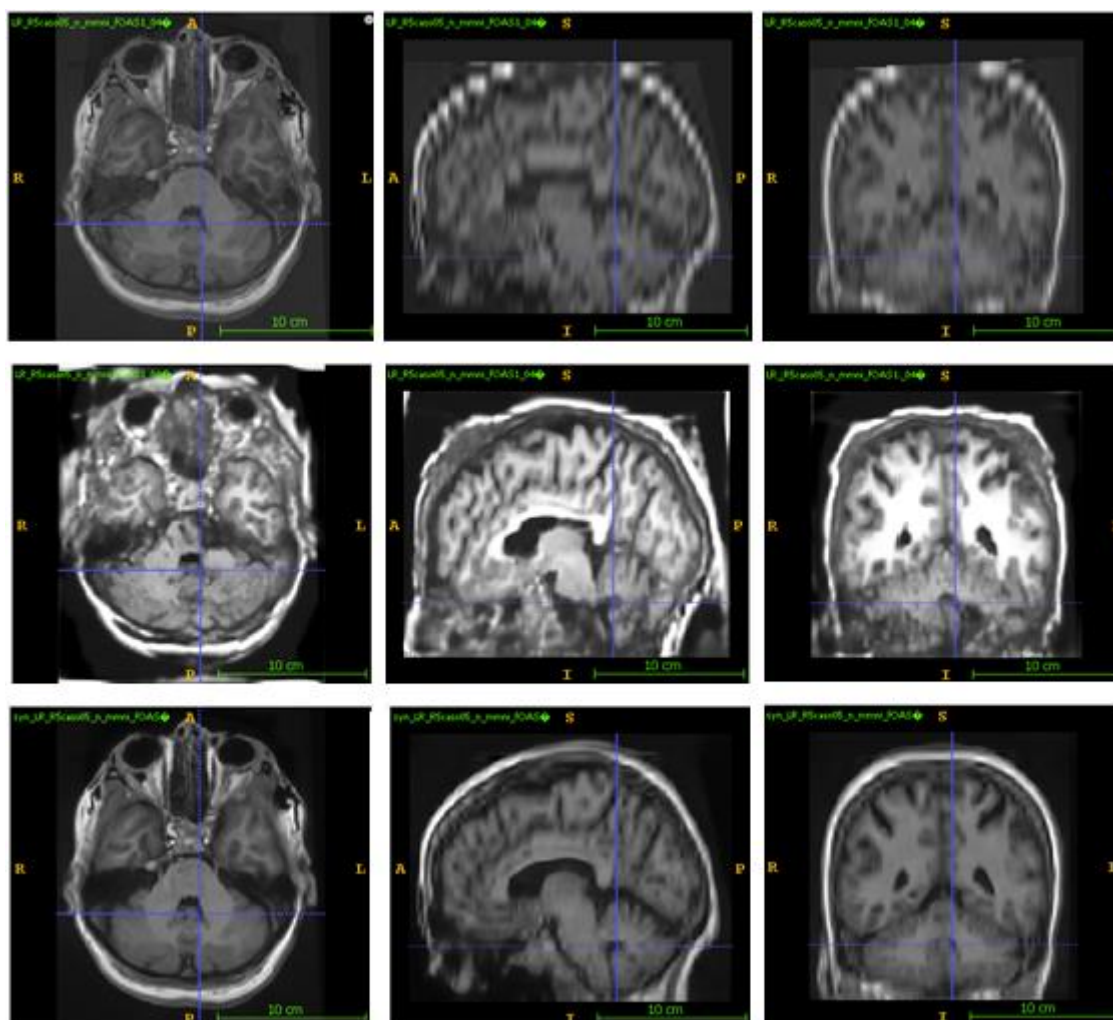


Figura 4.3. De arriba abajo. Imagen de baja resolución (5 mm) a mejorar. Imagen de superresolución obtenida por SynthSR. Imagen de superresolución obtenida por nuestro método.

