

## TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

**Métodos de preprocessamiento y  
segmentación en imágenes de  
ecografía transfontanelar aplicados  
al cálculo del volumen cerebral  
total en recién nacidos**

AUTOR: ÁLVARO DÁVILA LUNA

Cádiz, septiembre 2022

## TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

DIRECTOR: JOAQUÍN PIZARRO JUNQUERA  
AUTOR: ÁLVARO DÁVILA LUNA

Puerto Real, septiembre de 2022



# **Declaración personal de autoría**

---

Álvaro Dávila Luna con DNI 77496764-N, estudiante del Grado en Ingeniería Informática en la Escuela Superior de Ingeniería de la Universidad de Cádiz, como autor de este documento académico titulado Técnicas de preprocesamiento y segmentación de ecografías 3D aplicadas al calculo del volumen cerebral total en neonatos y presentado como Trabajo Final de Grado.

## **DECLARO QUE**

Es un trabajo original, que no copio ni utilizo parte de obra alguna sin mencionar de forma clara y precisa su origen tanto en el cuerpo del texto como en su bibliografía y que no empleo datos de terceros sin la debida autorización, de acuerdo con la legislación vigente. Asimismo, declaro que soy plenamente consciente de que no respetar esta obligación podrá implicar la aplicación de sanciones académicas, sin perjuicio de otras actuaciones que pudieran iniciarse. En Puerto Real, a septiembre de 2022

Fdo: Álvaro Dávila Luna



# Agradecimientos

---

*Quiero dar las gracias a mi tutor de TFG, Joaquín. Por su gran implicación y colaboración durante el desarrollo de este proyecto. Ha demostrado ser un gran profesional, pero además una gran persona.*

*También, a los compañeros con los que he compartido todos estos años de carrera. Con gusto puedo decir que en el grado de Ingeniería Informática de la Universidad de Cádiz existe uno de los mejores ambientes de colaboración y compañerismo que he vivido en toda mi etapa estudiantil. En especial a mis compañeros Fran, Antonio, Pablo, Sergio y David. Con los que he compartido muchísimos buenos momentos y, además, me han ayudado a desarrollarme como ingeniero informático con su ayuda y consejos.*

*A mi familia, por hacer posible que haya estudiado este grado y pueda desarrollar mi vocación.*

*A mi novia, Lucía. Por estar a mi lado apoyándome y animándome a seguir cuando no encontraba motivación. Por estar día a día acompañándome. Por hacer mejor mi vida desde el día que la conocí.*



# Resumen

---

Se realiza un estudio de distintas técnicas de segmentación aplicadas a imágenes biomédicas de ecografías 3D de volúmenes cerebrales de neonatos. Después, se muestra la aplicación de estas técnicas para el desarrollo de un software que a partir de una segmentación automática, permita al usuario retocar la segmentación, para la obtención del volumen cerebral total a partir de las ecografías 3D. Se utiliza este software para la creación de una base de datos, para el posterior entrenamiento de distintos tipos de redes neuronales convolucionales para conseguir segmentaciones automáticas y de calidad. Finalmente, se realiza un estudio de los resultados obtenidos y se discute si se ha desarrollado un método fiable para la segmentación de volúmenes cerebrales.

**Palabras clave:** Deep Learning, Ecografía 3D, Neurosonografía Neonatal, Segmentación del volumen cerebral.



# Índice general

---

Índice de figuras	ix
Índice de tablas	xii
<b>I Prolegómeno</b>	<b>1</b>
<b>1. Introducción</b>	<b>2</b>
1.1. Motivación . . . . .	2
1.2. Alcance y Objetivos . . . . .	3
1.3. Glosario de términos . . . . .	4
1.4. Organización del documento . . . . .	4
<b>2. Antecedentes</b>	<b>7</b>
2.1. Contexto . . . . .	7
2.1.1. Dificultades . . . . .	7
2.2. Estado del arte . . . . .	10
<b>3. Plan de gestión de proyecto</b>	<b>13</b>
3.1. Metodología de desarrollo . . . . .	13
3.2. Tecnologías . . . . .	14
3.3. Planificación temporal del proyecto . . . . .	15
3.4. Medios utilizados en el proyecto . . . . .	17
3.5. Costes . . . . .	17
3.6. Riesgos . . . . .	17
<b>II Desarrollo</b>	<b>19</b>
<b>4. Métodos Tradicionales de Segmentación</b>	<b>21</b>
4.1. Preprocesamiento . . . . .	21
4.1.1. Filtrado Gaussiano . . . . .	21
4.1.2. Filtro de Weiner . . . . .	21
4.1.3. Filtro de la mediana . . . . .	22
4.1.4. Filtros de detección de bordes: Sobel y Canny . . . . .	22
4.1.5. Filtros de detección de crestas: filtro de Frangi . . . . .	23
4.2. Segmentación . . . . .	23
4.2.1. K-means clústering . . . . .	23
4.2.2. Superpixels . . . . .	26
4.2.3. Random Walks . . . . .	26
<b>5. Deep Learning</b>	<b>31</b>
5.1. Operación de convolución . . . . .	31

5.2.	Redes neuronales convolucionales(CNN) . . . . .	31
5.2.1.	Arquitectura general . . . . .	32
5.2.2.	Técnicas de aumento de datos . . . . .	33
5.3.	Arquitectura UNET . . . . .	34
5.4.	Arquitectura SegNet . . . . .	34
5.5.	Entrenamiento de las redes UNET y SegNet . . . . .	36
5.5.1.	Software de segmentación semi-automática . . . . .	36
5.5.2.	Generar la base de datos . . . . .	42
5.5.3.	Entrenamiento . . . . .	44
5.5.4.	Resultados del entrenamiento . . . . .	45
5.5.5.	Mejora de la red SegNet . . . . .	49
5.5.6.	Cálculo del VCT . . . . .	49
5.5.7.	Más métodos para el cálculo del VCT . . . . .	50
<b>III</b>	<b>Epílogo</b>	<b>51</b>
<b>6.</b>	<b>Resultados y discusión</b>	<b>52</b>
6.1.	Comparativa de los distintos métodos para el cálculo del VCT . . . . .	52
<b>7.</b>	<b>Conclusiones</b>	<b>57</b>
7.1.	Objetivos alcanzados . . . . .	57
7.2.	Lecciones aprendidas . . . . .	57
7.3.	Trabajo futuro . . . . .	58
<b>Bibliografía</b>		<b>61</b>
<b>IV</b>	<b>Anexos</b>	<b>65</b>
<b>A.</b>	<b>Tablas de Resultado</b>	<b>66</b>
A.1.	Tabla comparativa de los distintos métodos para el cálculo del VCT . .	66

# Índice de figuras

---

2.1. Un médico realizando una ecografía transfontanelar a un recién nacido. Fuente: Imagen extraída de [1]	7
2.2. Figuras de ecografías de distinta calidad.	8
2.3. Ecografía del plano sagital en la que falta la parte superior derecha del cráneo.	9
2.4. Software 4D View.	10
3.1. Diagrama de Gantt.	16
4.1. A la izquierda la imagen original a la derecha la imagen filtrada.	21
4.2. A la izquierda la imagen original a la derecha la imagen filtrada.	22
4.3. A la izquierda la imagen original a la derecha la imagen filtrada.	22
4.4. Filtros de detección de bordes.	23
4.5. A la izquierda la imagen original a la derecha la imagen filtrada.	23
4.6. A la izquierda la imagen original con un filtro de la mediana a la derecha el resultado de k-means.	24
4.7. Resultado de aplicar Frangi y k-means al volumen completo.	25
4.8. A la izquierda la imagen original, a la derecha la imagen resultante de aplicar superpixel.	26
4.9. Representación de la imagen como un circuito. Fuente: Imagen extraída de [2].	28
4.10. Resultado de la segmentación utilizando Random Walks.	29
5.1. Ejemplo de convolución. Fuente: Imagen extraída de [3].	32
5.2. Red neuronal convolucional. Fuente: Imagen extraída de [4].	33
5.3. Arquitectura UNET. Cada cuadro azul corresponde a un mapa de características multicanal. El número de canales se especifica en la parte superior de la caja. El tamaño x-y se especifica en el borde inferior izquierdo de la caja. Las cajas blancas representan mapas de características copiados. Las flechas denotan las diferentes operaciones. Fuente: Imagen extraída de [5].	35
5.4. Arquitectura SegNet. No tiene capas completamente conectadas. El decoder realiza el upsampling utilizando los índices transferidos del encoder. Realiza la convolución para crear el mapa de características. Al final tiene un soft-max clasifier para la clasificación de los píxeles. Fuente: Imagen extraída de [6].	36
5.5. Software de segmentación con retoque manual. En amarillo veremos el contorno de la zona segmentada, a la izquierda en el cuadro de vista podremos seleccionar que plano queremos visualizar. La barra Slice permite seleccionar el número del plano que queremos ver y cuenta con botones para movernos por el volumen. Para modificar la segmentación contamos con las opciones de perfilar, aumentar, recortar, disminuir, dibujar una nueva elipse y suavizar	37

5.6.	Tercer y cuarto paso del algoritmo de segmentación utilizando Random Walks. . . . .	38
5.7.	Ejemplo de elipsoides iniciales mal situadas debido al alto ruido de la imagen, utilizando el algoritmo de segmentación. . . . .	40
5.8.	Ejemplo de error en una segmentación de una imagen del plano axial en uno de los extremos, debido a la propagación del error de Random Walks. Parte del cráneo ha sido clasificada como “cerebro” . . . . .	40
5.9.	Ejemplo de sobreestimación del cerebro por el algoritmo de segmentacion. . . . .	41
5.10.	Ejemplo de etiquetado de las imágenes segmentadas. (a) Imagen extraída del plano axial. (b) Etiqueta correspondiente con 3 clases. En azul el “cerebro”, en negro el “fondo”, en blanco ‘‘no-cerebro’’. . . . .	43
5.11.	Distribución (%) de píxeles en las tres clases. . . . .	46
5.12.	Métricas globales de ambas redes. . . . .	46
5.13.	Métricas por clase de ambas redes. . . . .	47
5.14.	Matriz de confusion de UNET. . . . .	47
5.15.	Matriz de confusion de SegNet. . . . .	47
5.16.	Comparación de las segmentaciones realizadas por ambas redes en tres imágenes distintas de un mismo volumen. En amarillo la segmentación Unet, en rojo la segmentación Segnet . . . . .	47
5.17.	Comparación de las segmentaciones realizadas por ambas redes en tres imágenes distintas de un mismo volumen. En amarillo la segmentación Unet, en rojo la segmentación Segnet . . . . .	48
6.1.	Ejemplo del método de la segmentación ajustando elipses en cada plano a partir de la segmentación de SegNet y calculando un elipsoide. Vemos como el elipsoide consigue llenar la parte del cerebro que falta. . . . .	53
6.2.	Tendencia del crecimiento del cerebro de un bebé, utilizando el método 7.E123_elipsoide. En el eje Y $cm^3$ . En el eje X la fecha en la que se realizó la ecografía. En azul oscuro la medición manual. En naranja el resultado de la segmentación utilizando una elipsoide a partir de las elipses estimadas en cada plano. En azul claro el error absoluto. . . . .	54
6.3.	Comparación del método 7.E123_elipsoide con las mediciones obtenidas con el método VOCAL. En el eje Y $cm^3$ . En el eje X los índices de cada volumen. En azul oscuro la medición manual. En naranja el resultado de la segmentación utilizando una elipsoide a partir de las elipses estimadas en cada plano. En azul claro el error absoluto. . . . .	54
6.4.	Comparación del método 6.V123_elipses con las mediciones obtenidas con el método VOCAL. En el eje Y $cm^3$ . En el eje X los índices de cada volumen. En azul oscuro la medición manual. En azul claro el resultado de la segmentación elipses estimadas en cada plano. En rosa el error absoluto. . . . .	55
6.5.	Comparación del método 5.E13_elipsoide con las mediciones obtenidas con el método VOCAL. En el eje Y $cm^3$ . En el eje X los índices de cada volumen. En azul oscuro la medición manual en cm <sup>3</sup> . En rojo el resultado de la segmentación utilizando una elipsoide a partir de las elipses estimadas en el plano axial y coronal. En verde el error absoluto. . . . .	55

# Índice de tablas

---

3.1. Costes de recursos software y humanos . . . . .	17
3.2. Tabla de riesgos . . . . .	18



# **Parte I**

## **Prolegómeno**

# 1. Introducción

---

## 1.1. Motivación

Este TFG forma parte de un proyecto más grande llamado *Automatización de la Neurosonografía Neonatal (AUTONENE)*. La investigadora principal de este proyecto es la doctora Isabel Benavente Fernández, Médico Especialista en Pediatría del Hospital Puerta del Mar.

Más de la mitad de los niños prematuros de muy bajo peso tienen el riesgo de padecer alteraciones cognitivas, motoras, sensoriales o conductuales [7][8]. Esto conlleva una gran carga para el niño y su familia, además de económica, estimándose que el coste del cuidado de la vida de un recién nacido prematuro con discapacidad es de unos 900 000 euros [9].

Actualmente, el objetivo principal es conseguir reducir el tiempo de diagnóstico de estas enfermedades, ya que se diagnostican entre los 2 y 5 años de edad.

Las principales técnicas de neuroimagen utilizadas son la resonancia magnética cerebral y la ecografía. Ambas tienen ventajas y desventajas. La resonancia magnética cerebral es actualmente la técnica "gold standard". Es la que proporciona imágenes de mejor calidad y las mejores medidas lineales y volumétricas. Cada resonancia magnética tiene un coste de 400€, requiere la inmovilización del niño por lo que hay que sedarlo y la hace más insegura, no todos los hospitales tienen acceso a esta tecnología. La ecografía 2D está muy limitada ya que es difícil obtener imágenes de calidad, además es imposible realizar estudios volumétricos. La ecografía 3D supone un salto tecnológico muy grande respecto de la 2D. Ya que permite la adquisición volumétrica cerebral. Una vez realizada la ecografía permite visualizar los tres planos del cerebro por separado y realizar medidas lineales y volumétricas.

En neonatología, la ecografía craneal se utiliza ampliamente en la evaluación del estado del cerebro de los recién nacidos, y más específicamente en los bebés prematuros. Este uso generalizado es debido a sus diversas ventajas respecto a la resonancia magnética como el bajo coste, la naturaleza no invasiva, radiación no ionizante, la comodidad del operador, la portabilidad y la accesibilidad.[10][11][12][13]

Las ecografías craneales solo pueden ser realizadas durante el periodo en el que la fontanela anterior aún está abierta, normalmente hasta los 18 meses de edad, pero normalmente suele usarse durante los primeros 5-6 meses, que es el periodo en el que pueden obtenerse las mejores ecografías. Después de esa edad, debido engrosamiento de las membranas del cerebro y el cierre de la fontanela, las estructuras cerebrales comenzarán a ser menos visibles[14][15].

La práctica clínica actual se basa en la evaluación visual y en algunas mediciones lineales 2D. Muchas investigaciones conducen a que sería interesante el uso de un mejor análisis cuantitativo sobre la evaluación visual y prueban la utilidad de otras medidas respecto de las lineales 2D, como las medidas volumétricas. Esto tiene el potencial de

mejorar el diagnóstico y el pronóstico de los trastornos de neurodesarrollo en los recién nacidos prematuros. Sin embargo, todavía no se ha podido adoptar en las clínicas porque requiere segmentar manualmente las estructuras anatómicas de interés en el cerebro, lo que lleva mucho tiempo y es propenso a la variabilidad inter e intraobservador [16].

El desarrollo de métodos automáticos para el análisis de imágenes de ecografía craneal puede aliviar estos retos haciendo que dicho análisis sea más objetivo, preciso y rápido. Los métodos automáticos incluyen la segmentación como paso previo importante para la extracción de los parámetros clínicos que los neonatólogos necesitan para realizar una evaluación y un diagnóstico basados en mediciones cuantitativas [17].

## 1.2. Alcance y Objetivos

El objetivo general del proyecto AUTONENE es desarrollar un software que integre distintas tecnologías, y sea accesible desde la web, para que el usuario, sin la necesidad de tener experiencia, pueda subir a la web una ecografía 3D cerebral de un neonato y que pueda obtener información diagnóstica y pronóstica de la misma.

AUTONENE está dividido en 3 Objetivos específicos. Objetivo(1). mejorar la adquisición de las ecografías 3D mediante un método/dispositivo que guíe al personal no experto en la adquisición de la ecografía. Objetivo(2). desarrollar algoritmos basados en el procesamiento de imágenes e Inteligencia Artificial que permitan obtener medidas de las diferentes estructuras cerebrales, además de una herramienta de ayuda que alerte de si hay algún problema detectado. Objetivo(3). Integrar estos algoritmos en una plataforma web que permita subir las ecografías y obtener las respuestas.

El objetivo principal de este TFG se encuentra dentro del ámbito del Objetivo(2). Se trata de obtener el volumen cerebral total (VCT) a partir de la ecografía 3D.

Objetivos específicos:

- **Objetivo 1:** Realización de un estudio de algoritmos de preprocesamiento y segmentación de imágenes para ver como se comportan con las imágenes de nuestro problema.
- **Objetivo 2:** Desarrollo de un software que proponga una segmentación del cerebro y permita al usuario retocarla.
- **Objetivo 3:** Utilizando el software anterior crear una base de datos con volúmenes bien segmentados y utilizarla para el entrenamiento de redes neuronales convolucionales.
- **Objetivo 4:** Realizar un estudio comparando las distintas técnicas y seleccionar la mejor.

## 1.3. Glosario de términos

- **DL:** Deep Learning.
- **CNN:** Convolutional Neural Network.
- **VCT:** Volumen Cerebral Total.
- **Slice:** Llamaremos slice a una imagen 2D de uno de los planos del volumen.
- **Registro de imagen:** Es el proceso de calcular transformaciones espaciales para alinear un conjunto de imágenes con una imagen de referencia.
- **Ruido sal y pieminta:** Ruido que cubre toda la imagen con píxeles blancos y negros.
- **Unppaded convolution:** Cuando se realiza la convolución se perderán parte de los bordes, lo que hará que se reduzca el tamaño de la imagen de salida.
- **Capa ReLu:** Capa que aplica la función de activación lineal rectificada.
- **Upsampling:** Aumento de la resolución espacial.
- **Downsampling:** Disminución de la resolución espacial.
- **Entrenamiento end-to-end:** Se refiere al entrenamiento de un sistema complejo representado por una red neuronal que se encargará de resolver el problema completo. Evitando los sistemas tipo pipeline que se utilizaban tradicionalmente con algoritmos tradicionales.
- **Batch Normalization:** Es una técnica que estandariza las entradas a una capa para cada mini-batch.
- **Stride:** Durante la convolución stride se refiere al tamaño de los saltos que realiza el kernel que realiza la convolución.
- **Epoch:** En el entrenamiento de la red neuronal un epoch significa una pasada por todo el conjunto de entrenamiento.
- **Batch:** En el entrenamiento de la red neuronal un Batch significa que se utilizan todos los datos para calcular el gradiente durante una iteración.
- **Mini-Batch:** En el entrenamiento de la red neuronal un Mini-Batch significa que se utiliza un subconjunto de todos los datos para calcular el gradiente durante una iteración.

## 1.4. Organización del documento

El presente proyecto está dividido en 3 grandes partes, que a su vez se dividen en capítulos, secciones y subsecciones.

- *Parte I: Prolegómenos*

- **Capítulo 1. Introducción:** En este capítulo veremos la motivación para la realización del proyecto, los objetivos propuestos y el alcance del mismo. También incluye un glosario de términos que será de ayuda para la lectura del documento.
- **Capítulo 2. Antecedentes:** Veremos el contexto del proyecto y las dificultades que encontramos en la segmentación de las imágenes de ecografía. Realizamos un análisis del estado del arte en este tipo de problemas.
- **Capítulo 3. Plan de gestión de proyecto:** Metodología de desarrollo utilizada para la realización del proyecto, tecnologías utilizadas, diagrama de gantt, medios utilizados, costes y plan de riesgos.

■ *Parte II: Desarrollo*

- **Capítulo 4. Métodos Tradicionales de Segmentación:** En este capítulo se realiza un estudio de algunos filtros y métodos tradicionales de segmentación aplicados a la segmentación de imágenes de ecografía transfontanelar en 2D y 3D.
- **Capítulo 5. Deep Learning:** En este capítulo veremos una introducción a las redes neuronales convolucionales. Estudiaremos las arquitecturas UNET y SegNet. Veremos como realizar el proceso de entrenamiento completo de estas redes. En primer lugar, veremos el software desarrollado que nos permite realizar la segmentación de los volúmenes que utilizaremos en el entrenamiento. Describiremos el proceso llevado a cabo para el registro de las imágenes y generación de nuestra base de datos. A continuación, entrenaremos una red UNET y una red SegNet preentrenada a partir de los pesos de VGG16, y veremos los resultados. Por último, veremos distintos métodos para calcular el volumen cerebral total a partir de las segmentaciones de las redes neuronales convolucionales.

■ *Parte III: Epílogo*

- **Capítulo 6. Resultados y discusión:** Realizaremos un estudio comparativo de los distintos métodos de cálculo del volumen cerebral total. Discutiremos cuál es el mejor de ellos y si podría utilizarse como un método fiable de medida de volúmenes cerebrales.
- **Capítulo 7. Conclusiones:** Comprobaremos si hemos alcanzado todos los objetivos propuestos, lecciones aprendidas y posibles mejoras para trabajo futuro.
- **Bibliografía:** Bibliografía utilizada durante el desarrollo del proyecto.

■ *Anexos*

- **Anexo A:** Tablas de resultado



## 2. Antecedentes

---

En este capítulo se detallan los datos que tenemos disponibles y las herramientas con las que contamos para la resolución del problema. También se describen algunos estudios realizados con ecografías 3D cerebrales y las tecnologías que utilizan.

### 2.1. Contexto

El proyecto se desarrolla en colaboración con la Unidad de Neonatología del Hospital Universitario Puerta del Mar de Cádiz. Ellos nos proporcionaran los datos necesarios.