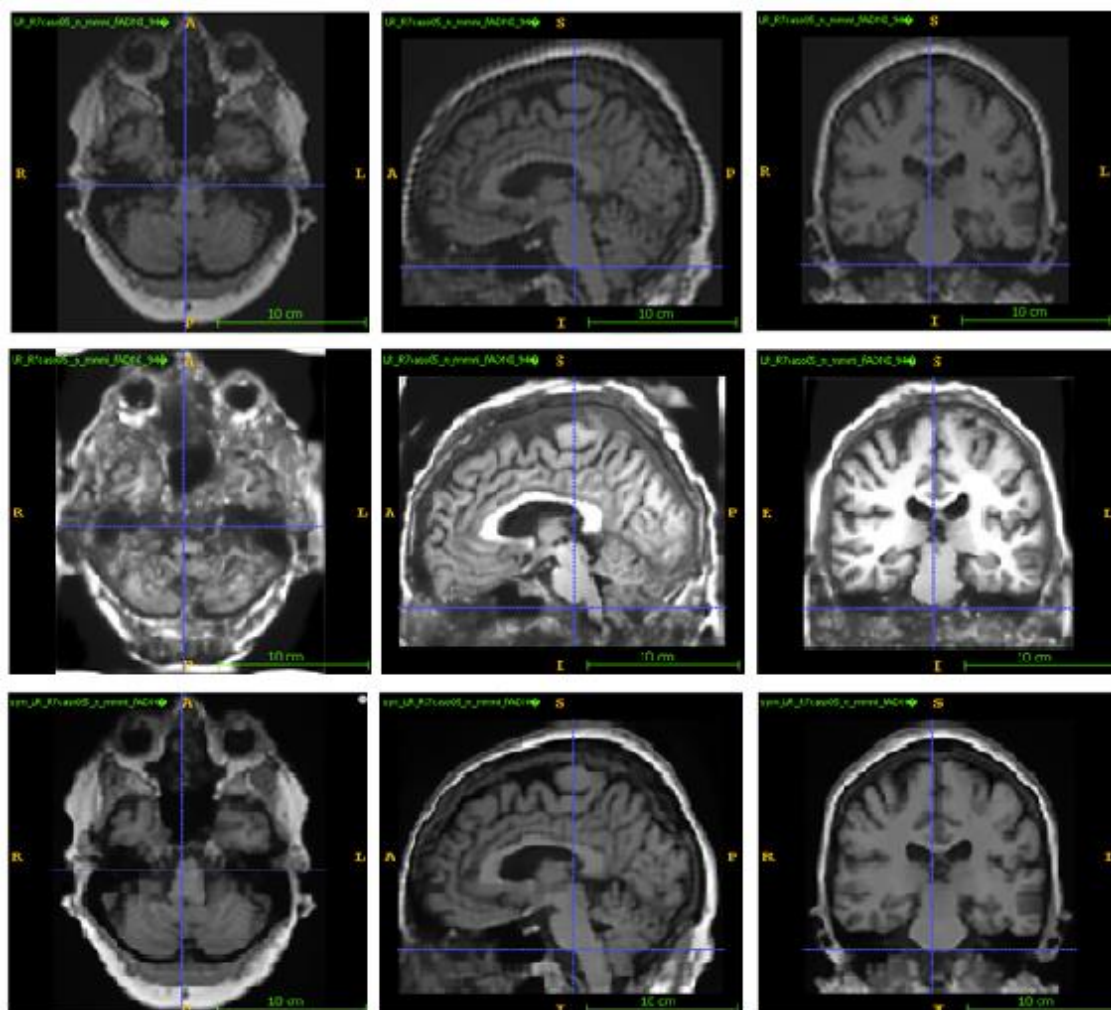


Figura 4.4. De arriba abajo. Otro ejemplo de una Imagen de baja resolución (7 mm) a mejorar. Imagen de superresolución obtenida por SynthSR. Imagen de superresolución obtenida por nuestro método.

De igual manera, se puede evaluar visualmente la calidad de la segmentación basada en las imágenes producidas con los distintos métodos. La figura 4.5 y 4.6 muestra un ejemplo de estas segmentaciones.

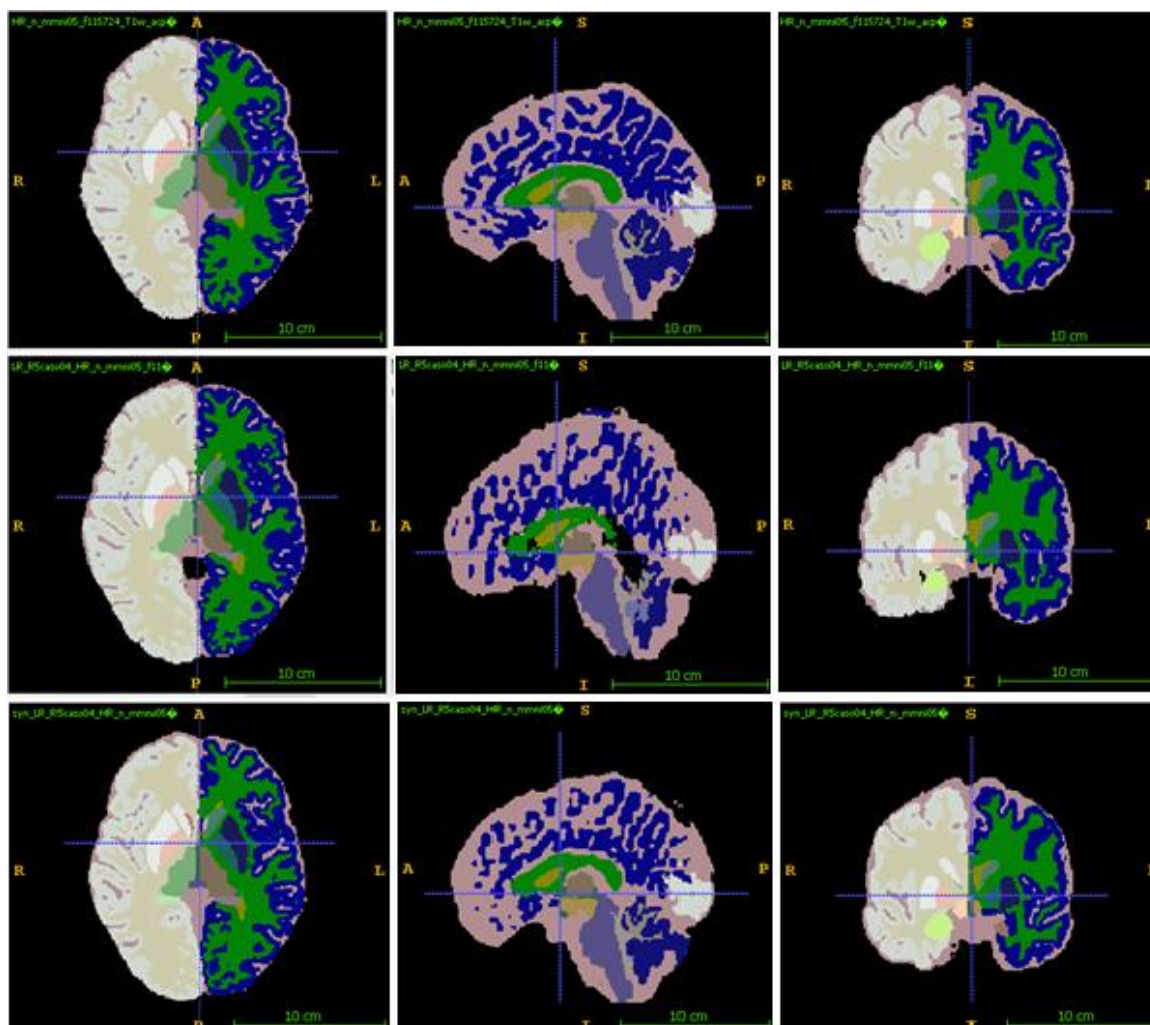


Figura 4.5. De arriba abajo, etiquetas de segmentación de la imagen original HR. Segmentación con SynthSeg de una imagen de 5mm de resolución. Segmentación con SynthSeg haciendo uso del método generado en el trabajo para mejorar la resolución a partir de la misma imagen de resolución de 5 mm.