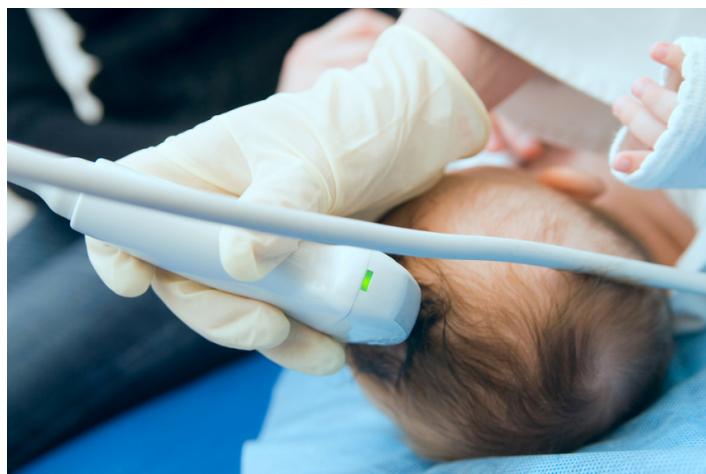


Figura 2.1: Un médico realizando una ecografía transfontanelar a un recién nacido.  
Fuente: Imagen extraída de [1]

Para el desarrollo del proyecto nos han proporcionado una base de datos formada por 384 volúmenes. Esta base de datos cuenta con distintas mediciones realizadas sobre los volúmenes utilizando el método VOCAL, entre ellas el volumen cerebral total. Serán las medidas que tomaremos de referencia para el desarrollo de nuestros estudios.

Son ecografías 3D realizadas entre 2018 y 2019. Por distintos médicos y ecógrafos. Esto, sumado al gran numero de volúmenes, hace que sea una muestra suficientemente representativa para nuestro estudio.

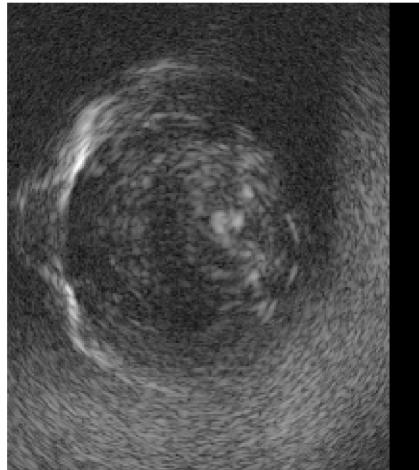
#### 2.1.1. Dificultades

La ecografía 3D, es un gran avance respecto la 2D. Permite visualizar los 3 planos del cerebro y obtener mediciones volumétricas y lineales. Es una tecnología mucho más barata que la resonancia magnética, puede realizarse por personal no especializado y es más segura para el niño. Sin embargo, no es perfecta.

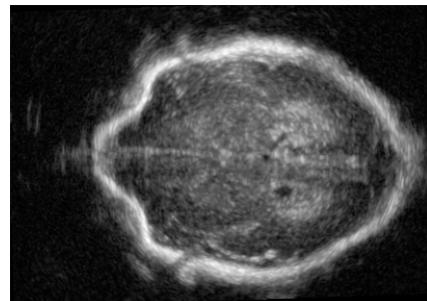
Las imágenes suelen ser ruidosas y con poca resolución espacial y de contraste. La calidad del resultado depende de la habilidad de la persona que realiza la ecografía. Estos fallos se pueden observar en la figura 2.2. Ambas imágenes son de la mitad del cerebro en el plano axial.

La imagen 2.2a tiene mucha cantidad de ruido, bajo contraste y además el cráneo esta cortado por una parte. Extraer el volumen de esta ecografía será más complicado.

La imagen 2.2b tiene un alto contraste y muy bajo ruido, a simple vista se distinguen bien el cráneo, el cerebro y el fondo.



(a) Ruido y partes cortadas.



(b) Buena resolución y contraste.

Figura 2.2: Figuras de ecografías de distinta calidad.

Otro problema que encontramos, independientemente del ruido, contraste y resolución es que no suele verse el volumen cerebral completo en las imágenes. Si viajamos por el plano sagital de la ecografía, en la figura 2.3 podemos ver que la parte superior derecha del plano está cortada y falta parte del cerebro.

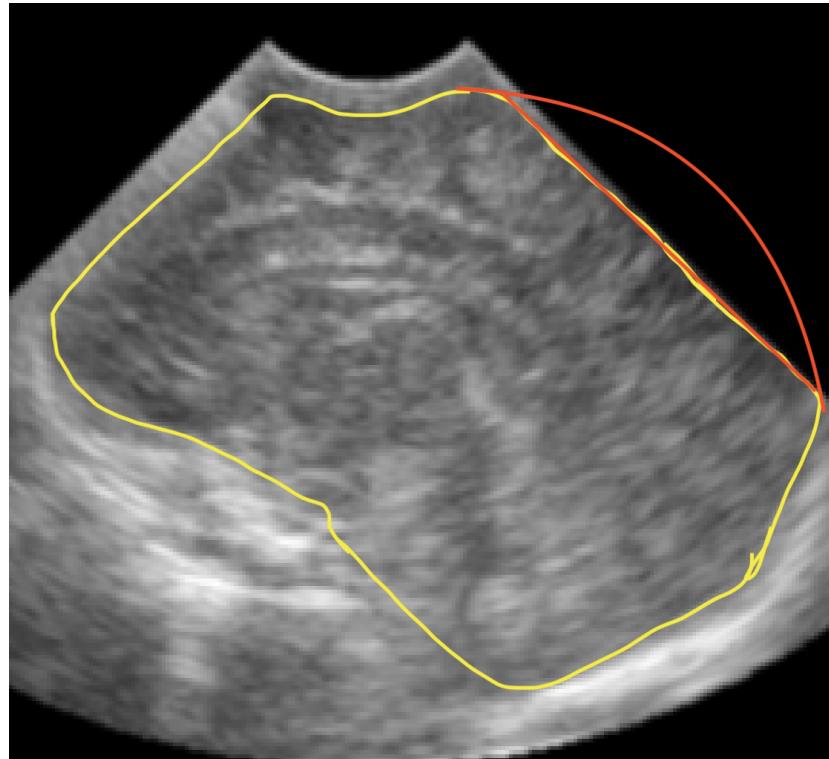


Figura 2.3: Ecografía del plano sagital en la que falta la parte superior derecha del cráneo.

## 2.2. Estado del arte

El software actual de referencia de análisis de ecografías es *4D view* de *General Electric Healthcare* [18]. Para obtener mediciones volumétricas utilizan el método VOCAL. Este método utiliza delineación manual de contornos. Es necesario que una persona con experiencia analice distintos planos y delimita las zonas de interés de forma manual, lo que le podría llevar entre 2 y 10 minutos.



Figura 2.4: Software 4D View.

Si buscamos bibliografía sobre métodos de automatización del tratamiento de ecografías 3D, podemos encontrar algunos resultados muy interesantes.

En Boucher et al.[19] Utilizan el cálculo de un elipsoide para estimar el volumen cerebral. El cerebro no tiene una forma de elipsoide perfecta, para corregir este error multiplican el volumen del elipsoide estimado por 0,95. Para estimar la elipsoide utilizan las medidas de dos elipses cuadradas con el cráneo, una en el plano sagital y otra en el coronal. En las ecografías 3D el volumen normalmente no se ve completo, para paliar este defecto toman puntos de referencia utilizando técnicas de registro de imagen.

En Milletari et al.[20] Hacen una comparación de 6 arquitecturas de redes neuronales convolucionales para la extracción de distintas estructuras cerebrales. Tanto en ecografías 3D como en resonancias magnéticas.

También es muy interesante el TFM de Reiner[21]. El objetivo era automatizar el estudio de los surcos del cerebro. Lo interesante para nuestro problema es el banco de filtros que utilizan. Están basados en la detección de bordes y crestas para resaltar los surcos y el cráneo. Para el resultado de bordes utilizan el filtro de la diferencia gaussiana y el Laplaciano de la distribución gaussiana. La función de los filtros de detección de crestas es detectar estructuras lineales que son más gruesas que una línea, lo que las hace difíciles de detectar con filtros de detectado de bordes. En las ecografías 3D los surcos y el cráneo podrían considerarse crestas. Además, también realizan un estudio de distintos algoritmos de clasificación de píxeles para la detección de surcos cerebrales.

De la revisión sobre la segmentación automática craneal de ecografías de neonatos usando DL, Khaled et al. [22], podemos extraer datos muy interesantes. Primero,

realizan una revisión de los métodos no basados en DL para la extracción de alguna estructura cerebral a partir de ultrasonidos. La mayoría de los métodos de segmentación utilizados se basan en técnicas de segmentación de imágenes clásicas. Primero se realiza un preprocessamiento de las imágenes aplicando filtros para eliminar el ruido de la imagen. Luego, se realiza la segmentación utilizando umbralización de intensidad o filtros de detección de bordes. Finalmente utilizan operaciones morfológicas para retopear la segmentación. Además, resaltan que la mayoría de métodos son de segmentación semiautomática, lo que significa que es necesaria la interacción del usuario.

Después pasan a realizar una revisión basada en los métodos que utilizan DL. La mayoría de métodos de segmentación que utilizan DL se basan en imágenes 2D extraídas de volúmenes 3D. El 3D DL todavía es una tarea muy complicada y con muchas limitaciones. Entrenar este tipo de redes con grandes volúmenes es muy caro computacionalmente para aplicaciones clínicas reales. Una red con volúmenes 3D requiere de más muestras de entrenamiento, ya que cuentan con más parámetros, y en este tipo de problemas no suele ser sencillo obtener más muestras.

También realizan una revisión de distintas arquitecturas de redes neuronales aplicadas a la segmentación de los ventrículos cerebrales, y resaltan que sería interesante evaluar esas medidas en relación con el VCT pero no hay estudios que realicen este tipo de mediciones.



# 3. Plan de gestión de proyecto

---

En esta sección se describen la metodología seguida para la gestión del proyecto, las tecnologías utilizadas en el desarrollo y la planificación temporal del mismo.

## 3.1. Metodología de desarrollo

La metodología de desarrollo es el conjunto de técnicas y métodos utilizados para organizar un equipo de trabajo. Aunque este proyecto ha sido desarrollado por un estudiante, en colaboración con su tutor, es interesante seleccionar una metodología de desarrollo y aplicarla durante todo el proceso.

Una metodología bien definida, junto con una buena planificación, nos ayudará a optimizar el tiempo de desarrollo y conseguir buenos resultados cumpliendo con todos los requisitos.

Las metodologías tradicionales de desarrollo de software se basan en una estructura lineal, en la que para que una etapa suceda debe de haberse completado la anterior. Estas metodologías son muy poco flexibles por lo que cada vez se usan menos, ya que en la actualidad se producen cambios muy rápidos e imprevistos.

Las metodologías ágiles han tomado una gran importancia en los últimos años, por su gran capacidad de adaptación a los cambios. Se basan en una estrategia incremental, en la que se van fijando objetivos funcionales y se van agregando al producto final. Así, en cada iteración lo que se obtiene es un producto completo al que se le van agregando funcionalidades en cada etapa de desarrollo.

Esto no significa que unas metodologías sean correctas y otras incorrectas, habrá metodologías tradicionales que se sigan siendo válidas para algunos proyectos y empresas.

Para el desarrollo de este proyecto se ha elegido la metodología de desarrollo SCRUM [23] adaptada a nuestro TFG, ya que SCRUM es una metodología pensada para la organización de equipos de distintos tamaños.

SCRUM se define como un marco de trabajo, no como una metodología. Trata de ser simple y, está compuesto por unas reglas básicas. Esto hace que sea fácil adaptarlo a cualquier proyecto.

SCRUM se basa en 5 eventos[24]:

- **Sprint:** Son el corazón de SCRUM. Tienen una duración de un mes o menos. Durante este tiempo suceden el resto de eventos que veremos a continuación y el desarrollo del proyecto.
- **Sprint Planning:** Inicia el *Sprint* delimitando el trabajo a realizar en el mismo.

- **Dayli Scrum:** Se trata de una reunión diaria de 15 minutos en la que se pone en común el trabajo realizado, posibles problemas y se ajusta el trabajo futuro
- **Sprint Review:** Se realiza al final del *Sprint*, se analizan los resultados y se determinan futuras adaptaciones.
- **Sprint Retrospective:** El objetivo es planear formas de mejorar la calidad y la efectividad.

Y en 3 artefactos[24]:

- **Product Backlog:** Es una lista ordenada y emergente que contiene las tareas que son necesarias realizar para mejorar el producto.
- **Sprint Backlog:** Es un subconjunto de los items del Product Backlog, sumado a un plan para llevar a cabo el *Increment*.
- **Increments:** Es un paso concreto hacia el objetivo final. Cada incremento se une al anterior para que cada vez se vaya añadiendo más valor al producto. Todos los incrementos deben de trabajar en conjunto, formando un producto funcional.

SCRUM se ha adaptado al desarrollo de este proyecto, simplificándolo. En una reunión inicial con el tutor se realizó el Product Backlog inicial y el primer Sprint Planning. A partir de ahí se fijaban Sprints de 2 o 3 semanas de duración que finalizaban en una reunión con el tutor en la que, en el mismo día, se hacían el Sprint Review, Sprint Retrospective y Sprint Planning. Cada día se realizaba un Dayli Scrum que consistía en revisar el trabajo del día anterior y fijar objetivos diarios, actualizando el Sprint Backlog.

## 3.2. Tecnologías

El lenguaje de programación que utilizaremos será MATLAB[25]. Matlab es un lenguaje de programación pensado para trabajar directamente con matrices y conjuntos, pero es mucho más que eso.

Cuenta con un entorno de desarrollo que proporciona las herramientas para realizar análisis de datos, visualización de gráficos de forma sencilla, creación de scripts y funciones, creación de aplicaciones, desarrollo de interfaces gráficas muy sencillas y computación paralela para acelerar los cálculos.

Algo muy interesante de Matlab son las MATLAB Toolboxes. Son un conjunto de funciones creadas específicamente para Matlab especializadas en un campo en concreto. Están testadas y son muy robustas. Además están adecuadamente documentadas[26] con información y ejemplos, lo que facilita mucho la tarea del programador. El inconveniente es que no están incluidas en el paquete base, deben adquirirse aparte.

En este proyecto utilizaremos:

- **Statistics and Machine Learning Toolbox:** Para el análisis estadístico.
- **Deep Learning Toolbox:** Para el desarrollo de algoritmos de segmentación de aprendizaje supervisado.

- **Image Processing Toolbox:** Para el procesado de las imágenes de ecografías.
- **Computer Vision System Toolbox:** Lo usaremos junto con Deep Learning toolbox para el tratamiento de los datos.

### 3.3. Planificación temporal del proyecto

El proyecto se ha desarrollado en un total de 9 Sprints.

- **Sprint 1 - Formación y estudio de los métodos de segmentación clásicos:** Este sprint de 4 semanas de dedicará a la familiarización con las ecografías. Consistirá en aprender los conocimientos básicos sobre medicina para entender las ecografías y el origen del problema. También se comenzará a trabajar con las ecografías utilizando Matlab, para familiarizarse con el entorno. Después, se realizará un estudio de los distintos métodos de segmentación clásicos aplicados a las ecografías 2D y 3D.
- **Sprint 2 - Implementación de un software para segmentar volúmenes con retoque manual:** En este Sprint de 2 semanas se implementará un software que presente una segmentación aproximada del volumen y permita retocarla y guardarla, para más tarde ser utilizado para la creación de una base de datos de volúmenes segmentados.
- **Sprint 3 - Creación de una base de datos con volúmenes segmentados usando el software:** En este Sprint de 3 semanas se segmentarán los volúmenes utilizando el software creado en el sprint anterior para disponer de una base de datos de volúmenes bien segmentados.
- **Sprint 4 - Investigación sobre redes neuronales convolucionales aplicadas a imágenes biomédicas:** En este Sprint de 2 semanas se realizará una investigación sobre redes neuronales convolucionales aplicadas para la segmentación.
- **Sprint 5: Entrenamiento de las redes neuronales:** En este sprint de 2 semanas se realizará un primer entrenamiento de redes neuronales convolucionales, desarrollando los Scripts necesarios.
- **Sprint 6 - Estudio del funcionamiento y análisis de resultados:** Sprint de 1 semana de duración en el que se realizará un análisis de los resultados.
- **Sprint 7 - Estudio de posibles mejoras al sistema del cálculo de volúmenes:** Este Sprint de 3 semanas se dedicará a intentar desarrollar algunas mejoras en los resultados, ya sea volviendo a entrenar alguna red o aplicando distintas formas para el cálculo de los volúmenes.
- **Sprint 8 - Redacción formal de la memoria en Latex:** Sprint de 4 semanas en el que se recogerán todos los datos obtenidos durante el desarrollo y se desarrollará una memoria en Latex.
- **Sprint 9 - Presentación para la defensa ante el tribunal:** Sprint de 12 días que se dedicará a preparar la presentación del proyecto ante el tribunal.

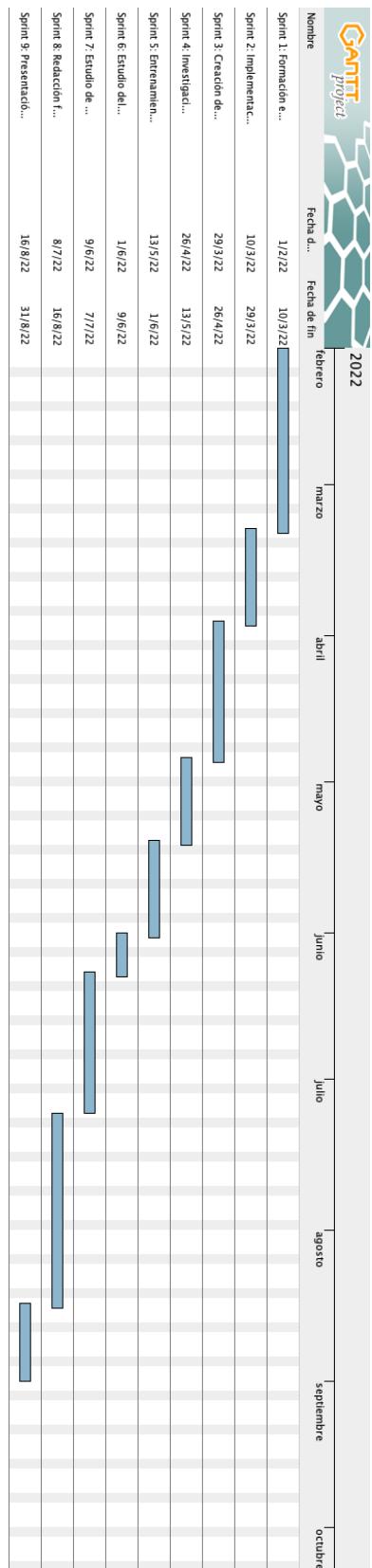


Figura 3.1: Diagrama de Gantt.

### **3.4. Medios utilizados en el proyecto**

La Unidad de Neonatología dispone de un ecógrafo con el que se han realizado las ecografías 3D que nos han proporcionado. Contamos con una base de datos con más de 300 volúmenes.

Para el desarrollo del software se ha utilizado un MacBook Pro 2017 con un i5 de doble nucleo a 2,3GHz y 16GB de RAM. Para tareas intensivas de computación se ha utilizado la estación de trabajo del laboratorio de Sistemas Inteligentes de Computación de la Escuela Superior de Ingeniería. Este equipo cuenta con un procesador Intel Xeon CPU E5-2620 v2 @2.1GHz, 32GB RAM y una tarjeta gráfica NVIDIA Tesla K20c y una NVIDIA Quadro K4000

### **3.5. Costes**

En costes no se incluyen los equipos utilizados, ya que estaban disponibles y no ha sido necesaria ninguna actualización del hardware. Un trabajo de final de grado de estas características se corresponde con 18 créditos ETC , según el sistema europeo un crédito se corresponde con 25 horas de trabajo lo que nos da un total de 450 horas de trabajo, que incluyen reuniones, investigación y desarrollo, además de la redacción de la memoria. Según Jobted.com[27] el salario medio de un ingeniero informático es de 36.500€/año en España, lo que resulta en 17,55€/hora.

Tabla 3.1

*Costes de recursos software y humanos*

Cantidad	Descripción	Coste	Importe Total
1	Anual Standard license MATLAB	840€	840€
1	Anual license Statistics and Machine Learning Toolbox	420€	420€
1	Anual license Deep Learning Toolbox	480€	480€
1	Anual license Image Processing Toolbox	420€	420€
1	Anual license Computer Vision System Toolbox	520€	520€
450	Horas Trabajadas	17,55€	7897,5€
Total			10 577,5€

### **3.6. Riesgos**

En esta sección veremos, resumidos en una tabla los posibles riesgos que podrían ocurrir durante el desarrollo del trabajo. Además, cada uno de para cada uno de ellos se evalúa la probabilidad de ocurrir, el impacto, formas de mitigación y su correspondiente plan de contingencia.

Tabla 3.2

*Tabla de riesgos*

Descripción	Probabilidad	Impacto	Mitigación	Plan de contingencia
Mala ejecución en la realización de la ecografía	30 %	Crítico	Educación y entrenamiento del personal sanitario no especializado	Revisar la ecografía y volverla a realizar si es necesario
Errores en el código	90 %	Crítico	Realizar pruebas software durante todo el desarrollo intentando minimizar los bugs en cada sprint	Realizar un mantenimiento adecuado del código
Falta de familiarización con el lenguaje y el material médico	90 %	Crítico	Realizar una formación inicial sobre ecografías cerebrales, aprendiendo a identificar los distintos planos y las distintas estructuras cerebrales	Ponerse en contacto con el equipo de médicos de neonatología del Hospital Puerta del Mar para consultar cualquier tipo de duda.
Dificultades en el desarrollo de la memoria	90 %	Crítico	Actualización diaria de toda la bibliografía consultada, creación de un diario con el trabajo que se ha realizado, preparar ejemplos durante el desarrollo para añadirlos a la memoria	Invertir más tiempo en la memoria y consultar TFGs en el archivo de la biblioteca de la Universidad de Cádiz[28]

# **Parte II**

## **Desarrollo**



# 4. Métodos Tradicionales de Segmentación

---

## 4.1. Preprocesamiento

El mayor problema que encontramos en las imágenes de ecografía es el ruido. Las interferencias granulares que encontramos en las ecografías son conocidas como *Speckles*[29]. La presencia de *Speckles* es debida a las interferencias que se producen cuando se superponen ondas. Son responsables de introducir en la imagen estructuras finas y falsas, enmascarando los límites del tejido real. En este apartado aplicaremos distintos filtros a las imágenes enfocados a disminuir el ruido y resaltar distintas estructuras cerebrales.

### 4.1.1. Filtrado Gaussiano

El filtro gaussiano es un filtro de convolución de suavizado que suele usarse para eliminar el ruido gaussiano. Utiliza la distribución gaussiana 2D para calcular la matriz de convolución. A mayor sigma, mayor será el suavizado.