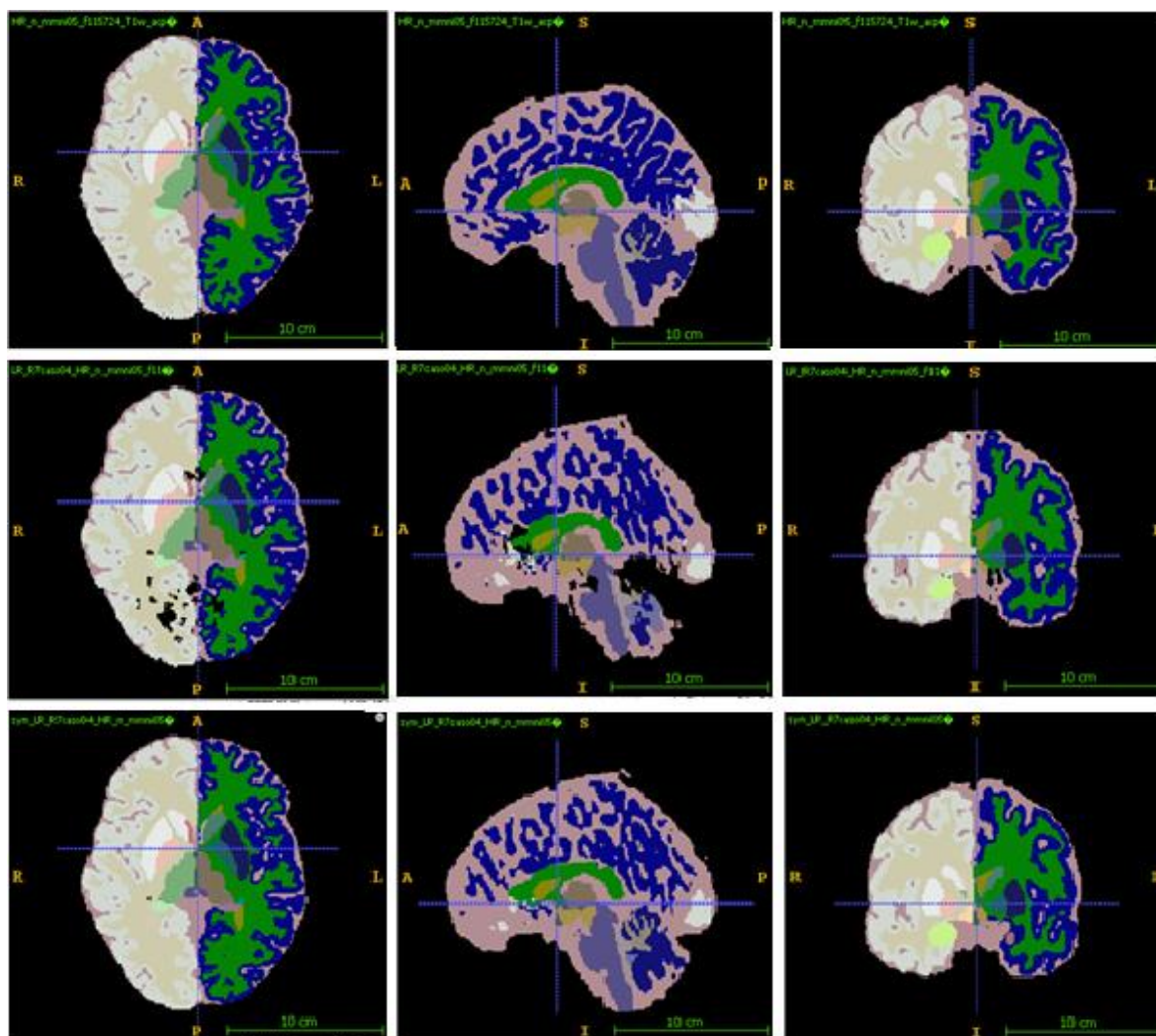


Figura 4.6. De arriba abajo, etiquetas de segmentación de la imagen original HR. Segmentación con SynthSeg de una imagen de 7mm de resolución. Segmentación con SynthSeg haciendo uso del método generado en el trabajo para mejorar la resolución a partir de la misma imagen de resolución de 7 mm.

5. DISCUSIÓN Y CONCLUSIONES

5.1. DISCUSIÓN

Una vez expuestos los resultados se va a pasar a hacer un análisis de estos y una discusión.

En primer lugar, se hicieron experimentos con una única red a partir de todas las imágenes independientemente de sus resoluciones, el segundo experimento trataba de hacer una red por cada una de las resoluciones. Como vemos en las tablas de resultados 4.3 y 4.4, se cumple la hipótesis principal basada en la filosofía clásica de *divide y vencerás*. Con los buenos resultados obtenidos con 7 redes diferentes a partir de ese momento el resto de los entrenamientos se hicieron usando ese enfoque.

Como podemos ver en la Tabla 4.4 los resultados obtenidos de las diferentes métricas de evaluación siguen la tendencia esperada. En las resoluciones más altas estos resultados son mejores que en las bajas debido a que para la red es más sencillo aprender y obtener los pesos óptimos en estas primeras ya que no distan tanto de la imagen original HR. Los resultados obtenidos con los datos de *test* de la base de datos usada han sido positivos ya que se obtienen buenos resultados para todas las métricas de evaluación.

Como se ha comentado, para poder ver verdaderamente el potencial de las redes generadas se ha hecho una comparación con el estado del arte de superresolución de imágenes, es decir con SynthSR. Ambas redes han sido puestas a prueba con las mismas imágenes de una BD diferente a la usada para el entrenamiento. De esta manera, como se ha explicado en el apartado 4.2 ninguna cuenta con ventaja. En la Tabla 4.5 podemos ver los resultados de las métricas obtenidas viendo también esa tendencia descendente de los productos obtenidos para cada resolución y además se ve claramente cómo estos resultados son favorables a nuestro método pues se mantienen en todo momento superiores a los obtenidos por SynthSR.

Como se puede observar en las figuras 4.3 y 4.4 ambos métodos hacen una buena reconstrucción. Sin embargo, puede haber mejoras, por un lado, vemos como en las imágenes generadas por SynthSR la zona central la hace de forma correcta, pero los bordes como la corteza o la zona del cerebelo no se llegan a definir de la forma correcta y en algunas estructuras cambia las formas. Por otro lado, el método generado para el TFG resuelve de manera más fidedigna estas zonas de cerebelo y corteza craneal permitiendo ver las zonas de los bordes en las que también hay información importante.

Por otro lado, como se ha comentado se ha analizado la capacidad del método propuesto haciendo uso de la segmentación. Para hacer esto contamos con las etiquetas de cada parte del cerebro de las imágenes originales de alta resolución. El objetivo es que si se realiza una buena superresolución de la imagen de baja resolución entonces la segmentación será más correcta. Para analizar los resultados se ha hecho una comparación con el método SynthSeg el cual realiza segmentación y es capaz de trabajar con imágenes de baja resolución en el caso que sea necesario. Para realizar este análisis se ha escogido la métrica de evaluación del índice DICE el cual mide la similitud entre las etiquetas producidas por la red tras realizar la superresolución y las etiquetas originales.