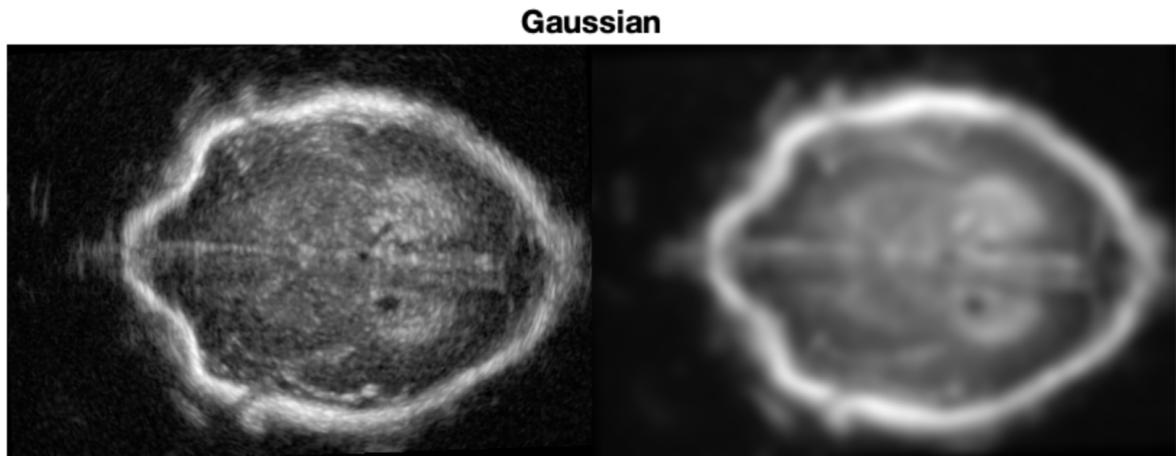


Figura 4.1: A la izquierda la imagen original a la derecha la imagen filtrada.

### 4.1.2. Filtro de Weiner

Es una estimación lineal de la imagen original. Es muy utilizado en el suavizado de imágenes.

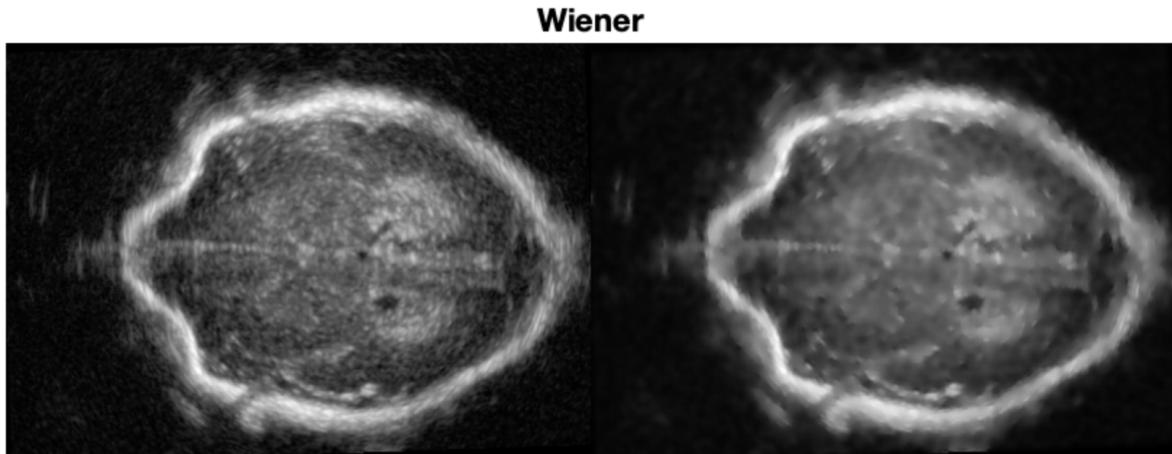


Figura 4.2: A la izquierda la imagen original a la derecha la imagen filtrada.

#### 4.1.3. Filtro de la mediana

Es una operación no lineal que suele usarse para reducir el ruido del tipo “sal y pimienta”. Es muy útil cuando se necesita reducir el ruido manteniendo los bordes. El filtro reemplaza cada píxel por el valor de la mediana de sus vecinos.

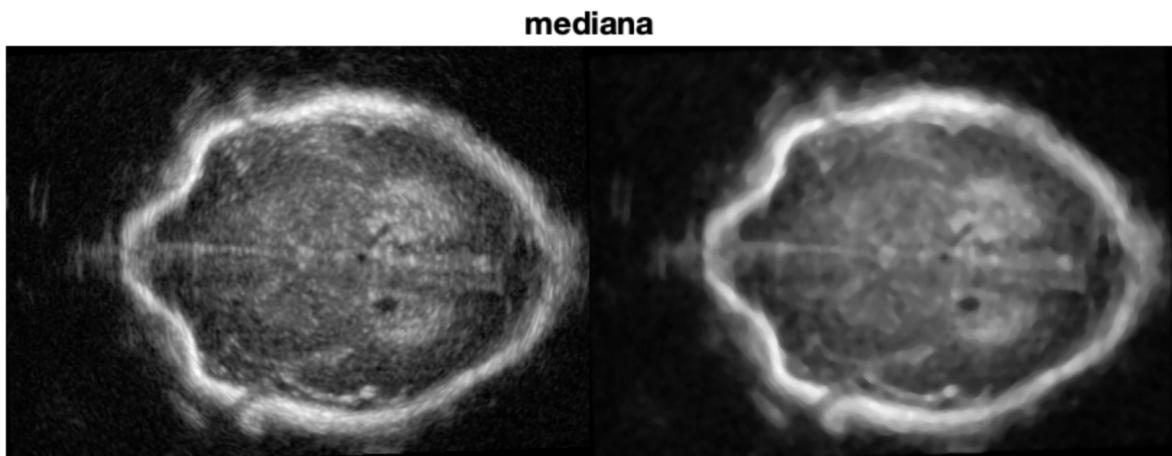
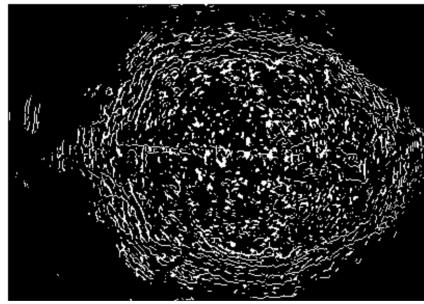


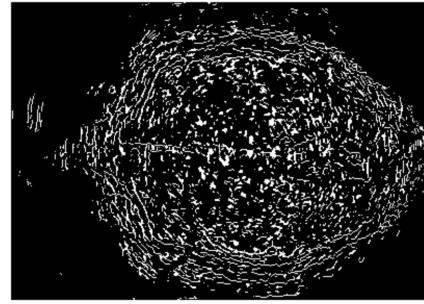
Figura 4.3: A la izquierda la imagen original a la derecha la imagen filtrada.

#### 4.1.4. Filtros de detección de bordes: Sobel y Canny

El filtro de Sobel detecta los bordes horizontales y verticales en una imagen en escala de grises. El filtro de Canny se usa para detectado de bordes en todas las direcciones, es un filtro basado en la primera derivada de una gaussiana. Como podemos ver en la figura 4.4 no son muy útiles en nuestro problema, ya que detectan demasiados bordes debido al alto ruido de las ecografías.



(a) Sobel



(b) Canny

Figura 4.4: Filtros de detección de bordes.

#### 4.1.5. Filtros de detección de crestas: filtro de Frangi

Originalmente diseñado para ser un filtro de resaltado de vasos sanguíneos[30], el filtro de Frangi es un filtro de detección de crestas. Llamamos crestas a estructuras lineales pero que son más anchas que una línea, por lo que son difíciles de detectar con filtros de resaltado de bordes. Como vemos en la figura 4.5, resalta muy bien el cráneo y además suaviza la imagen eliminando casi todo el ruido.

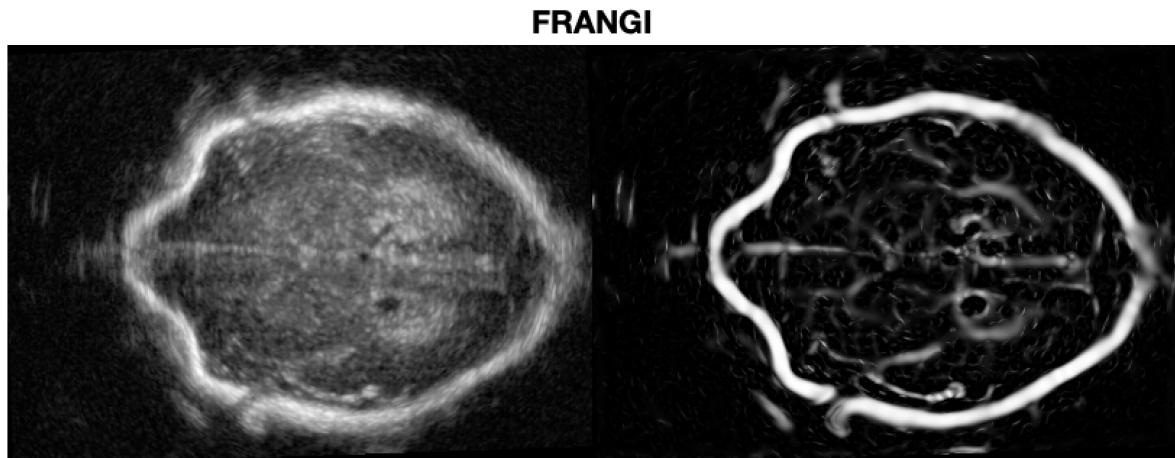


Figura 4.5: A la izquierda la imagen original a la derecha la imagen filtrada.

## 4.2. Segmentación

La segmentación de una imagen consiste en clasificar todos sus píxeles en una categoría. De esta forma sirve para identificar partes o regiones de una imagen. Podemos encontrar varios tipos de algoritmos con esta función.

### 4.2.1. K-means clustering

El algoritmo k-means clustering es un algoritmo de segmentación no supervisada que sirve para segmentar el área de interés del fondo. Divide los píxeles de entrada en

k-clústers basandose en k-centroides. La idea del algoritmo es minimizar la distancia de todos los píxeles de la imagen a su centroide. Supongamos k centroides:

- 1. Selecciona k puntos aleatorios como centroides iniciales  $\{m_1, m_2, \dots, m_k\}$
- 2. Para cada píxel  $x_j$  encuentra el clúster  $m_i$  más cercano a la característica  $f_j$  y asigna ese píxel al clúster i.
- 3. Recalcula  $\{m_1, m_2, \dots, m_k\}$  usando la media de los píxeles asignados.
- 4. Si se ha alcanzado el umbral de mínimo error para, si no vuelve al paso 2.

En la figura 4.6, podemos ver el resultado de aplicar a una imagen 2D el metodo de k-means con 3 centroides, a la imagen original le hemos aplicado un suavizado con el filtro de la mediana. Hemos seleccionado 3 centroides con la idea de diferenciar entre fondo(amarillo), cerebro(celeste) y cráneo(azul). Como vemos los resultados no son perfectos y confunde parte del ruido con el cerebro y parte del cerebro, con cráneo.

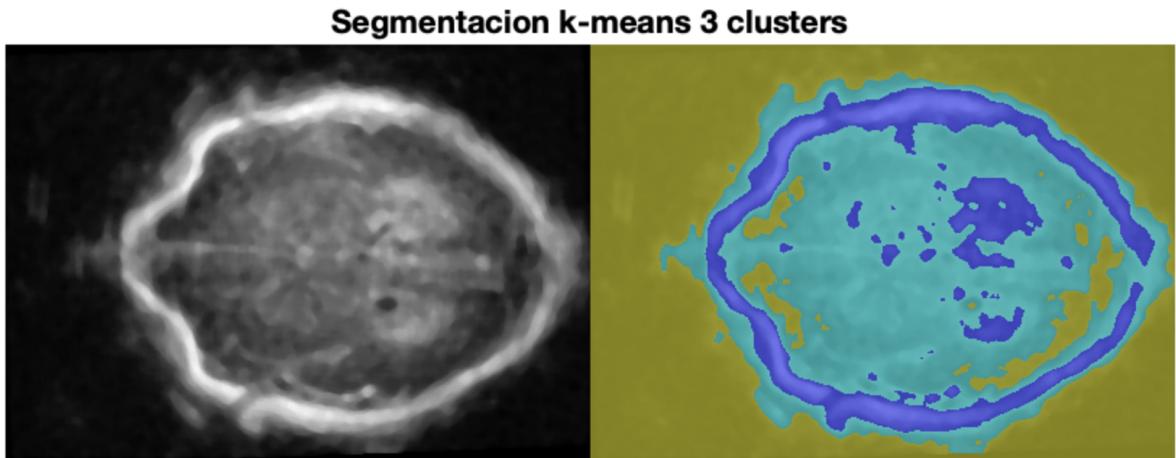


Figura 4.6: A la izquierda la imagen original con un filtro de la mediana a la derecha el resultado de k-means.

También podemos aplicar k-means pero utilizando, Frangi como preprocesamiento para intentar quedarnos solo con el cráneo. El procedimiento será aplicar el filtro de Frangi plano a plano a todo el volumen y después utilizar el algoritmo de k-means de matlab para imágenes 3D con 2 clústers. Utilizamos 2 clústers para ver si el algoritmo es capaz de diferenciar entre el cráneo (primer clúster) y el resto de la imagen (segundo clúster). Como vemos en la figura 4.7, el resultado no es perfecto, confunde el haz de luz inicial con el cráneo y hay muchas partes que están incompletas.

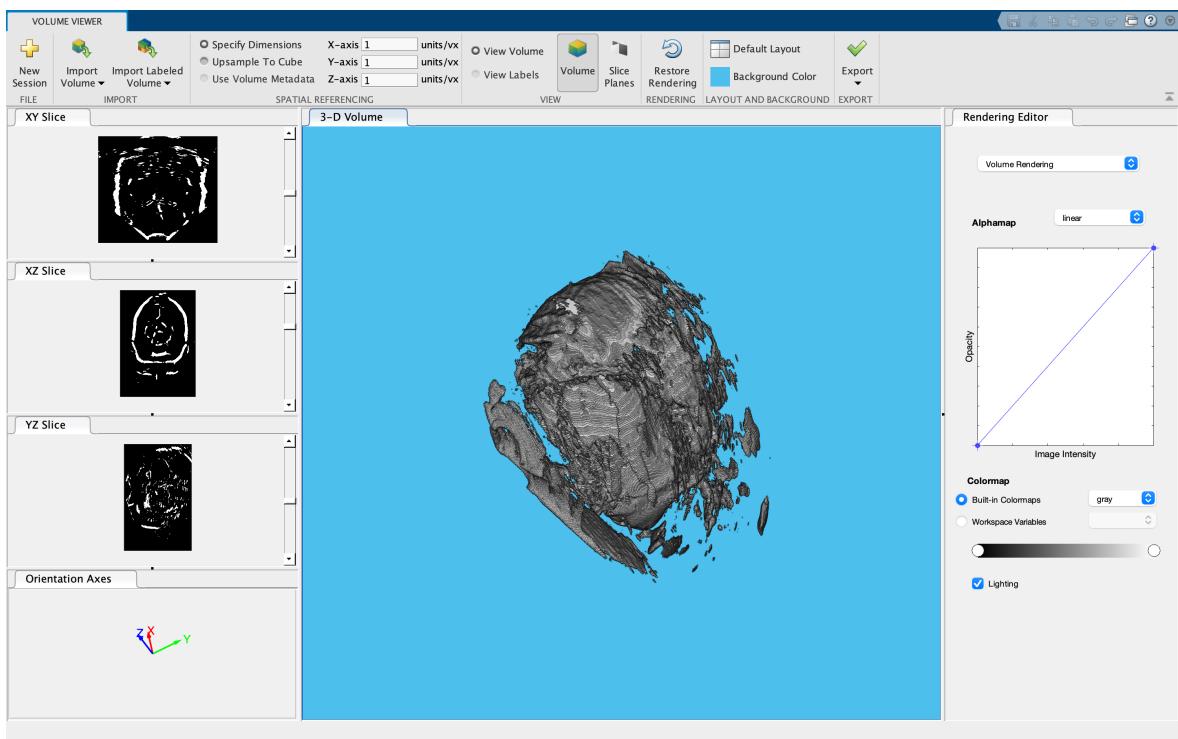


Figura 4.7: Resultado de aplicar Frangi y k-means al volumen completo.

#### 4.2.2. Superpixels

Un superpixel es un grupo de píxeles que comparten características similares y están situados unos al lado de otros en la matriz de píxeles de la imagen. Los algoritmos de Superpixel [31] agrupan los píxeles en regiones atómicas que tienen un significado común. Reducen la redundancia en la imagen, proporcionan una imagen primitiva y simple sobre la que calcular los vectores de características. Son muy útiles junto con métodos de segmentación, ya que aceleran los cálculos disminuyendo el tamaño de la imagen y producen mejores resultados porque simplifican la información.

Matlab tiene su propia función para el calculo de los superpixel[32] que esta basada en el algoritmo simple linear iterative clustering (SLIC)[31]. Tiene un funcionamiento muy similar a k-means. El algoritmo recibe como parámetro el numero k de superpixel que se quieren calcular. Como en este caso es muy importante la posición del píxel en el plano, podemos simplificar el calculo del centroide más cercano, comparándolo solo con los centroides que se sitúen cerca del píxel. El resto del procedimiento es muy similar. En la figura 4.8 podemos ver un ejemplo de esta función.

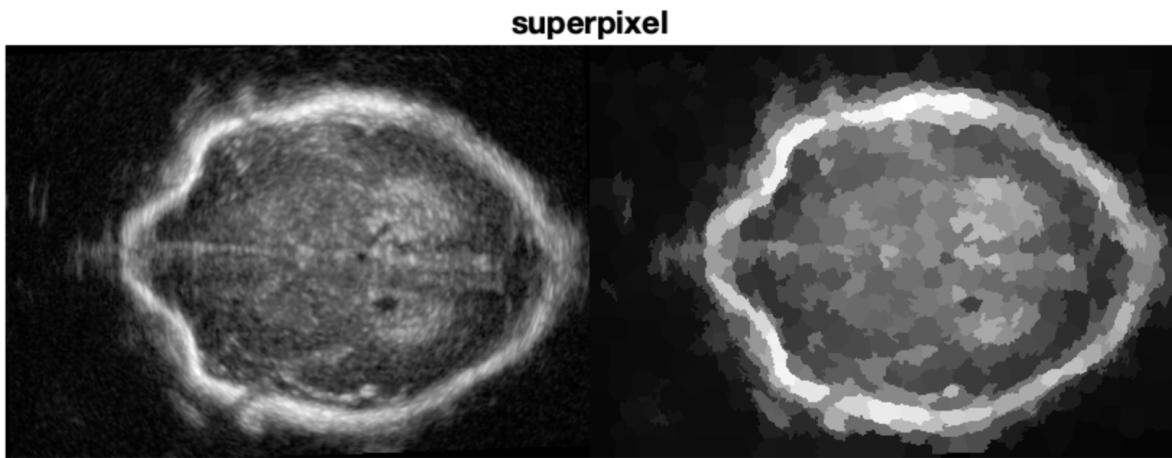


Figura 4.8: A la izquierda la imagen original, a la derecha la imagen resultante de aplicar superpixel.

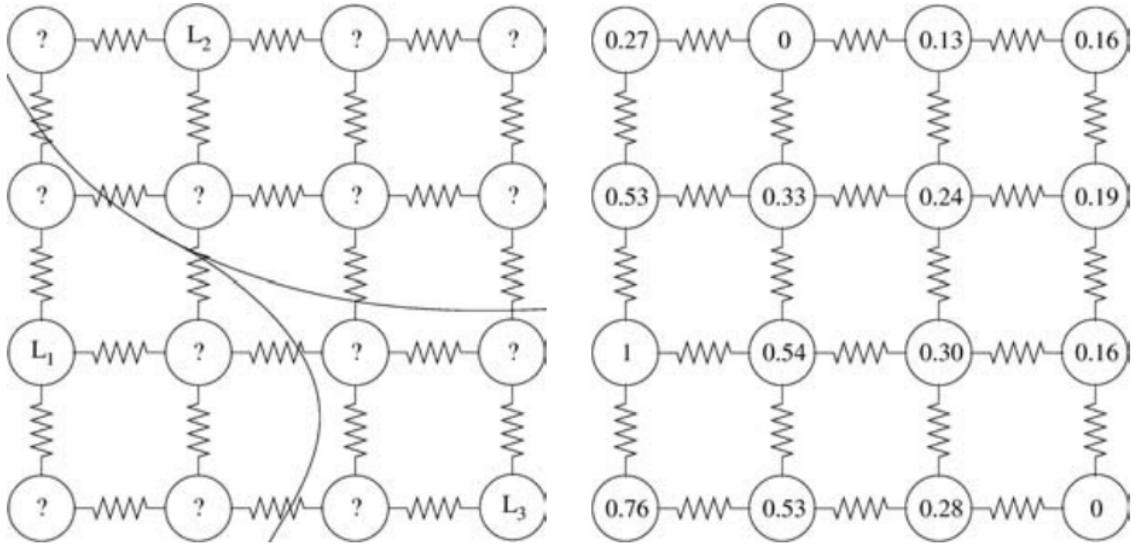
#### 4.2.3. Random Walks

Random Walks[2] se trata de un algoritmo de segmentación semiautomática. Dado un pequeño número de píxeles etiquetados, proporcionados por el usuario, se puede determinar la probabilidad de que un “caminante aleatorio” que parte de cada píxel no etiquetado llegue primero a uno de los píxeles etiquetados. Asignando cada píxel a la etiqueta de mayor probabilidad, se puede hacer una segmentación de buena calidad y rápida.

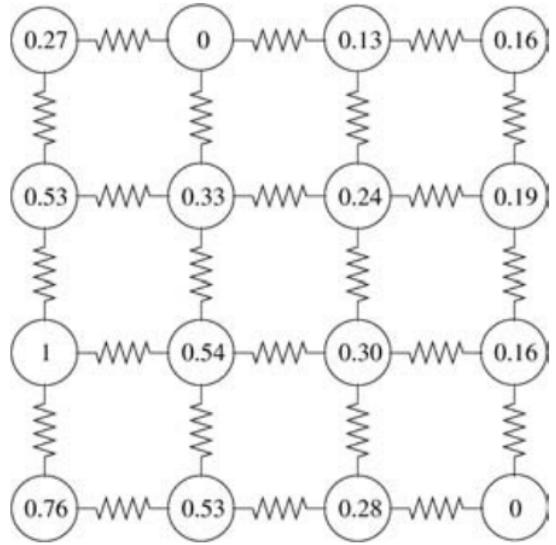
Entonces solo nos quedaría calcular la probabilidad de que un “caminante aleatorio” llegue desde un píxel no etiquetado a uno etiquetado. Para ello se representa la imagen como un grafo completamente conectado, y después este grafo se representa como un circuito eléctrico en el que el potencial de la etiqueta que queremos calcular se

coloca a 1 y el resto a 0. Después, se resuelve el potencial del resto de nodos, el potencial será equivalente a la probabilidad. El cálculo de estos potenciales está desarrollado en [2].

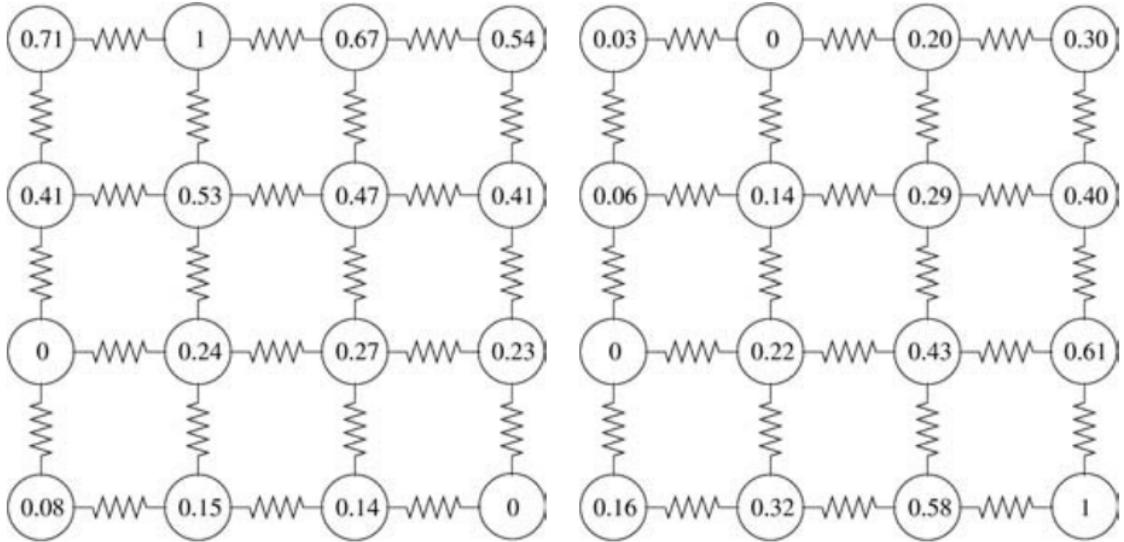
En la figura 4.10, podemos ver un ejemplo de aplicar Random Walks a una imagen 2D dibujando 2 máscaras para el fondo y 1 para el cerebro. En la figura 4.10a, encontramos las dos máscaras dibujadas por el usuario. Roja para el cerebro y azul para el fondo. En la figura 4.10b, podemos ver el resultado de la primera segmentación. Como vemos, ha quedado abierta por abajo, así que añadimos otra máscara de fondo dibujada por el usuario. En 4.10c, podemos ver el resultado final de la segmentación.



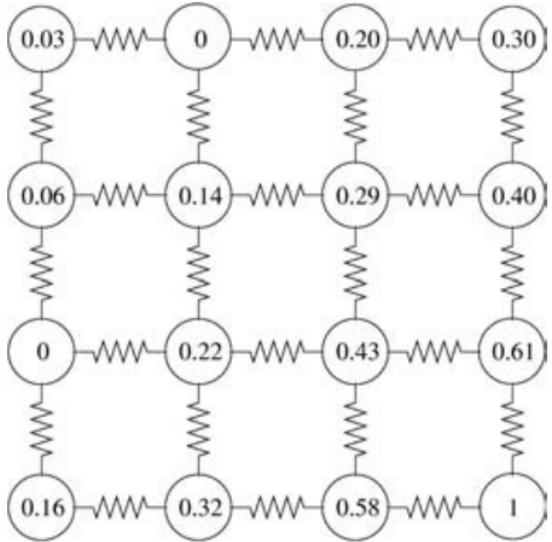
(a) Seed points with segmentation



(b) Probability that a random walker starting from each node first reaches seed  $L_1$

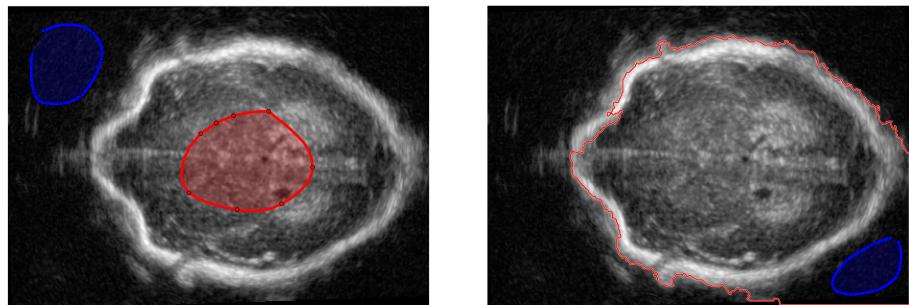


(c) Probability that a random walker starting from each node first reaches seed  $L_2$



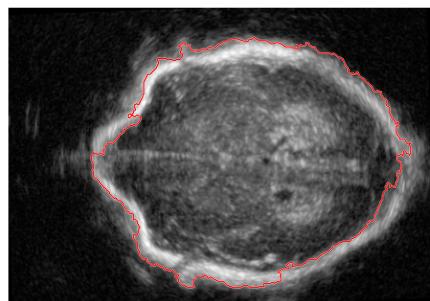
(d) Probability that a random walker starting from each node first reaches seed  $L_3$

Figura 4.9: Representación de la imagen como un circuito. Fuente: Imagen extraída de [2].



(a) Máscaras iniciales. En rojo los píxeles etiquetados como “cerebro”. En azul los píxeles etiquetados como “fondo”.

(b) En rojo el resultado de la segmentación inicial. Añadimos una segunda máscara de píxeles etiquetados como “fondo”.



(c) Resultado final. En rojo el contorno de la zona segmentada como “cerebro”.

Figura 4.10: Resultado de la segmentación utilizando Random Walks.



# 5. Deep Learning

---

El Deep Learning para la segmentación automática engloba aquellos algoritmos de aprendizaje automático que usando redes neuronales multicapa aprenden a realizar una tarea concreta a partir de un gran conjunto de datos. En este capítulo veremos como el Deep Learning puede ayudarnos a resolver nuestro problema.

## 5.1. Operación de convolución

Los filtros de convolución toman su nombre de la operación de convolución de dos matrices. Esta operación permite extraer información de la imagen analizándola poco a poco, como si realizáramos un escaneo de la imagen de izquierda a derecha y de arriba a abajo. Dada la imagen  $I$  de tamaño  $(x,y)$  y el filtro de convolución  $k$  de tamaño  $(3,3)$ . Colocamos el filtro arriba a la izquierda de la imagen e iremos desplazándolo poco a poco por la imagen extrayendo la información de la misma.

En la imagen se toma una pequeña parte de tamaño  $3 \times 3$  y se multiplica cada píxel  $I_{ab}$  por cada elemento en la posición correspondiente del kernel  $K_{ab}$ . Después se suman todos los píxeles del kernel y ese será el valor del píxel en la imagen  $S$ . Iremos desplazando el kernel  $K$  poco a poco por la imagen  $I$ , desde el principio hasta el final para tener como resultado una imagen completa, que no tiene por qué ser del mismo tamaño que la imagen de entrada, ya que esto dependerá de como se mueva el kernel por la imagen original.

## 5.2. Redes neuronales convolucionales(CNN)

Las CNN son principalmente usadas en el campo de reconocimiento de patrones en imágenes, aunque podrían usarse en otros ámbitos.

La principal ventaja que tienen sobre otros tipos de redes neuronales es que, al estar especializadas para el reconocimiento de patrones en imágenes, son mucho más eficientes a la hora de gestionar la complejidad de las mismas.

Supongamos que tenemos una red neuronal con sus capas completamente conectadas para realizar la tarea de segmentación. En la capa de entrada podríamos coger todos los píxeles de la imagen que se encuentran en forma de matriz y “aplanarlos” en un vector con sus valores. Tendríamos que conectar cada neurona de la capa oculta con todas las neuronas de la capa de entrada. De esta forma perderíamos la información espacial de cada uno de los píxeles y además tendríamos muchísimos parámetros.

Para procesar la imagen, manteniendo la estructura espacial de los píxeles y reduciendo la complejidad computacional, las CNN utilizan capas de convolución, de las que toman el nombre. Las capas de convolución son capas en las que cada una de

35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

\*

-2	-1	0
-1	1	1
0	1	2

=

				78

$35 \cdot (-2) + 40 \cdot (-1) + 41 \cdot 0 + 40 \cdot (-1) + 40 \cdot 1 + 42 \cdot 1 + 42 \cdot 0 + 46 \cdot 1 + 50 \cdot 2 = 78$

35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

\*

-2	-1	0
-1	1	1
0	1	2

=

				87

$40 \cdot (-2) + 41 \cdot (-1) + 45 \cdot 0 + 40 \cdot (-1) + 42 \cdot 1 + 46 \cdot 1 + 46 \cdot 0 + 50 \cdot 1 + 55 \cdot 2 = 87$

35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

\*

-2	-1	0
-1	1	1
0	1	2

=

35	40	41	45	50
40	78	87	94	52
42	98	283	108	55
48	120	125	127	60
56	60	65	70	75

Figura 5.1: Ejemplo de convolución. Fuente: Imagen extraída de [3].

sus neuronas aplicará un filtro de convolución a una pequeña región de la imagen, por esto se dice que las neuronas de la capa de convolución están localmente conectadas. El resultado de cada neurona un vector de características. También cuentan con otras capas que veremos en la sección 5.2.1 Arquitectura general.

### 5.2.1. Arquitectura general

Las CNN están formadas por 3 tipos de capas: Capas de convolución (*Convolution layers*), capas de agrupación (*pooling layers*), capas de clasificación (*fully-connected layers*) [4].

El funcionamiento básico de una red neuronal convolucional es el siguiente:

- 1. La capa de entrada contendrá el valor de los píxeles de la imagen. Será una matriz de dos dimensiones que guarde la estructura espacial de estos píxeles. Esto es muy importante, ya que en la tarea de segmentación de imágenes no solo es importante el valor de la intensidad de cada píxel, sino que su posición en la imagen y la relación con los píxeles que le rodea es de vital importancia.
- 2. Cada neurona de la capa de convolución esta conectada localmente a la capa de entrada. Es decir, cada neurona NO recibe todos los píxeles de la imagen, sino solo una pequeña región de esta. De esta se forma reduce muchísimo la

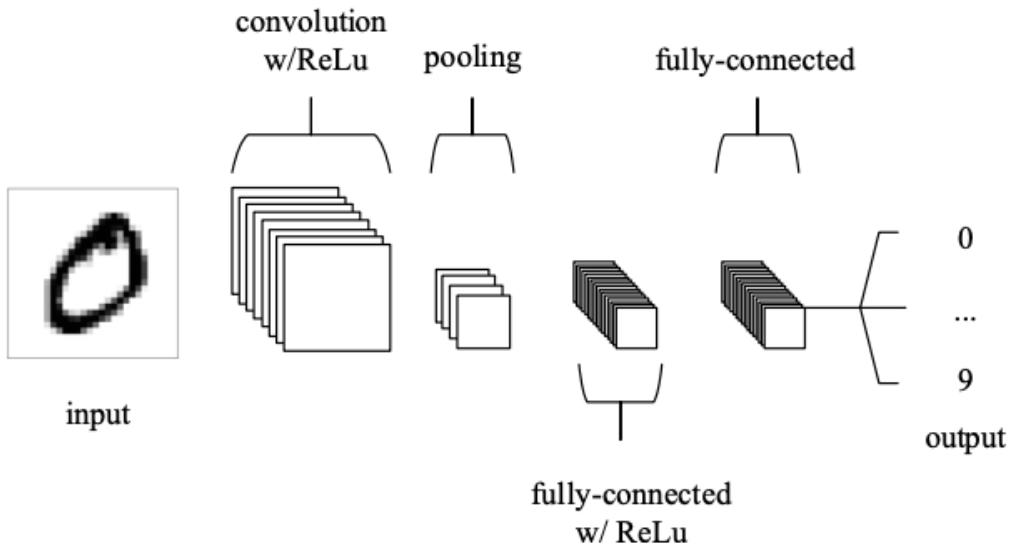


Figura 5.2: Red neuronal convolucional. Fuente:Imagen extraída de [4].

complejidad computacional y se consigue preservar el significado de la posición de cada píxel en la imagen. Cada neurona aplica una ventana de pesos, calcula las combinaciones lineales y aplica una función de activación no lineal.

Este es un ejemplo del cálculo de la salida de una neurona siendo la matriz de pesos  $W$ , la imagen  $X$ ,  $(p,q)$  la posición de la neurona y  $b$  el bias.

$$\sum_{i=1}^4 \sum_{j=1}^4 W_{ij} + X_{i+p,j+q} + b$$

- 3. La *pooling layer* realiza la operación de pooling. Sirve para reducir la dimensionalidad, preservando la estructura espacial. Un ejemplo de operación de pooling que podríamos utilizar es *Max Pooling*. Consiste en seleccionar grupos de píxeles, por ejemplo, de  $3 \times 3$  y quedarnos solo con el máximo de cada grupo.
- 4. Finalmente, toda la información extraída puede aplanarse e introducirse en capas completamente conectadas para realizar la decisión final, en nuestro caso la segmentación de la imagen.

### 5.2.2. Técnicas de aumento de datos

Cuando trabajamos con redes neuronales convolucionales es muy importante disponer del máximo número de datos posibles. El problema es que, al estar trabajando con imágenes, a veces es muy complicado obtener imágenes nuevas y además etiquetarlas es una tarea tediosa que normalmente debe ser realizada por humanos.

Por ejemplo, en nuestro caso para obtener más volúmenes tendríamos que solicitarlos al hospital y, si no disponen de ellos, un profesional tendría que realizar nuevas

ecografías a bebés del área de neonatología. Además, después habría que segmentar los nuevos volúmenes.

Las técnicas que, aplicando algunas transformaciones sencillas, modifiquen los datos de entrenamiento pero sin requerir volver a etiquetarlos manualmente se llaman técnicas de aumento de datos. Por ejemplo, si giramos una imagen 45 grados para una persona seguirá siendo la misma, pero su matriz de píxeles será completamente distinta.

Estas técnicas incluyen redimensión, rotación y reflexión de las imágenes.

### 5.3. Arquitectura UNET

UNET es una arquitectura especialmente desarrollada para la segmentación semántica de imágenes. Es decir, combinando las capas comunes de las redes neuronales convolucionales vistas en la sección anterior (5.2) los autores Olaf Ronneberger, Philipp Fischer y Thomas Brox han creado su propia arquitectura de red neuronal.

UNET es especialmente buena con conjuntos de imágenes reducidos y es bastante rápida [5]. Esta diseñada para la segmentación de imágenes biomédicas.

Podemos ver la arquitectura de la red en la figura 5.3. Esta formada por un camino contractivo, el lado izquierdo, y un camino expansivo, el lado derecho, lo que le da su forma característica de U. Cosiste en aplicar de forma repetida dos convoluciones de 3x3 (unppaded convolution), cada una seguida por una capa ReLu y una capa de max pooling con stride 2 para el downsampling. Después de cada paso de downsampling se duplican el número de características. Cada paso en la parte expansiva consiste en un upsampling del mapa de características seguido de una convolución 2x2 (up-convolution) que reduce a la mitad el mapa de características. Una concatenación con su correspondiente parte del mapa de características del camino contractivo, y dos convoluciones 3x3, cada una seguidas por una ReLu. Al final, una capa de convolución 1x1 es necesaria para mapear el vector de características al número deseado de clases. En total la red tiene 23 capas convolucionales.

### 5.4. Arquitectura SegNet

SegNet está diseñada para ser una arquitectura eficiente en la segmentación semántica de imágenes[6]. Normalmente debido a las operaciones de max-pooling y subsampling se pierde resolución en los mapas de características.

La motivación de diseñar SegNet surge de la necesidad de mapear características de baja resolución a la resolución de entrada. Fue diseñada principalmente para solucionar el problema de segmentación en imágenes de la carretera, en el que es muy importante delimitar bien el borde de los objetos más pequeños.

El encoder es casi idéntico al que encontramos en VGG16[33], la única diferencia es que no tiene las capas completamente conectadas. La clave de la arquitectura

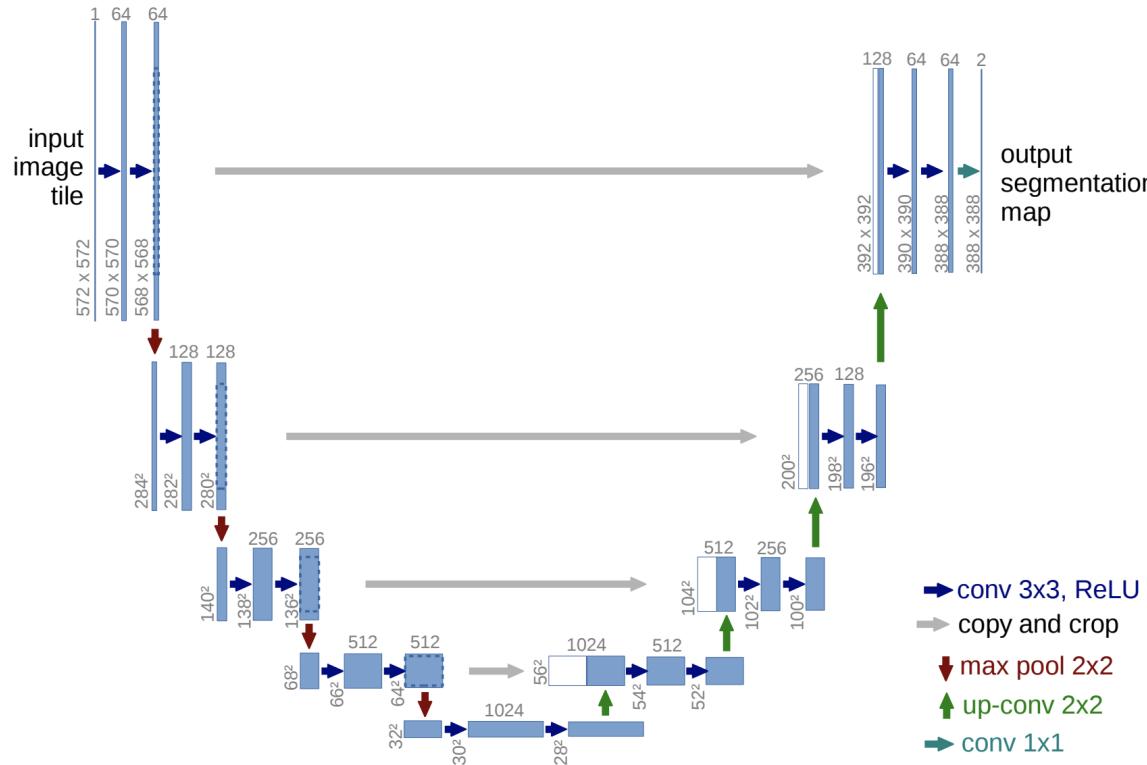


Figura 5.3: Arquitectura UNET. Cada cuadro azul corresponde a un mapa de características multicanal. El número de canales se especifica en la parte superior de la caja. El tamaño x-y se especifica en el borde inferior izquierdo de la caja. Las cajas blancas representan mapas de características copiados. Las flechas denotan las diferentes operaciones. Fuente: Imagen extraída de [5]

de SegNet está en el decoder. Cada decoder se corresponde con un encoder, están jerárquicamente organizados. Cada decoder utiliza los índices de max pooling de su correspondiente encoder para realizar el upsampling de los mapas de características de entrada. Esto tiene las siguientes ventajas:

- 1. Mejora la delineación de bordes.
  - 2. Reduce el número de parámetros permitiendo el entrenamiento end-to-end.
  - 3. Es fácil de implementar en cualquier otra arquitectura, aplicando solo algunos cambios.

Segnet tiene una red encoder y una red decoder, seguida de una capa de clasificación final. El encoder se corresponde con las 13 primeras capas convolucionales de VGG16, diseñada para la clasificación de objetos. Cada capa del decoder se corresponde con una capa del encoder, por lo que el decoder tiene 13 capas.

Cada encoder realiza la convolución para generar un banco de filtros, a continuación son normalizados utilizando Batch Normalization. Después, se les aplica una capa ReLu. Después, se realiza un max-pooling con una ventana de 2x2 y un stride de 2, creando un subsampling por un factor de 2.

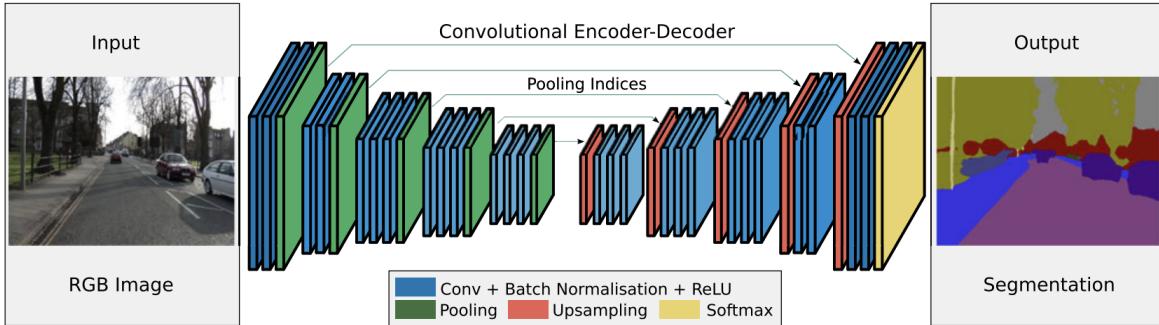


Figura 5.4: Arquitectura SegNet. No tiene capas completamente conectadas. El decoder realiza el upsampling utilizando los índices transferidos del encoder. Realiza la convolución para crear el mapa de características. Al final tiene un soft-max clasifier para la clasificación de los píxeles. Fuente: Imagen extraída de [6]

El decoder apropiado en la red de decoder realiza el upsampling utilizando los índices del max pooling memorizados. Estos mapas de características obtenidos son convolucionados para producir mapas densos de características. Se sigue la misma estructura de convolución que en el encoder.