De forma cuantitativa en la tabla 4.8 se puede observar como el índice DICE comparado en las resoluciones más elevadas es similar por lo que se genera una buena segmentación de estas

resoluciones (de la 1 a la 4). A partir de estas resoluciones el modelo generado en el trabajo es capaz de obtener un valor numérico de la métrica mayor que SynthSeg lo que nos indica que es capaz de realizar una mejora de la resolución y a partir de eso SynthSeg realiza la segmentación correcta. Por otro lado, se ve que los valores de la métrica generados al usar el método de SynthSR para realizar la mejora de resolución y posteriormente hacer uso de SynthSeg para la segmentación son los menores, lo que indica una mejora de resolución peor que nuestro método.

Como se puede observar en las imágenes, las segmentaciones realizadas a partir de imágenes de baja resolución (5 y 7 mm) (Figura 4.5 y 4.6) las cuales antes han sido tratadas por la red para realizar mejora de resolución sobre ellas obtienen mejores resultados haciendo uso antes del método generado en el trabajo, ya que con SynthSeg aplicado directamente aparecen zonas que no es capaz de etiquetar y segmentar de manera correcta, esto puede ser debido a una mala realización de aumento de resolución antes de generar la segmentación. De esta manera nuestro método obtiene resultados prometedores en todas las métricas de evaluación que se han estudiado.

5.2. CONCLUSIONES

El presente Trabajo de Fin de Grado ha diseñado e implementado de un conjunto de redes neuronales para la mejora de la resolución de imágenes de RMN cerebral. Para llevarlo a cabo se ha hecho uso de una base de datos en la cual encontramos 75 imágenes de alta resolución y a partir de estas muestras y multitud de transformaciones realistas se han generado las diferentes imágenes de baja resolución con las que se ha entrenado y evaluado la red generada.

Este problema de mejora de la resolución de imágenes es uno de los campos de investigación más activos. La solución propuesta se ha comparado con un método presentado recientemente (SynthSR), obteniendo mejores resultados en dos *datasets* distintos, uno similar al de entrenamiento y otro totalmente distinto lo cual da información sobre la generabilidad del enfoque propuesto.

Por otro lado, se debe tener en cuenta que estos resultados obtenidos pueden ser mejorados aportando datos de mayor calidad, usando diferentes tipos de procesado de las imágenes, funciones de error o clasificando las imágenes de forma diferente.

Finalmente, con el presente trabajo se hace visible de nuevo que las redes neuronales son una herramienta muy potente en el ámbito de la imagen médica, con una capacidad de aprendizaje y de progreso alta.

6. LÍNEAS FUTURAS

Como se ha expuesto los resultados del problema tratado obtenidos son buenos y prometedores, aunque seguramente mejorables con más tiempo y datos. Como líneas futuras para mejorar los resultados se podría hacer lo siguiente:

- Aumentar el número de datos entrenamiento. Esto aumentaría la diversidad de los datos y mejoraría la generalización.
- Incorporar información a priori como imágenes de HR registradas no-linealmente.
- Por otro lado, se podría mejorar la clasificación de las imágenes por apariencia en lugar de resolución para agrupar mejor las imágenes y homogeneizar los datos de entrenamiento.
- Finalmente, a modo de línea futura también se podría realizar una prueba de aplicación clínica y su validación, lo que implica colaboración con profesionales de salud y recopilación de datos clínicos relevantes para asegurar su utilidad y eficacia.

7. OBJETIVOS DE DESARROLLO SOSTENIBLE

Para finalizar la memoria en este apartado se va a ver el grado de relación que tiene el Trabajo de Fin de Grado con los Objetivos de Desarrollo Sostenible de la agenda 2030 (ODS). Para esto se va a hacer uso de una tabla en la que se presentan los objetivos y las relaciones, finalmente se procederá a explicar cada una de las relaciones resultantes.

Objetivos de desarrollo sostenible	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza				X
ODS 2. Hambre cero				X
ODS 3. Salud y bienestar	X			
ODS 4. Educación de calidad		X		
ODS 5. Igualdad de género				X
ODS 6. Agua limpia y saneamiento				X
ODS 7. Energía asequible y no contaminante				X
ODS 8. Trabajo decente y crecimiento económico			X	
ODS 9. Industria, innovación e infraestructuras	X			
ODS 10. Reducción de las desigualdades		X		
ODS 11. Ciudades y comunidades sostenibles				X
ODS 12. Producción y consumo responsables				X
ODS 13. Acción por el clima				X
ODS 14. Vida submarina				X
ODS 15. Vida de ecosistemas terrestres				X
ODS 16. Paz, justicia e instituciones sólidas				X
ODS 17. Alianzas para lograr objetivos				X

Tabla 7.1. Objetivos de desarrollo sostenible de la agenda de 2030.