Al final se utiliza un clasificador soft-max. Este clasificador, clasifica cada píxel independientemente. La salida del clasificador es una imagen con  $K$  canales, donde  $K$  es el número de clases. La segmentación se corresponderá con la clase de máxima probabilidad para cada píxel.

## 5.5. Entrenamiento de las redes UNET y SegNet

Para resolver nuestro problema entrenaremos 2 tipos de redes, UNET y SegNet. En esta sección veremos el software que hemos desarrollado para etiquetar los volúmenes, la generación de la base de datos y el entrenamiento de las redes.

### 5.5.1. Software de segmentación semi-automática

Se ha desarrollado un software que presenta una segmentación automática. Luego se puede visualizar esta segmentación, en los tres planos y como volumen, y retocarla. El retoque manual será necesario en casi todos los casos, ya que el algoritmo usado está bastante limitado.

Este software nos servirá para generar una base de datos de volúmenes bien segmentados, que nos permitan posteriormente generar las imágenes para entrenar las redes neuronales.

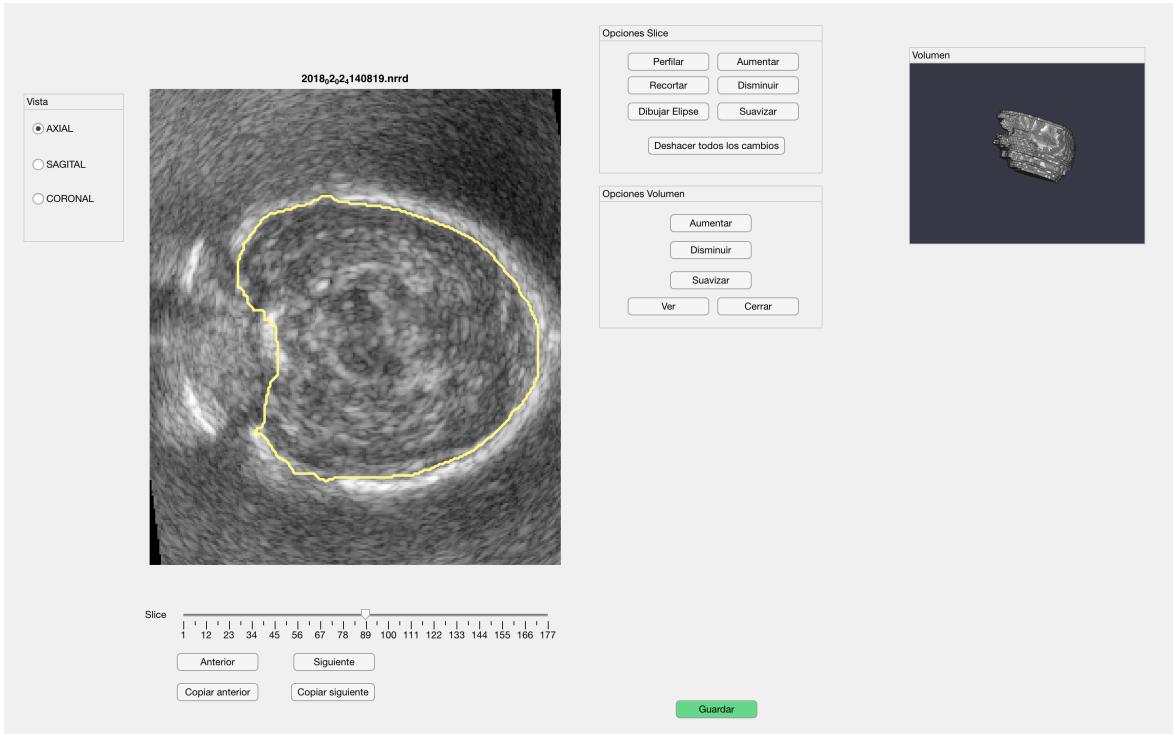


Figura 5.5: Software de segmentación con retoque manual. En amarillo veremos el contorno de la zona segmentada, a la izquierda en el cuadro de vista podremos seleccionar que plano queremos visualizar. La barra Slice permite seleccionar el número del plano que queremos ver y cuenta con botones para movernos por el volumen. Para modificar la segmentación contamos con las opciones de perfilar, aumentar, recortar, disminuir, dibujar una nueva ellipse y suavizar

## Algoritmo de Segmentación

El Algoritmo de segmentación automática del volumen desarrollado para el software esta basado en el algoritmo de Random Walks 4.2.3. El algoritmo segmenta una a una todas las imágenes del plano axial, resultando en la segmentación completa del volumen.

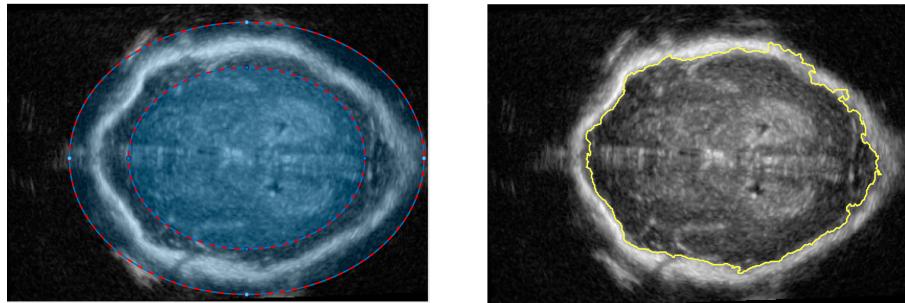
Como ya explicamos, el algoritmo Random Walks requiere de un etiquetado inicial para clasificar el resto de píxeles. Una opción sería pedirle al usuario que etiquetara el cerebro y el exterior, en cada imagen (cada volumen cuenta con mas o menos 170 imágenes en el plano axial), teniendo en cuenta que además después tendrá que retocar la segmentación.

Para facilitar la tarea se ha diseñado un algoritmo que realiza un etiquetado inicial automático. El funcionamiento es el siguiente:

- **1.** Busca el inicio y el final del cráneo. Utilizando el filtro de Frangi y regionprops<sup>[34]</sup> detecta el slice de inicio y de fin del craneo en el plano axial.
- **2.** Desde el slice del medio, que suele ser en el que mejor se diferencia el cráneo del fondo y del cerebro, se realiza una búsqueda del cráneo utilizando un suavizado de la mediana + Frangi + k-means. Después se interpola una ellipse sobre esta

segmentación, simulando el cráneo.

- **3.** Todos los píxeles que queden por fuera de la elipse que representa el cráneo se clasificarán como fondo. La misma elipse se disminuye un poco y los píxeles que queden dentro se clasificarán como cerebro. Véase la figura 5.6a.
- **4.** Ya tenemos las etiquetas iniciales para el Random Walk. Así que lo aplicamos y obtenemos la segmentación de la primera imagen. Véase la figura 5.6b.
- **5.** Para el resto de imágenes del volumen se utilizará la segmentación de la imagen anterior, ya que habrá poca variación en la imagen de slice a slice. Sabiendo que a partir del slice del medio el resto serán más pequeños, la segmentación del cerebro de la imagen anterior la haremos un poco más pequeña y clasificaremos esos píxeles como cerebro, después la haremos más grande y clasificaremos esos píxeles como fondo, después aplicaremos Random Walks de nuevo.
- **6.** Realizaremos el procedimiento anterior para todos los slices desde el inicio del cráneo hasta el fin, calculados en el paso 1.



(a) Paso 3 del algoritmo de segmentación con Random Walk. Las dos elipses calculadas para iniciar el algoritmo de RW.  
(b) Paso 4 del algoritmo de segmentación con random Walks Resultado de la segmentación a partir de las máscaras calculadas automáticamente.

Figura 5.6: Tercer y cuarto paso del algoritmo de segmentación utilizando Random Walks.

## Herramientas de retoque

Como ya sabemos, la segmentación anterior no será perfecta. Normalmente la segmentación automática suele estimar que el cerebro es más grande, tomando parte del cráneo como cerebro. En algunos casos si la ecografía es muy ruidosa, simplemente no funcionará.

Así que hemos desarrollado algunas herramientas de retoque integradas en una interfaz de usuario diseñada con Matlab App Designer[35]:

- **Perfilar:** Permite dibujar un contorno con el ratón. Todos los píxeles del interior del contorno quedarán clasificados como cerebro.
- **Recortar:** Permite dibujar un contorno con el ratón. Todos los píxeles del interior del contorno quedarán clasificados como fondo.

- **Aumentar:** Aplica una dilatación a la máscara. Aumentando el tamaño del cerebro.
- **Disminuir:** Aplica una erosión a la máscara. Disminuyendo el tamaño del cerebro.
- **Dibujar Elipse:** Permite dibujar una nueva elipse que sustituirá a la segmentación actual.
- **Suavizar:** Aplica un suavizado de bordes. Normalmente la segmentación automática tiene como resultado dientes de sierra que pueden ser suavizados para mejorar la segmentación.
- **Deshacer todos los cambios:** Deshace los cambios volviendo a la segmentación original.

## Limitaciones del algoritmo

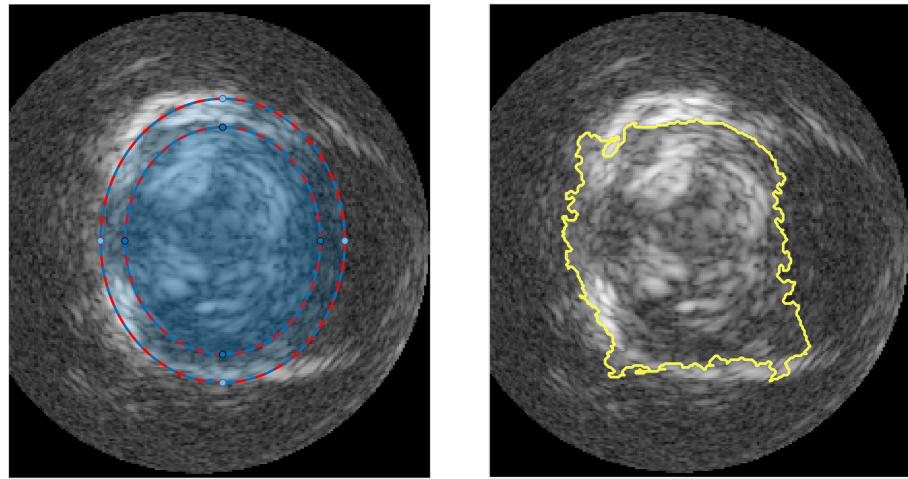
El algoritmo desarrollado no es perfecto, por eso se han dispuesto al usuario las herramientas necesarias para el retoque de las segmentaciones.

El primer problema que puede ocurrir es que, debido al ruido, las elipsoides iniciales se sitúen mal en la imagen. Este error es de gran impacto, ya que se propagará por el resto de los slices dando como resultado una segmentación inservible. Véase la figura 5.7.

Otro problema, es que el algoritmo utiliza como base para la segmentación la imagen anterior, por lo que un pequeño error en la máscara anterior se propagará por el resto de slices empeorando en cada iteración. Esto suele provocar que se realicen mal las segmentaciones de las imágenes que pertenecen a los extremos. Véase la figura 5.8.

Otro error muy común, es que el algoritmo tiende a sobreestimar el tamaño del cerebro confundiendo parte del mismo con el cráneo. Además, este error suele propagarse por el resto de planos por lo que para corregirlo es muy útil realizar una disminución del volumen segmentado. Véase la figura 5.9.

Además de todo esto, el tamaño del cráneo tiende a ser subestimado. Por lo que el inicio y el fin no serán reales y solo se segmentan las imágenes que se sitúan en las capas intermedias del cerebro.



(a) Elipses mal situadas.

(b) Resultado.

Figura 5.7: Ejemplo de elipsoides iniciales mal situadas debido al alto ruido de la imagen, utilizando el algoritmo de segmentación.

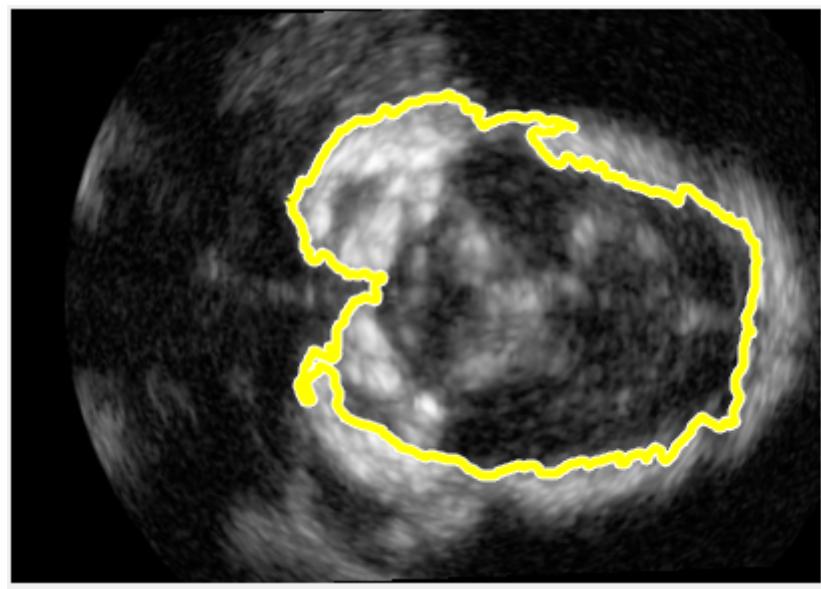
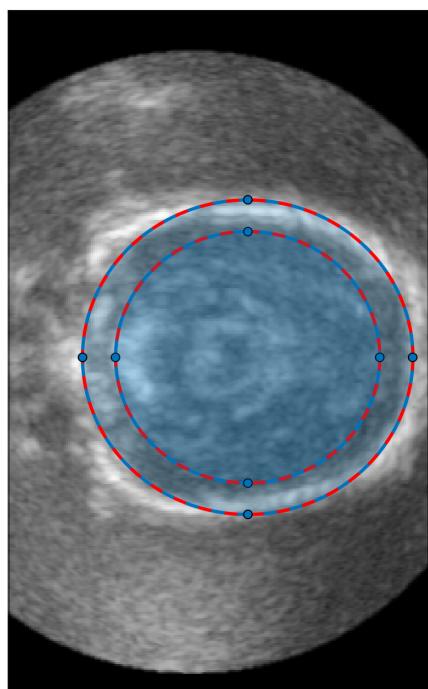
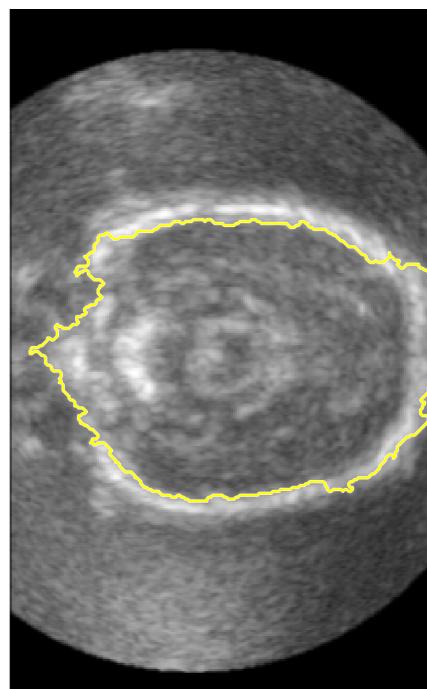


Figura 5.8: Ejemplo de error en una segmentación de una imagen del plano axial en uno de los extremos, debido a la propagación del error de Random Walks. Parte del cráneo ha sido clasificada como “cerebro”



(a) Elipses iniciales.



(b) Resultado.

Figura 5.9: Ejemplo de sobreestimación del cerebro por el algoritmo de segmentacion.

### 5.5.2. Generar la base de datos

Para el entrenamiento de ambas redes se han segmentado, utilizando el software de segmentación desarrollado, 59 volúmenes. El proceso de segmentación es bastante lento y tedioso, por lo que necesitamos 3 semanas para realizar las segmentaciones.

Entrenaremos las redes para que aprendan a segmentar imágenes 2D del plano axial de los volúmenes. La segmentación de un volumen, siguiendo un protocolo similar al que vimos en la segmentación con Random Walks, consistirá en que la red segmentará una a una todas las imágenes del plano axial, resultando en un volumen completamente segmentado.

Para el entrenamiento, extraeremos todas las imágenes del plano axial de los volúmenes segmentados que tengan una máscara del cerebro correspondiente. Por lo que el conjunto de imágenes de entrenamiento estará formado, mayoritariamente, por imágenes que pertenecen a la zona media del cerebro que es la parte donde el algoritmo de Random Walks segmenta mejor. Esperando que la red sea capaz de aprender a segmentar el resto por si sola, pues en los extremos el algoritmo es poco fiable según hemos podido comprobar.

Para el entrenamiento de ambas redes utilizaremos 3 clases, en la figura 5.10 podemos ver un ejemplo. A partir de las máscaras del cerebro de las que disponemos, etiquetaremos los píxeles de la máscara como cerebro, los píxeles que sean igual a 0 como fondo y todos los píxeles que sean distinto de 0 como no-cerebro. Estos píxeles podrían ser ruido o el haz de luz de las ecografías pero servirá para descartarlos como cerebro. Así, también conseguimos equilibrar el número de píxeles que pertenecen a cada clase.

Hay que tener en cuenta que la entrada de SegNet son imágenes con 3 canales preparadas para imágenes a color RGB. La solución que hemos desarrollado ha sido generar estos tres canales, seleccionando la siguiente imagen y la anterior. Esta idea podría ser interesante ya que le proporciona a la red algo de información espacial. Por lo tanto contamos con un script llamado *preprocessdataset.m* que, si la imagen tiene alguna parte etiquetada como cerebro, apila esa imagen, la anterior y la siguiente y genera su etiqueta correspondiente.

Partiendo de los 59 volúmenes iniciales, segmentados con random walk de forma automática según hemos comentado anteriormente y retocadas las segmentaciones con el software desarrollado, se generaron 6692 imágenes.

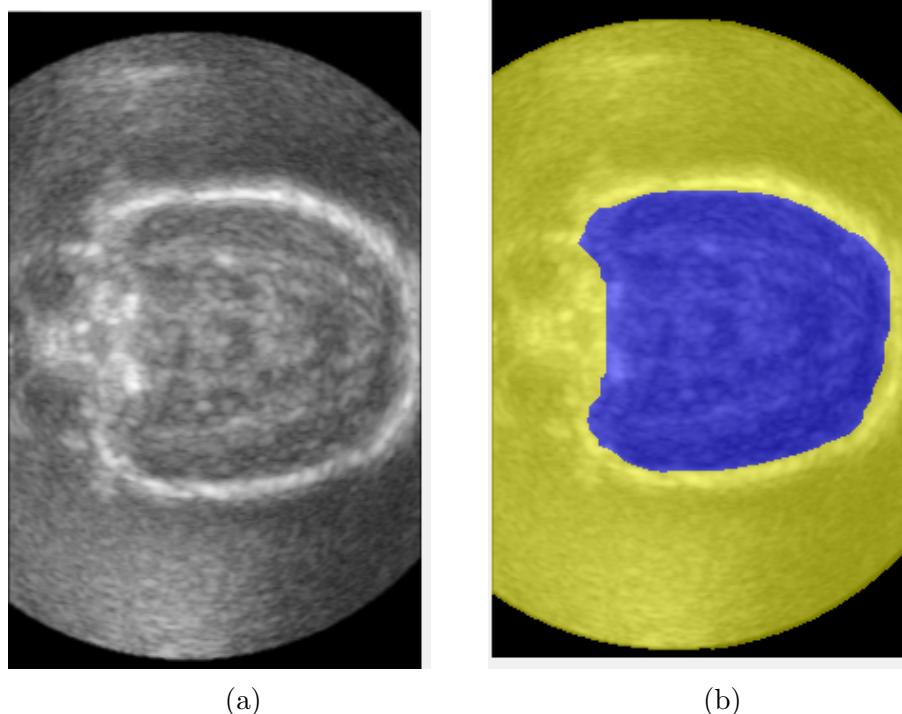


Figura 5.10: Ejemplo de etiquetado de las imágenes segmentadas. (a) Imagen extraída del plano axial. (b) Etiqueta correspondiente con 3 clases. En azul el “cerebro”, en negro el “fondo”, en blanco ‘‘no-cerebro’’.

### 5.5.3. Entrenamiento

En esta sección veremos como hemos realizado el entrenamiento de cada red. Ambas han sido entrenadas en el mismo conjunto de datos de entrenamiento, en concreto de las 6692 imágenes disponibles el 80 % se usaron como datos de entrenamiento, el 10 % como datos de validación y el otro 10 % como datos de test.

#### UNET

La arquitectura UNET se ha configurado con una profundidad del encoder de 3 niveles, por lo que el decoder será del mismo nivel. El encoder reduce la muestra (*downsampling*) por un factor de  $2^D$  donde D es el número de niveles, el decoder aumenta la muestra (*upsampling*) también por un factor de  $2^D$ . El número de niveles se ha seleccionado de forma empírica, realizando distintas pruebas y viendo cual proporciona mejores resultados.

La capa de entrada de las imágenes se ha configurado para que realice normalización de las intensidades de las imágenes entre 0 y 1.

La capa de salida de la red es una capa DICE. Esta capa proporciona una clasificación para cada pixel de la imagen utilizando la perdida generalizada DICE. La capa utiliza la perdida generalizada de DICE para solucionar el problema del desequilibrio de clases de la segmentación semántica. Controla la contribución de cada clase a la perdida ponderando las clases por el tamaño inverso de la región esperada.

#### SegNet

Para la red segnet se ha utilizado un modelo preentrenado VGG16. Estos modelos tienen un encoder de 5 niveles de profundidad. Como capa de salida se ha añadido una capa de clasificación de píxeles con ponderación de pesos de los píxeles.

#### Opciones de entrenamiento

Ambas redes han sido entrenadas con las mismas opciones de entrenamiento. Con imágenes de 200x200.

- solvername: 'sgdm'. Optimizador de descenso de gradiente estocástico con momentum.
- LearnRateSchedule: 'piecewise'. especifica la tasa de aprendizaje inicial como 0,03 y el factor de regularización L2 como 0,0005, e indica al software que reduzca la tasa de aprendizaje cada número determinado de epochs multiplicando por un factor determinado.
- InitialLearnRate: '1e-2'.
- ValidationData: se han usado los datos de validación seleccionados.