En este trabajo se pretende mejorar la calidad de imagen de imágenes de resonancia magnética del cerebro y esto puede contribuir a varios Objetivos de Desarrollo Sostenible. Para comenzar está relacionado con 4 objetivos. En primer lugar con el ODS 3 de salud y bienestar, debido a que la mejora de la calidad de imagen puede facilitar el diagnóstico haciéndolo más preciso y permitiendo la detección temprana de enfermedades, ODS 4 de educación de calidad pues haciendo esta mejora se podría ayudar en la educación tanto médica como científica, ODS 9 industria, innovación e infraestructuras en este caso generar una nueva técnica de mejora de resolución puede suponer innovación en el campo de imagen médica lo que puede contribuir en el avance de la industria y por último, ODS 10 de reducción de desigualdades, al mejorar la calidad de imagen se puede reducir la desigualdad que puede haber entre aquellas personas que se pueden permitir tener acceso a máquinas de alta calidad y aquellas que no cuentan con estos medios por determinadas limitaciones.

8. BIBLIOGRAFÍA

- Alcañas Martínez, U. (2015). *Resonancia Magnética, los protocolos y su práctica*. Obtenido de <https://protocolosresonanciamagnetica.org/author/ursula/>
- Artola Moreno, Á. (2019). Clasificación de imágenes usando redes neuronales convolucionales en Python. <https://idus.us.es/handle/11441/89506>
- Avants, B., Tustison, N., & Song, G. (2008). Advanced normalization tools (ANTs). *Insight J*, 1–35. <https://doi.org/10.54294/uvnhin>
- Azpilcueta, R. (2020). *Dispositivo de adquisición de señales mioeléctricas*.
- Bello, E. (2023). Redes Generativas Antagónicas: La IA que crea obras de arte desde cero. Thinking for Innovation. <https://www.iebschool.com/blog/redes-generativas-antagonicas-tecnologia/>
- Billot, B., Greve, D. N., Puonti, O., Thielscher, A., Van Leemput, K., Fischl, B., Dalca, A. V., Iglesias, J. E., & ADNI. (2023). SynthSeg: Segmentation of brain MRI scans of any contrast and resolution without retraining. *Medical Image Analysis*, 86, 102789. <https://doi.org/10.1016/j.media.2023.102789>
- Carmona Benítez, M. (2014). *Clasificación automática del tipo de árbol a partir de las características de sus hojas*. Madrid.
- Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2016). *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)* (arXiv:1511.07289). arXiv. <http://arxiv.org/abs/1511.07289>
- Datta, P. (2021, marzo 4). All about Structural Similarity Index (SSIM): Theory + Code in PyTorch. SRM MIC. <https://medium.com/srm-mic/all-about-structural-similarity-index-ssim-theory-code-in-pytorch-6551b455541e>
- DELSOL, S. (2021, septiembre 21). Coeficiente de correlación lineal ¿Qué es? <https://www.sdelsol.com/glosario/coeficiente-de-correlacion-lineal/>
- Díaz, I. R. R. (2014). IMÁGENES DIAGNÓSTICAS: CONCEPTOS Y GENERALIDADES.
- enlace, O., Facebook, Twitter, Pinterest, electrónico, C., & aplicaciones, O. (2019, octubre 21). Visión artificial con redes convolucionales (CNN). <https://www.ellaberintodefalken.com/2019/10/vision-artificial-redes-convolucionales-CNN.html>

- Falbel, D., & Keydana, S. (2019). Posit AI Blog: Image segmentation with U-Net. <https://blogs.rstudio.com/tensorflow/posts/2019-08-23-unet/>
- Fonov, V., Evans, A. C., Botteron, K., Almli, C. R., McKinstry, R. C., Collins, D. L., & Brain Development Cooperative Group. (2011). Unbiased average age-appropriate atlases for pediatric studies. *NeuroImage*, 54(1), 313-327. <https://doi.org/10.1016/j.neuroimage.2010.07.033>
- Future Space. (2021). *Redes Neuronales y Deep Learning. Capítulo 2: La Neurona*. Obtenido de <https://www.futurespace.es/redes-neuronales-y-deep-learning-capitulo-2-la-neurona/>
- García, I. E. G. (2020). Desarrollo de una red neuronal convolucional volumétrica para la síntesis de imagen en resonancia magnética nuclear.
- Gómez Tejada, A. (2020). *Aplicaciones de "Deep Learning" en entorno ROS*. Sevilla.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Networks (arXiv:1406.2661). arXiv. <https://doi.org/10.48550/arXiv.1406.2661>
- Hernández, L. J. R. (2020). Estudio comparativo de algoritmos de súper resolución de una sola imagen basados en aprendizaje profundo.
- Herrera, J. M. (2006). Segmentación Robusta de Imágenes de RM cerebral. Director: Dra. Montserrat Robles Viejo, Dr. Luis Martí-Bonmatí. Universidad Politécnica de Valencia, Departamento de Física Aplicada.
- Iglesias, J. E., Billot, B., Balbastre, Y., Tabari, A., Conklin, J., Gilberto González, R., Alexander, D. C., Golland, P., Edlow, B. L., Fischl, B., & Alzheimer's Disease Neuroimaging Initiative. (2021). Joint super-resolution and synthesis of 1 mm isotropic MP-RAGE volumes from clinical MRI exams with scans of different orientation, resolution and contrast. *NeuroImage*, 237, 118206. <https://doi.org/10.1016/j.neuroimage.2021.118206>
- InteractiveChaos. (2018). *Esquema de la red neuronal para clasificación*.
- Kingma, D. P., & Ba, J. (2017). *Adam: A Method for Stochastic Optimization* (arXiv:1412.6980). arXiv. <https://doi.org/10.48550/arXiv.1412.6980>
- Lafuente, J., & Moreno, L. (2006). Técnica de la imagen por resonancia magnética. *Libro publicado por la sociedad española de radiología SERAN*, 9-28.
- López-Lozano, D. F. (2021). *Metodología de segmentación de instancias multiplano para la detección de tromboembolismo pulmonar*.
- Marzoa Tanco, M. (2019). Super-resolución en imágenes. <https://www.colibri.udelar.edu.uy/jspui/handle/20.500.12008/26226>

- Merino, M. (2019, marzo 31). Conceptos de inteligencia artificial: Qué son las GANs o redes generativas antagónicas. Xataka. <https://www.xataka.com/inteligencia-artificial/conceptos-inteligencia-artificial-que-gans-redes-generativas-antagonicas>
- Morell, S. (2023). Diseño e implementación de una red neuronal artificial para la segmentación de lóbulos del cerebelo a partir de imágenes multimodales de RM de alta resolución
- National Instruments. (2023). *Peak Signal-to-noise Ratio as an Image Quality Metric*. Obtenido de <https://www.ni.com/es-es/shop/data-acquisition-and-control/add-ons-for-data-acquisition-and-control/what-is-vision-development-module/peak-signal-to-noise-ratio-as-an-image-quality-metric.html>
- Raj, B. (2018). Data Augmentation: how to use Deep Learning when you have limited data.
- Rivera, D. M., Puentes, S., & Caballero, L. (2011). Resonancia magnética cerebral: Secuencias básicas e interpretación. *Universitas Médica*, 52(3), 292-306.
- Rivera Phillips, J. A. (2019). *Diagnóstico de la degradación en reboilers: un modelo en base al aprendizaje profundo*. Santiago de Chile.
- Ronneberger, O., Fischer, P., & Brox, T. (2015, mayo 18). U-Net: Convolutional Networks for Biomedical Image Segmentation. ArXiv.Org. <https://arxiv.org/abs/1505.04597v1>
- Ros García, A. (2019). Sistema de percepción de elementos viarios usando técnicas de visión por computador para aplicación en conducción autónoma. <https://repositorio.upct.es/handle/10317/8286>
- Rouhiainen, L. (2018). *Inteligencia artificial, 101 cosas que debes saber hoy sobre nuestro futuro*.
- Serreta, A. B., & Alberich-Bayarri, Á. (2018). Radiaciones no ionizantes II. Resonancia magnética. Bases físicas, equipos y control de calidad.
- Silvestre Llopis, J. (2019). *Clasificación de imágenes naturales y multi-espectrales mediante redes neuronales convolucionales y normalización divisiva*. Valencia.
- Soto, P. S. (2021). Diseño de una red neuronal convolucional para la segmentación de estructuras subcorticales cerebrales.
- Velasco, L. (2020). *Optimizadores en redes neuronales profundas: un enfoque práctico*. Obtenido de <https://velascoluis.medium.com/optimizadores-en-redes-neuronales-profundas-un-enfoque-pr%C3%A1ctico-819b39a3eb5>
- Vilarroig, C. A. (2019). Diseño y desarrollo de un método de superresolución en imágenes de RM usando Deep Learning.

Yushkevich, P. A., & Gerig, G. (2017). ITK-SNAP: An Intractive Medical Image Segmentation Tool to Meet the Need for Expert-Guided Segmentation of Complex Medical Images. *IEEE Pulse*, 8(4), 54-57. <https://doi.org/10.1109/MPUL.2017.2701493>

Parte II.