- MaxEpochs: 10. El número de epochs es el número de veces que han pasado los datos por el algoritmo de aprendizaje.
- MiniBatchSize: 8. Un miniBatch es un subconjunto de los datos que se utiliza para evaluar el gradiente de la función de perdida y evaluar los pesos.
- Shuffle: 'every-epochs'. Mezcla los datos después de cada epoch.

## Aumento de datos

Para el entrenamiento de ambas redes se han usado técnicas de aumento de datos. De esta forma, sin tener que segmentar manualmente más volúmenes, aumentamos nuestros datos disponibles significativamente.

Se ha aplicado rotación desde -45 hasta 45 grados. Cizallamiento horizontal y vertical desde -30 hasta 30 grados. Translación en el eje X e Y desde -10 hasta 10 píxeles.

### 5.5.4. Resultados del entrenamiento

En esta sección veremos los resultados del entrenamiento de ambas redes. Se han utilizado el 10 % de las imágenes disponibles como datos de test para calcular las medidas, en total 662.

En la tabla de la figura 5.12, podemos ver las métricas globales de cada red. Veámoslas una a una:

- **GlobalAccuracy:** es la relación entre los píxeles clasificados correctamente, independientemente de la clase, y el número total de píxeles. Nos sirve para obtener una estimación rápida y poco costosa desde el punto de vista computacional del porcentaje de píxeles clasificados correctamente, en esta métrica SegNet es mejor que Unet.
- **MeanAccuracy:** es la media del Accuracy de todas las clases, en esta métrica SegNet, también, supera a Unet.
- **Mean IoU:** la métrica (intersection over union), es muy importante y ampliamente utilizada. Si en una imagen clasificáramos todos los píxeles como cerebro, tendríamos un 100 % de acierto pero la segmentación sería errónea. Para cada clase, IoU es la relación entre los píxeles clasificados correctamente y el número total de píxeles reales y predichos en esa clase. MeanIoU es la media de IoU de todas las clases. En esta métrica SegNet, también, supera a Unet.
- **WeightedIoU:** Es el promedio del IoU de cada clase ponderado por el número de píxeles de cada clase. Es muy útil cuando hay descompensación de número de píxeles en cada clase, como es el caso. En esta métrica SegNet también supera a Unet.

- **MeanBFScore:** El boundary F1 (BF) Score, indica lo bien que se alinea la frontera predicha de cada clase con la frontera verdadera. En esta métrica SegNet, también, supera a Unet.

En la figura 5.13 podemos ver las métricas anteriormente explicadas, por clase. Como vemos para el cerebro, que es la clase que nos interesa ya que el resto lo descartaremos, SegNet tiene mayor tasa de acierto y mejor indice IoU y mejor BFScore. Lo que la hace muy superior a Unet para nuestro problema.

En la figura 5.14 y la figura 5.15, podemos ver las matrices de confusión para Unet y SegNet respectivamente. La matriz de confusión nos ayuda a ver cuando nuestra red se “confunde” entre dos clases. En las celdas en las que la fila y columna tienen el mismo nombre, se encuentran los píxeles bien clasificados para esa clase, en las celdas con fila y columna de distinto nombre se encuentran los falsos positivos.

En la figura 5.16 y la figura 5.17 podemos ver algunas segmentaciones realizadas por las redes.

Según todas estas métricas, podemos concluir que la red SegNet es superior a Unet para la resolución de nuestro problema de segmentación.

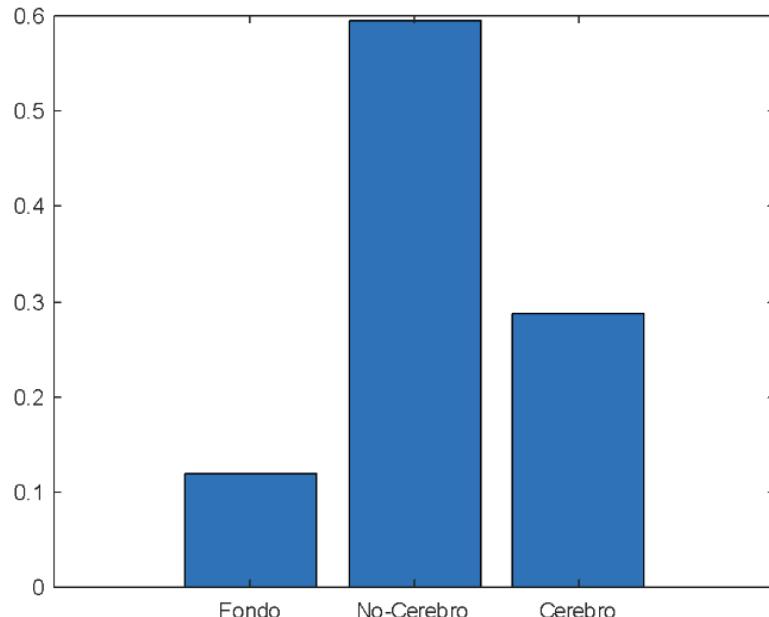


Figura 5.11: Distribución (%) de píxeles en las tres clases.

	Global Accuracy	Mean Accuracy	MeanIoU	Weighted IoU	Mean BF Score
SegNet	0.97	0.98	0.95	0.95	0.82
Unet	0.94	0.96	0.90	0.89	0.72

Figura 5.12: Métricas globales de ambas redes.

	SegNet			Unet		
	Accuracy	IoU	Mean BF Score	Accuracy	IoU	Mean BF Score
Fondo	0.99	0.98	0.99	0.99	0.98	0.99
No-Cerebro	0.97	0.96	0.86	0.91	0.90	0.75
Cerebro	0.98	0.92	0.61	0.98	0.81	0.42

Figura 5.13: Métricas por clase de ambas redes.

Unet	Fondo	No cerebro	cerebro
Fondo	3179500	179	8
No-cerebro	45515	14388000	1448400
cerebro	1320	156680	7220500

Figura 5.14: Matriz de confusión de UNET.

segnet	Fondo	No cerebro	cerebro
Fondo	3173700	5860	89
No-cerebro	43272	15348000	490110
cerebro	1677	137620	7239200

Figura 5.15: Matriz de confusión de SegNet.

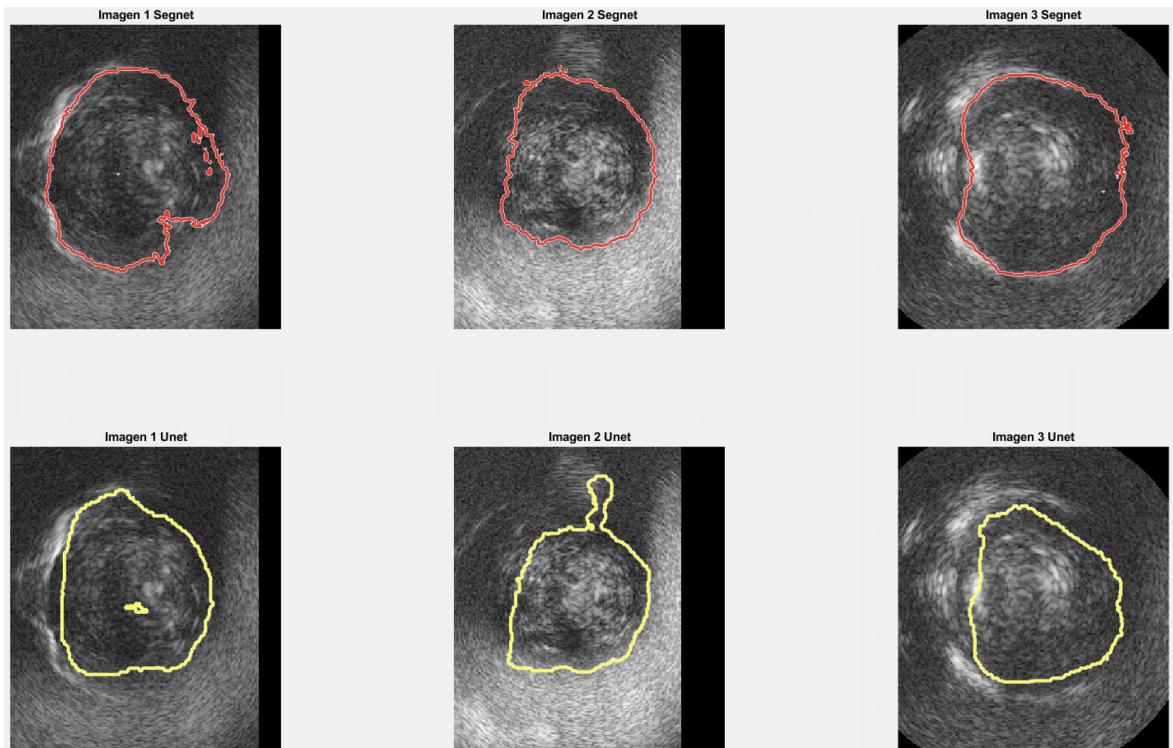


Figura 5.16: Comparación de las segmentaciones realizadas por ambas redes en tres imágenes distintas de un mismo volumen. En amarillo la segmentación Unet, en rojo la segmentación Segnet

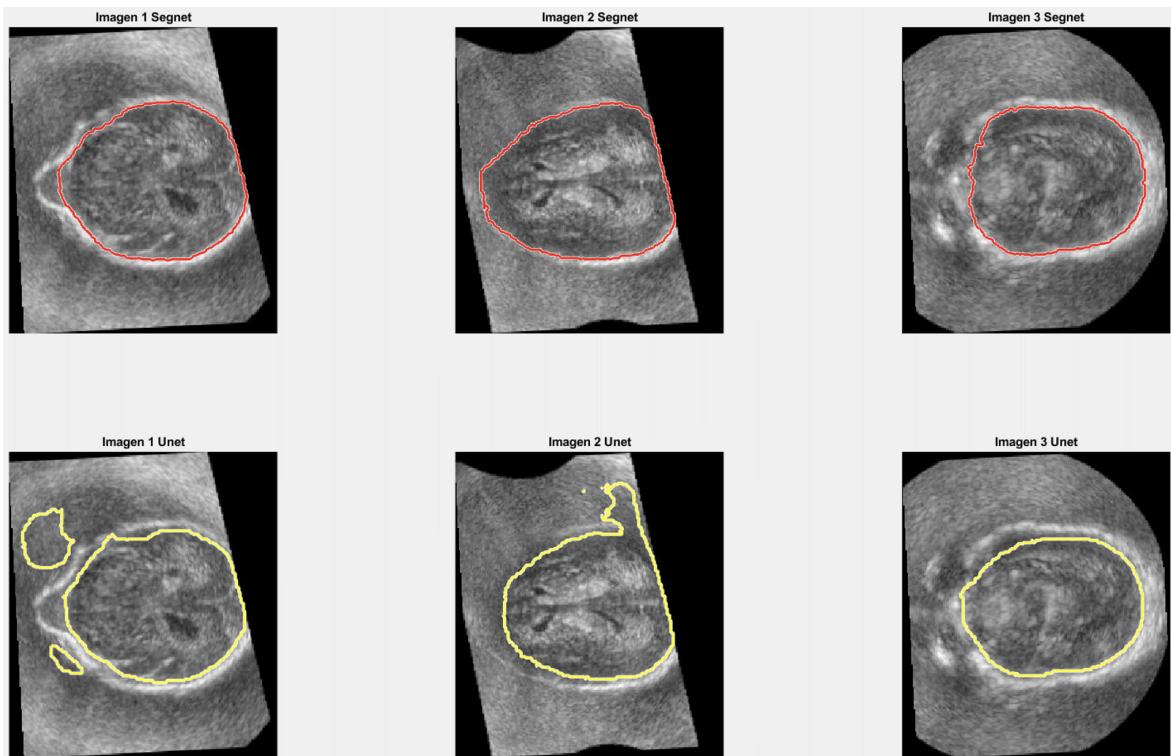


Figura 5.17: Comparación de las segmentaciones realizadas por ambas redes en tres imágenes distintas de un mismo volumen. En amarillo la segmentación Unet, en rojo la segmentación Segnet

### 5.5.5. Mejora de la red SegNet

Viendo que la red SegNet era superior a la Unet, decidimos continuar con la investigación basandonos en la red SegNet tratando de mejorar sus resultados.

Después de realizar varias pruebas, vimos que la red solía equivocarse más a la hora de segmentar los extremos, ya que hemos realizado el entrenamiento con las imágenes de la parte central del volumen cerebral.

Nuestro software de segmentación semiautomática 5.5.1 estimaba que el cráneo terminaba antes de lo previsto. Entonces solo segmentamos los planos intermedios del cerebro y entrenamos a la red con dichos planos.

Sin embargo a la hora de segmentar los volúmenes, a la red, le pasábamos todos los planos. Nuestra red cometía errores sobre todo en los extremos, ya que de esta parte no tenía información en la etapa de entrenamiento. Esperábamos que la red aprendiera a segmentar los extremos de forma automática pero en muchos casos la segmentación era errónea o parcial.

Utilizando los volúmenes segmentados por la primera red, retocamos manualmente los extremos necesarios con nuestro software, y entrenamos una nueva red SegNet preentrenada con VGG16. Incrementando los datos de entrenamiento de 6992 a 9322 imágenes. La red resultante mejoraba ligeramente las métricas y observamos que segmentaciones en los extremos eran más precisas. Esta es la red que usaremos para el cálculo del VCT.

### 5.5.6. Cálculo del VCT

Nuestras redes neuronales solo son capaces de segmentar una imagen 2D. La unión de todas las segmentaciones de cada slice en un volumen forman el cerebro. Podemos calcular su volumen como la suma de todos los voxels que lo forman multiplicado por el tamaño del voxel que viene especificado en la información de cada ecografía.

Tendremos que diseñar un algoritmo que se encargue de segmentar un volumen completo utilizando las redes actuales.

Este algoritmo lo hemos llamado *segmentaVCT*. Recibe como entrada el volumen que deseamos segmentar y la red neuronal entrenada que queremos que realice la tarea. Y como salida devuelve el volumen segmentado.

- 1. Aplica un suavizado de la mediana al volumen completo y redimensiona el volumen a imágenes de 200x200, que es el tamaño que hemos utilizado para segmentar la red.
- 2. Plano a plano segmenta el volumen completo.
- 3. Redimensiona el volumen al tamaño de entrada original.
- 4. Dada una ecografía 3D, el volumen cerebral VCT queda determinado como la unión de los slices segmentados. Como puede contener errores, por ejemplo

varias zonas segmentadas por separado, nos quedaremos con la de mayor volumen usando regionsprop3.

### 5.5.7. Más métodos para el cálculo del VCT

Como vimos en la sección 2.1.1 muchas ecografías están cortadas. Esto es debido a que cuando el bebé crece el cerebro aumenta de tamaño y es posible que se salga del tamaño máximo del haz que genera la ecografía y por tanto se obtiene un volumen parcial del cerebro. Podemos ver un ejemplo en la figura 2.3.

Para solucionar este problema, se realizaron distintas pruebas de cálculos para aproximar un elipsoide a partir del volumen segmentado, esperando que este elipsoide sea capaz de aproximar la parte que falta. Los métodos fueron:

- **1.V\_cnn:** Volumen calculado por la red SegNet.
- **2.V1\_elipses:** Volumen interpolando elipses en cada slice del plano axial.
- **3.E1\_elipsoide:** Volumen interponiendo un elipsoide a partir de V1.
- **4.V13\_elipses:** Volumen interpolando elipses en cada slice del plano axial y coronal.
- **5.E13\_elipsoide:** Volumen interponiendo un elipsoide a partir de V13.
- **6.V123\_elipses:** Volumen interpolando elipses en cada slice de todos los planos, axial, sagital y coronal.
- **7.E123\_elipsoide:** Volumen interponiendo un elipsoide a partir de V123.

## **Parte III**

### **Epílogo**

# 6. Resultados y discusión

---

En este capítulo se recogerán los resultados del cálculo del volumen cerebral total utilizando los distintos métodos y se realizará una discusión sobre cual es el más fiable y sobre si podría utilizarse como método sustitutivo al manual o a la resonancia magnética que es el “gold standard” actual.

El método que tomaremos como referencia para realizar las comparaciones es el VOCAL. Para ello contamos con las medidas de 155 volúmenes de TEST, completamente nuevos para realizar las pruebas.

## 6.1. Comparativa de los distintos métodos para el cálculo del VCT

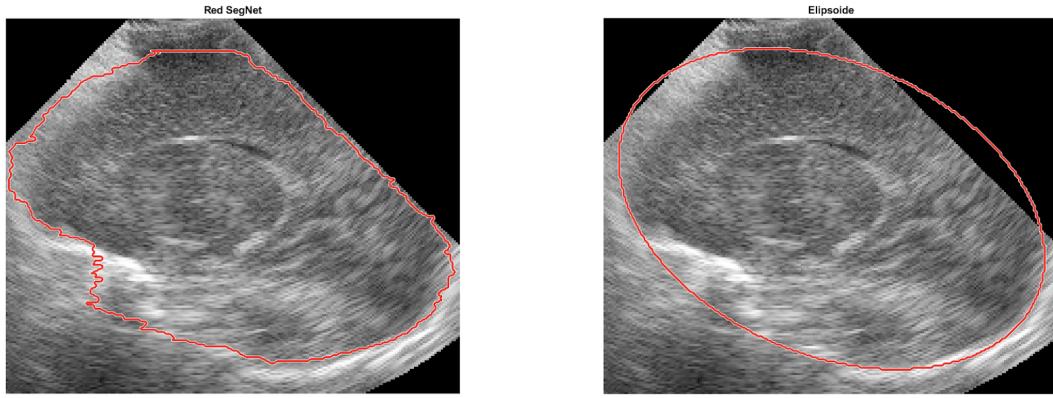
En esta sección veremos los resultados de aplicar los distintos métodos vistos en la sección 5.5.7 a 145 volúmenes distintos. Los volúmenes utilizados han sido reservados para el conjunto de TEST, por lo que son totalmente nuevos para la red. La tabla con los resultados se puede ver en el anexo A.1.

En la tabla del anexo A.1 podemos ver la media de los errores absolutos y la desviación estándar. Según estas dos mediciones, el mejor método es estimar un elipsoide a partir de las elipses segmentadas en cada uno de los planos (7.E123 elipsoide). Este método nos deja con una media de error absoluto de  $11,65cm^3$  y una desviación estándar de  $7,96cm^3$ , la segmentación que realiza la red nos da una media de error absoluto de  $17,79cm^3$  y una desviación estándar de  $15,04cm^3$ . Por lo que podemos concluir que mejora la segmentación inicial.

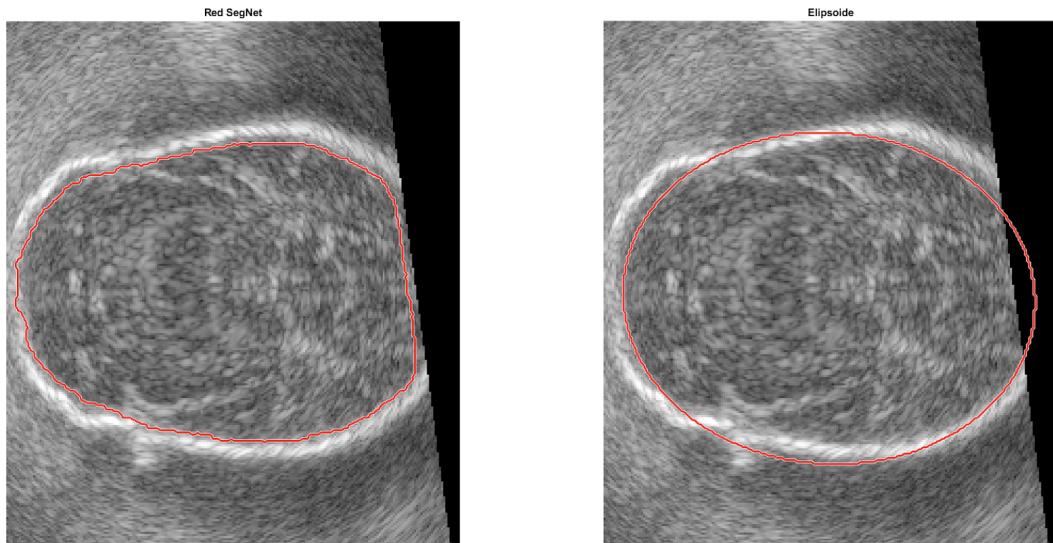
Sin embargo, si estudiamos los datos, podemos ver como la exactitud de cada método depende del volumen que se analice. Esto es debido a la gran variabilidad de calidad que existen en estas imágenes. Para los volúmenes en los que la red realiza un buen trabajo acercándose bastante a la segmentación manual, el método de estimación con un elipsoide empeora el resultado, ya que parece que “estropea” el buen trabajo realizado por la red.

Teniendo en cuenta que la media del tamaño de los cerebros es de 214,33, nos da un error medio del 0,05 %, para el método de la elipsoide. Lo que lo convierte en un método bastante fiable. Además, si vemos las tendencias de las mediciones en la figura 6.3, la tendencia es muy similar a la de las medidas manuales.

El método de estimación de una elipsoide a partir de las elipses segmentadas logra obtener el resultado esperado, consiguiendo minimizar el error en volúmenes más grandes, completando los volúmenes en los que el cerebro está cortado en la ecografía como podemos ver en la figura 6.1.



(a) Segmentación SegNet



(b) Elipsoide Aproximado.

Figura 6.1: Ejemplo del método de la segmentación ajustando elipses en cada plano a partir de la segmentación de SegNet y calculando un elipsoide. Vemos como el elipsoide consigue llenar la parte del cerebro que falta.

Los gráficos resultantes tienen formas de dientes de sierra debido a que disponemos de distintos volúmenes pertenecientes a un mismo bebé por lo que podemos ver como el cerebro va creciendo en el tiempo. En la figura 6.2 podemos ver la tendencia del crecimiento del cerebro de un bebé para el método manual y el mejor método.