PRESUPUESTOS

Índice

1.	INTRODUCCIÓN	68
2.	PRESUPUESTO DESCOMPUESTO	68
2.1.	PRESUPUESTO MANO DE OBRA	68
2.2.	PRESUPUESTO SOFTWARE	68
2.3.	PRESUPUESTO HARDWARE	69
3.	PRESUPUESTO TOTAL	69
3.1.	PRESUPUESTO DE EJECUCIÓN MATERIAL	69
3.2.	PRESUPUESTO DE EJECUCIÓN POR CONTRATA	70

1. INTRODUCCIÓN

En el presente apartado se realiza un estudio de los presupuestos que se deben tener en cuenta a la hora de la realización del TFG expuesto anteriormente, con el objetivo de realizar estimación económica de los costes del proyecto. Para realizarlo se han descompuesto los distintos costes en mano de obra, hardware y software para que los cálculos sean más sencillos y que se pueda apreciar los costes de cada uno. Al final del capítulo se exponen los presupuestos totales tanto de ejecución de material como ejecución por contrata. Para hacer los cálculos se ha usado el programa Arquímedes que sirve para hacer el cálculo de presupuestos de proyectos.

2. PRESUPUESTO DESCOMPUESTO

2.1. PRESUPUESTO MANO DE OBRA

Para llevar a cabo el presente TFG se ha necesitado a un estudiante del Grado de Ingeniería Biomédica en su último año, que ha sido el encargado de desarrollar y dar forma al proyecto, un tutor doctor en Ingeniería Informática encargado de dirigir, supervisar y enseñar al estudiante.

Para poder comenzar a hacer cálculos se realiza una búsqueda para conocer el sueldo base de los ingenieros. El sueldo base del *senior* es de 59500 € mientras que del *junior* es 29000 €. Considerando que la jornada laboral de ambos es de 8 horas, sin contar fines de semana y vacaciones, se obtienen 224 jornadas laborales al año lo que vienen siendo 1792 h/año. Con el sueldo base y las horas trabajadas se puede deducir el salario por hora de ambos, del ingeniero biomédico *junior* vemos que es de 16,18 €/h mientras que para el ingeniero informático *senior* es de 33,22 €/h. Los resultados se muestran en la tabla 2.1.

Tipo	Cantidad (h)	Coste unitario (€/h)	Coste total(€)
Ingeniero biomédico <i>junior</i>	300	16,18	4 854,00
Ingeniero informático <i>senior</i>	100	33,22	3 322,00
Subtotal			8176,00

Tabla 2.1. Presupuesto de mano de obra.

2.2. PRESUPUESTO SOFTWARE

A continuación, se va a estimar el coste del software, se ha tenido en cuenta el coste de las licencias de cada uno de los programas utilizados. El único programa que supone costes es MATLAB 2023^a, su licencia es anual de 12 meses con un coste fijo. Con estos datos calculamos el factor de amortización.

Tipo	Cantidad (Uds.)	Factor amortización	Precio (€)	Coste total (€)
MATLAB 2022a	1	5/12	262,00	21,83
ANTs	1	-	0,00	0,00
ITK-SNAP	1	-	0,00	0,00
Notepad ++	1	-	0,00	0,00
Subtotal				21,83

Tabla 2.2. Presupuesto de software.

2.3. PRESUPUESTO HARDWARE

Para terminar con los presupuestos descompuestos se analizan los costes del hardware usado. Debemos tener en cuenta el equipo u ordenador usado en el laboratorio junto con la GPU o tarjeta gráfica usada, para calcular sus costes usamos un factor de amortización que se obtiene teniendo en cuenta el periodo de vida útil de los elementos. E trabajo se ha llevado a cabo durante 5 meses y la vida útil del ordenador y de la GPU generalmente es de 6 años lo que vienen siendo 72 meses. El factor de amortización en este caso es 5/72, el coste total se puede observar en la tabla 2.3.

Tipo	Cantidad (Uds.)	Factor amortización	Precio (€)	Coste total (€)
Ordenador	1	5/72	1 000,00	69,44
GPU	1	5/72	780,00	54,17
Subtotal				123,61

Tabla 2.3. Presupuesto hardware.

3. PRESUPUESTO TOTAL

3.1. PRESUPUESTO DE EJECUCIÓN MATERIAL

Para calcular este presupuesto se realiza la suma de cada uno de los presupuestos parciales, los costes totales los podemos ver en la tabla 3.1.

Unidad de obra	Coste total (€)
Mano de obra	8 176,00
Software	21,83
Hardware	123,61
Total	8 321,44

Tabla 3.1. Presupuesto de ejecución material.

3.2. PRESUPUESTO DE EJECUCIÓN POR CONTRATA

Finalmente, para hacer el cálculo del presupuesto por contrata hay que tener en cuenta los impuestos establecidos y añadirlos a los costes ya calculados. Los resultados finales se pueden ver en la tabla 3.2.

Tipo	Coste total (€)
Presupuesto ejecución material	8 327,59
Beneficio industrial (6%)	499,66
Gastos generales (13%)	1 082,59
Total antes de impuestos	9 909,83
IVA	2 081,06
Presupuesto ejecución por contrata	11 990,13

Tabla 3.2. Presupuesto ejecución por contrata.

Finalmente teniendo en cuenta los impuestos, gastos generales y el beneficio industrial el coste del proyecto asciende a once mil novecientos noventa euros con trece céntimos.