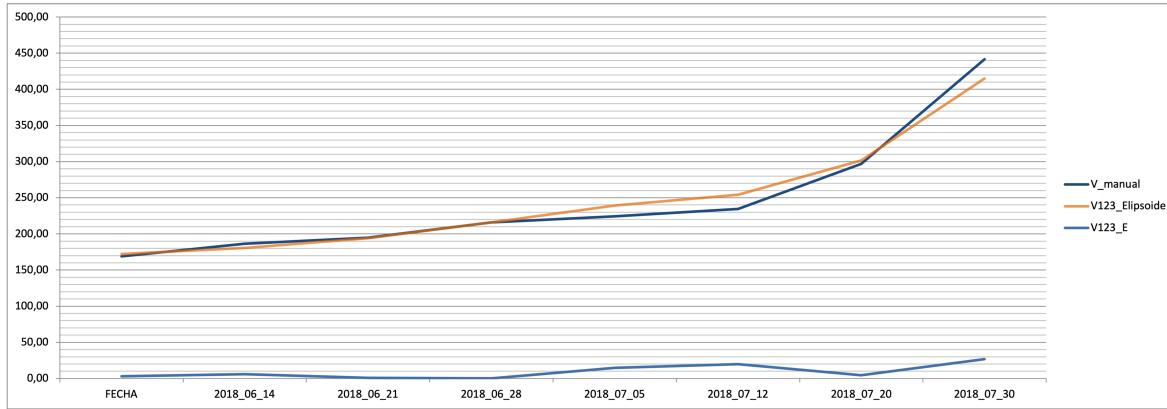


Figura 6.2: Tendencia del crecimiento del cerebro de un bebé, utilizando el método 7.E123\_elipsoide. En el eje Y  $cm^3$ . En el eje X la fecha en la que se realizó la ecografía. En azul oscuro la medición manual. En naranja el resultado de la segmentación utilizando una elipsoide a partir de las elipses estimadas en cada plano. En azul claro el error absoluto.

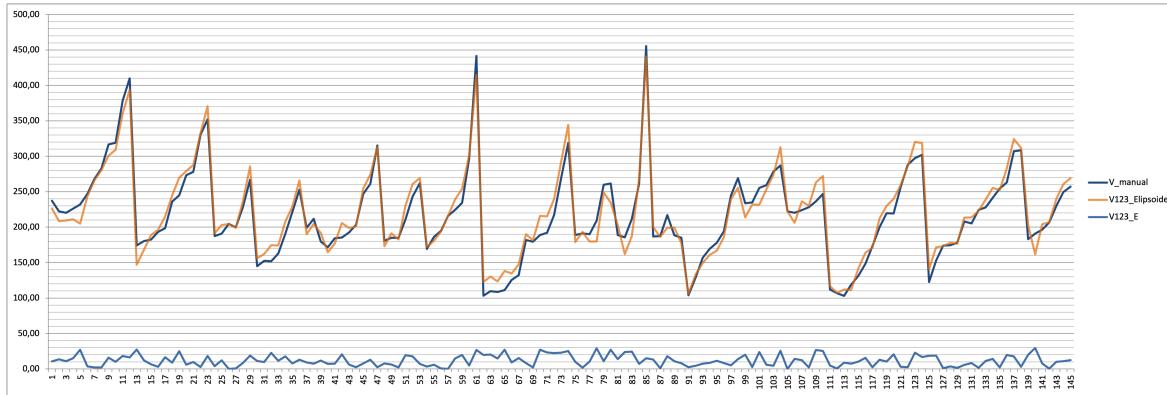


Figura 6.3: Comparación del método 7.E123\_elipsoide con las mediciones obtenidas con el método VOCAL. En el eje Y  $cm^3$ . En el eje X los índices de cada volumen. En azul oscuro la medición manual. En naranja el resultado de la segmentación utilizando una elipsoide a partir de las elipses estimadas en cada plano. En azul claro el error absoluto.

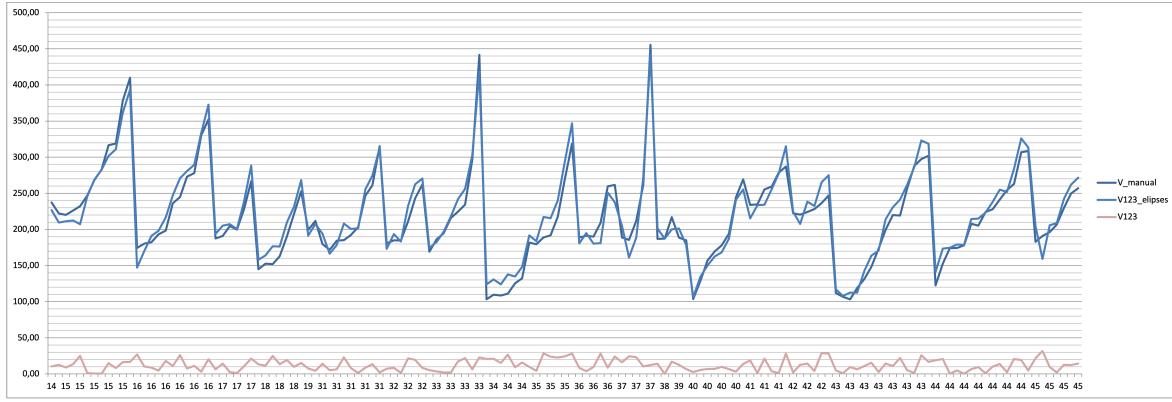


Figura 6.4: Comparación del método 6.V123\_elipses con las mediciones obtenidas con el método VOCAL. En el eje Y  $\text{cm}^3$ . En el eje X los índices de cada volumen. En azul oscuro la medición manual. En azul claro el resultado de la segmentación elipses estimadas en cada plano. En rosa el error absoluto.

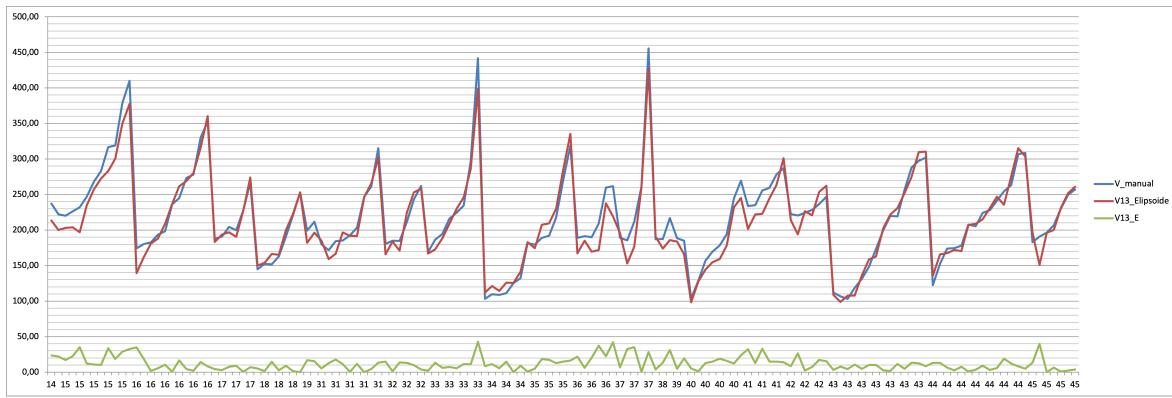


Figura 6.5: Comparación del método 5.E13\_elipsoide con las mediciones obtenidas con el método VOCAL. En el eje Y  $\text{cm}^3$ . En el eje X los índices de cada volumen. En azul oscuro la medición manual en  $\text{cm}^3$ . En rojo el resultado de la segmentación utilizando una elipsoide a partir de las elipses estimadas en el plano axial y coronal. En verde el error absoluto.



# 7. Conclusiones

---

## 7.1. Objetivos alcanzados

El objetivo general de este proyecto era realizar un estudio sobre los distintos algoritmos de segmentación tanto clásicos como en el estado del arte del área de Deep Learning. Pero también, si era posible, desarrollar un software que realizara el cálculo automático de los volúmenes cerebrales.

Este objetivo general final se ha conseguido. Como hemos visto en los resultados a través de la red neuronal SegNet, junto con nuestro método de segmentación estimando elipses en cada plano se consiguen resultados bastante buenos. Que permiten realizar estudios del crecimiento de un volúmen cerebral a largo plazo y de forma automatizada.

Para llegar a este objetivo general, se han cumplido los distintos objetivos específicos propuestos, ya que se ha ido construyendo este software poco a poco y cada objetivo era dependiente del anterior.

Se ha realizado un estudio de las distintas técnicas, se ha desarrollado un software que permite retocar las segmentaciones automáticas propuestas, se han estudiado distintas redes neuronales para la segmentación de imágenes en el estado del arte y finalmente se ha conseguido desarrollar un software de segmentación automática fiable.

Además, como vimos en el capítulo 6, se pueden realizar estudios del crecimiento del volumen cerebral de un bebé dibujando gráficas similares a la que encontramos en la figura 6.2.

Finalmente, se ha mejorado el software de segmentación para que realice la segmentación inicial propuesta con la red SegNet entrenada, sustituyendo al algoritmo basado en Random Walks. De esta forma un especialista puede modificar la segmentación si fuera necesario y, por tanto, el valor del VCT.

Con este software de segmentación automática de volúmenes cerebrales se tarda en segmentar un volumen nuevo 1' 30''. El método VOCAL requiere de personal experto y para realizar una segmentación se necesitan unos 10 minutos.

## 7.2. Lecciones aprendidas

Son muchas las horas que he dedicado a este proyecto, pero también las lecciones que he aprendido. Estoy seguro de que si tuviera que volver a abordar este mismo problema lo haría de una forma totalmente distinta, y creo que esa es una muy buena señal de la cantidad de errores que he cometido durante el desarrollo del proyecto y que ya no volveré a cometer.

La primera clave en este tipo de proyectos es la organización. No solo del código, ni de los documentos o bibliografía, sino también de las ideas. Llevar un diario de trabajo de que se ha realizado y que se podría mejorar, tareas pendientes, puede suponer una gran mejora en la eficacia a la hora de retomar el trabajo al día siguiente.

Sobre aprendizajes tecnológicos me llevo mucho. Hasta ahora, había trabajado muy poco en el procesamiento computacional de imágenes. Solo tenía unas nociones básicas y unos conocimientos algo oxidados. Gracias a repasar los algoritmos de segmentación clásicos y el preprocesamiento de imágenes, he obtenido una muy buena base de conocimiento que me sirve para comprender como funcionan algoritmos más complejos y en el estado del arte. Nunca había estudiado las redes neuronales convolucionales y he podido ver de primera mano su potencial en el campo de la visión computacional. Son capaces de resolver problemas de una gran complejidad en muy poco tiempo, adquiriendo conciencia de la relevancia que tienen en el futuro tecnológico cercano y el impacto que pueden tener en la sociedad.

Además de estos aprendizajes tecnológicos tan importantes, me llevo algunos conocimientos sobre problemas y enfermedades que no conocemos y que afectan, o podrían afectar, a tantos recién nacidos en el mundo. Tener estos conocimientos no solo es útil para trabajar en este tipo de problemas, sino que también te forma como persona.

### 7.3. Trabajo futuro

Este proyecto es una muestra más del gran avance que suponen las redes neuronales convolucionales en el campo de la visión computacional. Y abre una ventana a la posibilidad de obtener segmentaciones 3D a partir de ecografías que podrían aplicarse a otros campos. Además, si se quisiera perfeccionar el cálculo del VCT se me ocurren varias ideas:

- Podrían realizarse segmentaciones manuales en los tres planos y, en vez de depender solo del plano axial para el entrenamiento de la red, entrenar a las distintas redes utilizando los tres planos y comprobar si se obtienen mejores resultados.
- Se podrían buscar otros métodos para el cálculo del VCT que no estén basados en la segmentación manual de volúmenes para el entrenamiento, ya que el etiquetado de cantidades tan grandes de datos requiere de mucho tiempo. Por ejemplo, se podría entrenar a una red que calculase de forma automática algunas medidas de distancia entre partes del cerebro, entre los ventrículos, del ventrículo al cráneo, etc. A partir de estas medidas se podría tratar de determinar, mediante una ecuación, el VCT.
- Se podría buscar un mejor método a la hora de situar las elipses en los planos, buscando que logren representar mejor la parte que falta del cerebro.

Como vimos en la introducción este proyecto es una pequeña parte de AUTONENE. Sería muy interesante integrar este software en el programa principal y seguir desarrollando las distintas áreas del mismo.





# Bibliografía

---

- [1] “Ecografía transfontanelar en el lactante.” <https://www.familiaysalud.es/medicinas/pruebas-complementarias/ecografia-transfontanelar-en-el-lactante> (Última visita: 2022/09/03), 2022.
- [2] L. Grady and G. Funka-Lea, “Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials,” in *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, pp. 230–245, Springer, 2004.
- [3] F. G. Palomares, J. A. Monsoriu, and E. Alemany, “Aplicación de la convolución de matrices al filtrado de imágenes,” *Modelling in Science Education and Learning*, vol. 9, no. 1, pp. 97–108, 2016.
- [4] K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” *arXiv preprint arXiv:1511.08458*, 2015.
- [5] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015.
- [6] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [7] A. R. Synnes, S. Anson, A. Arkesteijn, A. Butt, R. E. Grunau, M. Rogers, and M. F. Whitfield, “School entry age outcomes for infants with birth weight 800 grams,” *The Journal of pediatrics*, vol. 157, no. 6, pp. 989–994, 2010.
- [8] L. D. Breeman, J. Jaekel, N. Baumann, P. Bartmann, and D. Wolke, “Preterm cognitive function into adulthood,” *Pediatrics*, vol. 136, no. 3, pp. 415–423, 2015.
- [9] N. Marlow, D. Wolke, M. A. Bracewell, and M. Samara, “Neurologic and developmental disability at six years of age after extremely preterm birth,” *New England journal of medicine*, vol. 352, no. 1, pp. 9–19, 2005.
- [10] S. Liu, Y. Wang, X. Yang, B. Lei, L. Liu, S. X. Li, D. Ni, and T. Wang, “Deep learning in medical ultrasound analysis: A review,” *Engineering*, vol. 5, no. 2, pp. 261–275, 2019.
- [11] M. J. Brouwer, L. S. De Vries, L. Pistorius, K. J. Rademaker, F. Groenendaal, and M. J. Binders, “Ultrasound measurements of the lateral ventricles in neonates: why, how and when? a systematic review,” *Acta paediatrica*, vol. 99, no. 9, pp. 1298–1306, 2010.
- [12] M. Riccabona, “Potential role of 3dus in infants and children,” *Pediatric radiology*, vol. 41, no. 1, pp. 228–237, 2011.

- [13] I. Timor-Tritsch, A. Monteagudo, and P. Mayberry, “Three-dimensional ultrasound evaluation of the fetal brain: the three horn view,” *Ultrasound in Obstetrics and Gynecology: The Official Journal of the International Society of Ultrasound in Obstetrics and Gynecology*, vol. 16, no. 4, pp. 302–306, 2000.
- [14] L. Pogliani, G. Zuccotti, M. Furlanetto, V. Giudici, A. Erbetta, L. Chiapparini, and L. Valentini, “Cranial ultrasound is a reliable first step imaging in children with suspected craniosynostosis,” *Child’s Nervous System*, vol. 33, no. 9, pp. 1545–1552, 2017.
- [15] P. Gupta, K. S. Sodhi, A. K. Saxena, N. Khandelwal, and P. Singhi, “Neonatal cranial sonography: a concise review for clinicians,” *Journal of pediatric neurosciences*, vol. 11, no. 1, p. 7, 2016.
- [16] I. Benavente-Fernández, E. Ruiz-González, M. Lubian-Gutiérrez, S. P. Lubián-Fernández, Y. Cabrales Fontela, C. Roca-Cornejo, P. Olmo-Duran, and S. P. Lubián-López, “Ultrasonographic estimation of total brain volume: 3d reliability and 2d estimation. enabling routine estimation during nicu admission in the preterm infant,” *Frontiers in Pediatrics*, p. 740, 2021.
- [17] J. P. Kusanovic, J. K. Nien, L. F. Gonçalves, J. Espinoza, W. Lee, M. Balasubramaniam, E. Soto, O. Erez, and R. Romero, “The use of inversion mode and 3d manual segmentation in volume measurement of fetal fluid-filled structures: comparison with virtual organ computer-aided analysis (vocal™),” *Ultrasound in obstetrics & gynecology*, vol. 31, no. 2, pp. 177–186, 2008.
- [18] “General Electric HealthCare.” <https://www.gehealthcare.com/about/spin-off?showPopup=false> (Última visita: 2022/08/10), 2022.
- [19] M.-A. Boucher, S. Lippé, C. Dupont, I. S. Knoth, G. Lopez, R. Shams, R. El-Jalbout, A. Damphousse, and S. Kadoury, “Computer-aided lateral ventricular and brain volume measurements in 3d ultrasound for assessing growth trajectories in newborns and neonates,” *Physics in Medicine & Biology*, vol. 63, p. 225012, nov 2018.
- [20] F. Milletari, S.-A. Ahmadi, C. Kroll, A. Plate, V. Rozanski, J. Maiostre, J. Levin, O. Dietrich, B. Ertl-Wagner, K. Bötzel, and N. Navab, “Hough-CNN: Deep learning for segmentation of deep brain regions in MRI and ultrasound,” *Computer Vision and Image Understanding*, vol. 164, pp. 92–102, 2017.
- [21] R. D. Smet, “Automatic analysis of cerebral sulci in ultrasound images,” Master’s thesis, Universidad de Gante, Bélgica, 2019.
- [22] R. Khaled, A. Ortega-Leon, J. Pizarro, I. B. Fernández, S. P. L. López, and L. C. Gontard, “Automatic Neonatal Cranial Ultrasound Segmentation using Deep Learning: A Review,” p. 16.
- [23] “What is SCRUM?.” <https://www.scrum.org/resources/what-is-scrum> (Última visita: 2022/08/11), 2022.

- [24] K. Schwaber and J. Sutherland, *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*, 11 2020. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100>.
- [25] “What is matlab?.” <https://uk.mathworks.com/products/matlab.html>(Última visita: 2022/08/11), 2022.
- [26] “Matlab documentation.” <https://uk.mathworks.com/help/>(Última visita: 2022/08/11), 2022.
- [27] “Sueldo del ingeniero informático en españa.” <https://es.talent.com/convert?salary=36000&start=year&end=hour>(Última visita: 2022/08/12), 2022.
- [28] “Archivo de tfgs del grado de ingeniería informática de la universidad de cádiz.” <https://rodin.uca.es/handle/10498/17922>(Última visita: 2022/08/12), 2022.
- [29] R. G. Dantas, E. T. Costa, and S. Leeman, “Ultrasound speckle and equivalent scatterers,” *Ultrasonics*, vol. 43, no. 6, pp. 405–420, 2005.
- [30] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, “Multiscale vessel enhancement filtering,” in *International conference on medical image computing and computer-assisted intervention*, pp. 130–137, Springer, 1998.
- [31] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “Slic superpixels compared to state-of-the-art superpixel methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.
- [32] “superpixel.” <https://uk.mathworks.com/help/images/ref/superpixels.html>(Última visita: 2022/08/17), 2022.
- [33] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014.
- [34] “Regionprops3.” <https://uk.mathworks.com/help/images/ref/regionprops3.html> (Última visita: 2022/08/22), 2022.
- [35] “MATLAB App Designer.” <https://uk.mathworks.com/products/matlab/app-designer.html> (Última visita: 2022/08/22), 2022.



## **Parte IV**

### **Anexos**

# A. Tablas de Resultado

---

En este anexo recogeremos las tablas de Excel con todos los resultados obtenidos.

## A.1. Tabla comparativa de los distintos métodos para el cálculo del VCT

En esta tabla podemos ver los resultados de comparar distintos métodos para el cálculo del VCT. A partir de la segmentación realizada por la red neuronal, se han desarrollado distintos métodos para la estimación del VCT.

- **1.V\_cnn:** Volumen calculado por la red SegNet
- **2.V1\_elipses:** Volumen interpolando elipses en cada slice del plano axial.
- **3.E1\_elipsoide:** Volumen interponiendo un elipsoide a partir de V1.
- **4.V13\_elipses:** Volumen interpolando elipses en cada slice del plano axial y coronal.
- **5.E13\_elipsoide:** Volumen interponiendo un elipsoide a partir de V13.
- **6.V123\_elipses:** Volumen interpolando elipses en cada slice de todos los planos, axial, sagital y coronal
- **7.E123\_elipsoide:** Volumen interponiendo un elipsoide a partir de V123.

En esta tabla contamos con los resultados de aplicar cada método a 145 volúmenes distintos, completamente nuevos usados para el test. Los errores absolutos de estos respecto de la medición manual. La media y la desviación estándar de los errores.

N	ID	FECHA	V_manual	Volumenes								Errores							
				V_cnn	V1_elipses	V1_Ellipsoide	V13_elipses	V13_Ellipsoide	V123_elipses	V123_Ellipsoide	Vcnn	V1	V1_E	V13	V13_E	V123	V123_E		
29	14	2018-04-16	237,22	200,77	202,94	204,07	213,55	213,52	226,76	226,51	36,45	34,28	33,15	23,67	23,70	10,46	10,71		
31	15	2018-04-13	222,20	186,65	187,88	188,68	199,84	199,99	209,55	208,60	35,54	34,32	33,51	22,36	22,23	12,64	13,59		
32	15	2018-04-16	220,23	194,87	194,75	193,96	205,04	203,11	211,24	209,44	25,36	25,48	26,28	15,19	17,12	8,99	10,79		
33	15	2018-04-17	226,10	196,16	196,20	196,56	204,47	203,77	212,50	211,04	29,95	29,90	29,54	21,63	22,34	13,61	15,06		
34	15	2018-04-19	232,09	190,40	190,89	189,99	198,31	196,74	207,16	204,95	41,68	41,20	42,10	33,78	35,35	24,93	27,13		
35	15	2018-05-02	247,02	225,27	224,46	223,55	237,39	234,77	245,81	243,50	21,75	22,56	23,47	9,63	12,25	1,21	3,53		
36	15	2018-05-09	268,12	245,19	245,07	244,21	260,69	257,43	268,91	265,94	22,93	23,05	23,91	7,43	10,69	0,79	2,18		
37	15	2018-05-16	282,86	261,36	262,11	261,90	274,29	272,61	282,40	280,71	21,50	20,75	20,95	8,56	10,24	0,45	2,14		
38	15	2018-05-23	316,68	273,11	272,99	275,85	283,16	283,02	301,59	300,84	43,57	43,69	40,83	33,52	33,66	15,09	15,84		
39	15	2018-05-30	318,97	289,17	289,27	289,39	302,58	300,24	310,93	308,93	29,81	29,70	29,58	16,39	18,73	8,04	10,04		
40	15	2018-06-13	378,31	337,76	336,47	336,03	352,00	349,62	361,86	359,99	40,55	41,83	42,28	26,30	28,69	16,45	18,32		
41	15	2018-06-21	409,98	348,64	352,22	359,21	372,81	377,49	393,36	393,94	61,33	57,76	50,76	37,17	32,49	16,61	16,04		
42	16	2018-04-13	174,25	125,90	129,50	131,74	137,69	139,36	147,18	146,89	48,34	44,74	42,50	36,56	34,88	27,07	27,35		
43	16	2018-04-16	180,37	154,26	154,28	154,36	162,30	161,85	170,02	168,17	26,11	26,09	26,01	18,07	18,52	10,35	12,20		
44	16	2018-04-17	182,50	172,57	173,21	172,89	181,80	180,83	191,10	188,98	9,93	9,29	9,61	0,70	1,67	8,60	6,48		
45	16	2018-04-19	193,34	179,76	180,00	180,27	188,33	187,82	198,08	196,26	13,57	13,34	13,07	5,00	5,51	4,75	2,92		
46	16	2018-05-02	198,43	196,95	198,15	198,50	210,35	208,79	216,60	214,90	1,48	0,28	0,07	11,92	10,36	18,17	16,46		
47	16	2018-05-09	236,15	225,56	224,87	224,58	238,77	236,69	247,05	244,89	10,59	11,27	11,57	2,62	0,54	10,90	8,74		
48	16	2018-05-16	244,76	244,45	245,84	247,86	261,69	261,31	270,79	269,80	0,31	1,08	3,10	16,93	16,55	26,03	25,04		
49	16	2018-05-24	273,23	254,12	254,75	255,82	269,28	268,87	280,86	279,07	19,11	18,48	17,41	3,97	4,36	7,64	5,85		
50	16	2018-05-30	277,98	267,33	267,63	268,03	281,47	279,91	289,28	287,67	10,65	10,35	9,95	3,49	1,93	11,30	9,69		
51	16	2018-06-13	330,32	295,43	295,76	303,71	311,93	316,19	333,67	332,89	34,89	34,57	26,61	18,39	14,14	3,35	2,56		
52	16	2018-06-21	352,30	342,43	342,19	343,69	362,66	360,32	372,68	370,62	9,87	10,11	8,60	10,37	8,02	20,39	18,33		
53	17	2018-04-16	187,45	177,56	177,32	176,51	185,27	183,16	193,89	191,40	9,89	10,13	10,94	2,18	4,29	6,44	3,95		
54	17	2018-04-17	190,81	186,11	186,93	185,40	195,73	193,67	205,25	202,90	4,70	3,88	5,41	4,92	2,86	14,44	12,09		
55	17	2018-04-18	204,55	192,13	191,66	190,11	198,84	196,86	207,25	204,49	12,42	12,90	14,45	5,71	7,69	2,70	0,06		
56	17	2018-04-19	199,68	180,86	182,03	182,76	191,89	190,53	200,80	199,19	18,83	17,65	16,93	7,80	9,15	1,11	0,49		
57	17	2018-05-03	227,74	218,86	218,58	217,38	229,78	227,16	235,50	236,00	8,88	9,16	10,36	2,04	0,58	10,76	8,26		
58	17	2018-05-10	266,85	261,85	262,49	261,86	275,52	273,97	288,12	285,65	5,00	4,36	4,99	8,67	7,11	21,27	18,80		
59	18	2018-04-16	144,84	144,28	145,26	144,76	151,16	149,99	157,94	156,10	0,55	0,43	0,07	6,33	5,15	13,11	11,26		
60	18	2018-04-17	152,51	147,70	147,82	147,20	155,18	153,98	164,04	162,09	4,81	4,68	5,31	2,68	1,47	11,54	9,59		
61	18	2018-04-18	151,91	158,42	159,62	158,17	168,51	166,48	176,78	174,38	6,52	7,72	6,27	16,61	14,57	24,88	22,47		
62	18	2018-04-19	162,77	158,63	159,02	158,26	166,79	165,27	176,34	174,19	4,13	3,75	4,50	4,02	2,50	13,58	11,43		
63	18	2018-05-03	190,48	191,48	191,21	190,29	201,24	199,46	209,72	208,00	1,00	0,73	0,19	10,76	8,98	19,24	17,52		
64	18	2018-05-10	223,39	215,78	215,07	214,06	225,28	222,90	231,19	229,10	5,62	6,32	7,33	3,89	1,51	9,80	7,71		
65	18	2018-05-17	252,88	238,33	239,85	240,27	253,88	253,02	268,20	265,92	14,56	13,04	12,63	0,99	0,13	15,31	13,03		
66	19	2018-05-16	199,14	174,98	176,32	175,26	183,57	182,15	191,54	190,00	24,16	22,82	23,88	15,57	16,99	7,60	9,14		
67	19	2018-05-30	211,79	190,80	190,72	188,92	199,46	196,41	207,42	204,55	21,00	21,07	22,88	12,33	15,38	4,37	7,24		
98	31	2018-05-18	179,91	178,38	178,60	176,86	187,21	185,10	194,00	191,67	1,52	1,30	3,04	7,30	5,19	14,09	11,76		
99	31	2018-05-21	171,66	154,37	153,85	152,59	160,45	158,98	166,41	164,45	17,29	17,81	19,07	11,21	12,68	5,25	7,21		
100	31	2018-05-22	184,50	159,84	159,12	160,77	166,78	166,63	178,26	177,16	24,65	25,38	23,72	17,72	17,87	6,24	7,33		
101	31	2018-05-25	185,31	184,92	184,13	183,09	198,42	196,56	208,23	205,95	0,39	1,18	2,22	13,11	11,25	22,92	20,64		
102	31	2018-05-28	192,41	186,31	185,28	185,58	193,66	191,92	200,62	198,73	6,10	7,13	6,83	1,25	0,49	8,21	6,32		
103	31	2018-06-01	203,15	179,95	180,07	182,54	190,92	191,58	201,54	200,80	23,20	23,08	20,61	12,23	11,57	1,61	2,35		
104	31	2018-06-13	246,80	234,05	234,44	234,38	248,56	246,71	255,51	254,41	12,75	12,37	12,42	1,76	0,10	8,70	7,60		
105	31	2018-06-19	261,07	251,61	252,67	253,45	265,91	265,42	274,56	274,08	9,46	8,40	7,62	4,84	4,35	13,49	13,01		
106	31	2018-07-04	315,23	286,66	287,91	291,29	301,39	301,72	313,24	312,87	28,56	27,31	23,94	13,83	13,50	1,98	2,36		
107	32	2018-05-23	180,57	155,84	158,09	158,25	165,79	165,76	173,35	172,99	24,73	22,48	22,32	14,83	14,81	7,22	7,58		
108	32	2018-05-24	185,02	176,13	177,26	175,27	185,96	183,76	193,50	191,16	8,89	7,76	9,75	0,94	1,26	8,48	6,14		
109	32	2018-05-25	184,64	158,54	159,31	161,26	170,15	170,87	183,25	182,58	26,10	25,34	23,38	14,50	13,77	1,39	2,06		
110	32	2018-06-13	211,20	215,99	216,21	215,41	226,23	224,22	232,62	230,47	4,79	5,01	4,21	15,03	13,03	21,43	19,28		
111	32	2018-06-21	243,00	239,13	239,66	239,15	254,81	252,91	262,58	260,64	3,87	3,33	3,85	11,82	9,91	19,58	17,65		
112	32	2018-07-02	262,10	246,61	246,33	248,72	259,61	258,21	270,42	269,27	15,49	15,77	13,38	2,49	3,89	8,31	7,17		
113	33	2018-06-14	169,07	161,68	161,04	159,89	169,00	166,86	174,43	172,22	7,43	8,04	8,18	0,07	2,21	5,36	3,15		
114	33	2018-06-21	186,32	161,58	162,13	162,18	174,71	172,82	182,71	180,52	24,75	24,19	24,15	11,62	13,51	3,61	5,80		
115	33	2018-06-28																	

116	33	2018.07.05	215,93	199,92	199,87	198,82	210,49	208,68	217,89	215,88	16,01	16,06	17,11	5,44	7,26	1,95	0,05
117	33	2018.07.12	224,48	214,21	214,78	215,08	232,59	230,05	241,71	239,12	10,27	9,69	9,39	8,11	5,58	17,24	14,64
118	33	2018.07.20	234,41	232,20	233,71	233,93	248,01	245,92	256,31	254,12	2,22	0,71	0,48	13,59	11,51	21,90	19,70
119	33	2018.07.30	296,85	265,54	266,69	271,55	288,33	285,72	303,24	301,49	31,31	30,15	25,30	8,52	11,13	6,40	4,64
120	33	2018.09.18	441,68	378,94	380,85	383,04	401,11	399,06	418,73	414,91	62,74	60,83	58,64	40,57	42,62	22,94	26,76
121	34	2018.07.02	103,35	104,89	105,66	105,71	112,54	111,91	123,90	122,90	1,54	2,31	2,35	9,19	8,56	20,55	19,54
122	34	2018.07.03	109,73	116,55	117,40	117,32	122,23	121,23	130,82	130,09	6,82	7,67	7,59	12,50	11,50	21,09	20,36
123	34	2018.07.04	108,66	107,19	107,63	108,56	114,28	114,21	124,02	123,50	1,47	1,03	0,10	5,62	5,55	15,36	14,84
124	34	2018.07.06	111,09	115,24	115,30	116,20	125,96	125,84	137,70	137,99	4,15	4,22	5,12	14,87	14,76	26,62	26,90
125	34	2018.07.13	125,48	117,63	117,92	118,39	126,48	125,56	134,78	134,52	7,84	7,56	7,09	1,00	0,08	9,30	9,05
126	34	2018.07.20	132,40	133,38	133,67	135,06	141,49	141,35	148,14	147,58	0,98	1,27	2,66	9,09	8,95	15,74	15,18
130	35	2018.07.06	181,75	179,29	178,24	178,02	184,19	182,78	191,69	190,00	2,46	3,52	3,73	2,44	1,02	9,94	8,24
131	35	2018.07.09	179,43	168,93	169,04	167,57	176,18	174,49	183,66	181,39	10,49	10,39	11,85	3,24	4,93	4,24	1,97
132	35	2018.07.13	188,97	196,01	196,84	196,53	208,81	207,36	217,26	215,86	7,05	7,87	7,57	19,84	18,40	28,29	26,89
133	35	2018.07.19	191,95	198,61	198,90	201,53	209,97	209,50	215,69	215,27	6,66	6,94	9,58	18,02	23,73	23,32	
134	35	2018.07.30	217,43	211,78	214,54	216,31	230,68	230,12	240,05	239,77	5,64	2,88	1,12	13,25	12,70	22,62	22,35
135	35	2018.08.20	269,49	271,28	272,62	272,68	285,16	284,28	293,82	292,49	1,79	3,13	3,20	15,68	14,79	24,33	23,01
136	35	2018.09.18	318,83	321,04	321,96	321,94	337,07	335,23	346,90	344,19	2,21	3,12	3,10	18,24	16,40	28,06	25,35
137	36	2018.07.09	189,00	159,15	160,03	160,73	168,31	167,16	180,51	178,86	29,85	28,97	28,27	20,69	21,84	8,49	10,14
138	36	2018.07.12	191,24	172,91	177,07	176,53	186,50	185,01	194,93	193,16	18,33	14,17	14,71	4,74	6,23	3,68	1,92
139	36	2018.07.13	190,01	157,74	157,72	160,84	169,01	169,52	180,41	179,63	32,27	32,29	29,17	21,00	20,49	9,60	10,38
140	36	2018.07.16	209,08	158,01	160,53	165,03	172,63	171,81	180,94	179,91	51,04	48,52	44,02	36,42	37,24	28,11	29,14
144	36	2018.07.30	259,77	220,89	223,48	226,28	239,02	237,49	251,00	248,82	38,88	38,88	39,29	20,75	22,28	8,77	10,94
145	36	2018.07.31	261,63	200,99	200,65	212,02	226,35	219,37	237,63	234,56	60,64	60,98	49,61	35,28	42,26	24,00	27,07
146	37	2018.07.13	188,99	183,72	184,44	183,55	197,10	195,58	205,02	202,84	5,27	4,55	5,44	8,11	6,60	16,04	13,85
147	37	2018.07.16	185,56	136,88	142,51	147,58	150,04	152,85	161,13	161,75	48,68	43,05	37,98	35,52	32,71	24,43	23,81
149	37	2018.07.31	211,79	169,82	173,09	173,66	178,40	176,64	188,58	187,44	41,97	38,70	38,13	33,39	35,15	23,21	24,35
150	37	2018.08.19	261,28	251,52	251,06	250,16	263,36	260,75	271,52	268,50	9,77	10,23	11,12	2,08	0,54	10,24	7,21
151	37	2018.10.02	455,51	398,80	400,09	403,19	427,73	427,40	443,00	440,51	56,71	55,42	52,32	27,78	28,11	12,51	15,00
152	38	2018.07.13	186,94	180,77	181,19	181,94	191,97	190,63	201,18	200,10	6,18	5,75	5,00	5,02	3,69	14,24	13,16
153	38	2018.07.16	187,35	163,68	167,74	169,54	173,74	173,94	187,25	186,31	23,67	19,61	17,81	13,61	13,41	0,10	1,04
154	38	2018.07.23	216,91	171,76	172,21	176,91	185,70	185,79	199,97	199,10	45,14	44,70	40,00	31,21	31,12	16,94	17,81
155	39	2018.07.13	188,56	178,33	178,39	181,12	184,40	183,79	201,33	199,46	10,23	10,17	7,44	4,16	4,77	12,77	10,90
156	39	2018.07.16	185,05	154,30	156,65	158,16	166,23	165,67	178,40	177,08	30,75	28,40	26,89	18,82	19,38	6,65	7,97
157	40	2018.07.31	103,80	90,23	91,73	93,86	97,78	98,59	106,49	106,13	13,57	12,06	9,93	6,02	5,21	2,69	2,33
158	40	2018.08.20	128,96	120,51	120,34	120,31	128,44	127,72	134,59	133,45	8,44	8,61	8,64	0,52	1,24	5,63	4,49
159	40	2018.08.30	156,84	135,28	135,43	136,29	143,98	144,13	150,02	149,51	21,56	21,41	20,55	12,86	12,71	6,82	7,33
160	40	2018.09.05	169,57	147,97	147,41	147,58	155,41	154,64	162,49	160,91	21,60	22,16	21,99	14,16	14,94	7,08	8,66
161	40	2018.09.12	178,15	152,69	152,23	151,95	160,36	159,13	168,26	166,77	25,46	25,92	26,20	17,79	19,02	9,89	11,38
162	40	2018.09.17	194,09	168,07	169,15	170,00	178,79	178,05	187,29	185,90	26,02	24,94	24,08	15,30	16,03	6,79	8,18
164	40	2018.10.05	244,42	220,70	220,15	220,37	233,90	232,04	241,13	239,43	23,72	24,27	24,05	10,52	12,38	3,29	4,99
165	40	2018.10.15	269,26	228,84	230,21	232,59	243,61	245,28	255,44	255,32	40,42	39,04	36,67	25,65	23,97	13,82	13,94
166	41	2018.08.19	233,84	188,19	190,54	192,90	203,00	201,42	215,30	213,73	45,65	43,30	40,94	30,84	32,42	18,54	20,11
167	41	2018.08.24	234,86	211,07	212,87	213,23	222,11	221,95	233,73	232,04	23,79	21,99	21,63	12,75	12,91	1,13	2,82
168	41	2018.08.27	255,48	214,09	214,56	213,66	224,10	222,65	234,30	231,77	41,39	40,92	41,82	31,38	32,83	21,18	23,71
169	41	2018.08.31	259,25	229,61	233,41	231,54	246,45	244,51	255,51	253,31	29,64	25,84	27,71	12,80	14,74	3,74	5,94
170	41	2018.09.10	278,41	254,12	256,48	255,22	266,04	263,49	277,14	273,86	24,28	21,93	23,18	12,36	14,92	1,26	4,54
171	41	2018.09.14	286,97	283,44	286,34	284,95	303,86	301,08	315,03	312,47	3,53	0,63	2,02	16,88	14,11	28,06	25,49
172	42	2018.08.19	222,40	203,22	207,89	206,35	215,77	213,90	224,53	222,42	19,19	14,52	16,05	6,63	8,51	2,13	0,01
173	42	2018.08.24	220,32	186,14	188,41	187,71	194,88	193,95	207,66	206,06	34,17	31,91	32,60	25,44	26,37	12,66	14,26
174	42	2018.08.27	224,11	211,65	215,53	214,16	228,03	226,58	238,38	236,49	12,47	8,58	9,95	3,92	2,47	14,26	12,37
175	42	2018.08.31	228,15	211,57	213,87	212,62	222,34	220,71	232,33	230,21	16,58	14,28	15,53	5,81	7,44	4,18	2,06
176	42	2018.09.10	236,37	234,15	236,09	236,11	255,42	253,60	265,45	263,01	2,22	0,28	0,26	19,05	17,23	29,08	26,64
177	42	2018.09.14	246,98	253,74	253,36	251,35	264,95	262,30	275,07	272,10	6,77	6,38	4,37	17,97	15,32	28,09	25,12
178	43	2018.08.24	112,04	102,25	102,79	103,53	108,71	108,71	116,98	116,74	9,78	9,25	8,51	3,33	3,33	4,94	4,70
179	43	2018.08.27	106,86	92,48	92,80	93,05	99,52	99,02	107,93	107,49	14,43	14,06	13,81	7,34	7,85	1,06	0,63
180	43	2018.08.29	103,13	101,80	102,18	102,69	108,17	107,60	112,64	111,91	1,33	0,95	0,44	5,04	4,47	9,51	8,78

182	43	2018	09	12	131,52	127,39	127,59	128,70	136,80	136,31	142,47	141,91	4,13	3,93	2,82	5,28	4,80	10,95	10,39
183	43	2018	09	19	148,38	149,04	149,55	152,51	157,80	158,62	163,80	163,92	0,66	1,17	4,23	9,42	10,24	15,42	15,54
184	43	2018	09	25	173,22	139,21	140,74	148,03	160,66	162,90	170,54	170,90	34,01	32,47	25,19	12,55	10,32	2,67	2,31
185	43	2018	10	05	199,59	190,98	190,80	190,97	203,80	202,25	214,07	212,39	8,61	8,79	8,62	4,22	2,67	14,49	12,80
186	43	2018	10	11	219,78	211,22	211,09	211,45	222,90	221,64	230,74	229,98	8,56	8,69	8,34	3,12	1,86	10,95	10,19
187	43	2018	10	18	219,26	220,11	220,42	220,47	232,34	230,78	241,22	239,92	0,85	1,16	1,21	13,08	11,52	21,96	20,66
188	43	2018	10	24	256,54	240,64	240,95	240,17	254,24	251,80	261,74	259,39	15,90	15,58	16,37	2,29	4,74	5,20	2,85
189	43	2018	10	31	288,02	258,16	258,52	261,44	275,57	274,67	286,60	285,63	29,85	29,50	26,57	12,45	13,34	1,42	2,39
190	43	2018	11	11	297,34	296,03	296,62	296,17	312,34	309,82	323,11	320,18	1,31	0,71	1,17	15,01	12,48	25,77	22,85
191	43	2018	11	16	301,98	287,97	290,58	299,31	308,77	310,35	318,63	318,63	14,01	11,39	2,67	6,79	8,38	16,66	16,66
192	44	2018	08	29	122,49	129,77	129,79	130,80	136,48	135,73	141,52	140,87	7,28	7,30	8,31	13,99	13,24	19,03	18,38
193	44	2018	09	05	152,74	159,72	159,42	158,71	167,32	165,73	173,38	171,61	6,98	6,68	5,97	14,58	12,99	20,65	18,87
194	44	2018	09	12	174,04	160,05	159,88	159,07	168,79	167,53	174,50	173,02	13,98	14,15	14,96	5,25	6,50	0,46	1,02
195	44	2018	09	17	174,36	162,91	162,73	162,53	173,47	171,74	179,24	177,96	11,45	11,63	11,85	0,88	2,61	4,89	3,61
196	44	2018	09	20	178,10	163,72	164,16	163,32	171,94	170,50	178,52	176,66	14,37	13,94	14,77	6,16	7,60	0,43	1,43
197	44	2018	09	25	207,77	199,38	198,89	199,51	208,05	206,99	214,53	213,38	8,39	8,88	8,26	0,28	0,78	6,76	5,61
198	44	2018	10	01	205,31	201,48	200,51	200,60	210,39	208,71	214,85	213,64	3,82	4,80	4,70	5,08	3,40	9,54	8,33
199	44	2018	10	05	224,29	204,74	205,11	206,05	215,67	215,23	223,29	222,73	19,55	19,18	18,24	8,62	9,06	1,00	1,56
200	44	2018	10	10	227,86	216,22	216,75	220,20	230,20	230,97	237,86	238,93	11,64	11,11	7,66	2,34	3,11	10,01	11,07
201	44	2018	10	18	241,29	234,39	234,10	236,82	247,71	247,02	255,07	255,33	6,90	7,19	4,47	6,42	5,73	13,78	14,04
202	44	2018	10	24	254,41	213,87	216,12	224,40	228,83	235,44	252,04	252,20	40,54	38,30	30,02	25,58	18,97	2,37	2,21
203	44	2018	10	31	263,13	261,60	261,31	262,84	277,16	275,49	283,83	282,84	1,53	1,82	0,29	14,03	12,36	20,70	19,71
204	44	2018	11	11	306,90	297,45	297,34	298,85	317,61	315,07	326,01	324,45	9,45	9,57	8,05	10,70	8,16	19,10	17,55
205	44	2018	11	16	308,58	286,46	286,03	287,69	305,84	303,53	313,30	311,65	22,12	22,55	20,89	2,73	5,05	4,72	3,07
206	45	2018	09	14	182,91	189,14	189,18	189,32	197,87	196,41	204,80	202,84	6,23	6,27	6,41	14,96	13,49	21,89	19,93
207	45	2018	09	17	190,88	131,00	134,24	144,28	147,90	150,88	159,24	161,50	59,88	56,65	46,61	42,99	40,00	31,64	29,39
208	45	2018	09	21	196,35	188,31	188,63	188,27	197,44	196,24	205,88	204,06	8,04	7,72	8,08	1,09	0,10	9,53	7,71
209	45	2018	09	27	206,72	185,36	188,85	189,68	200,86	200,18	208,66	206,95	21,36	17,86	17,04	5,85	6,54	1,95	0,23
210	45	2018	10	05	229,87	221,66	222,85	221,52	232,59	230,77	242,52	239,94	8,21	7,03	8,35	2,71	0,90	12,64	10,07
211	45	2018	10	11	249,58	236,88	240,02	241,58	250,99	251,79	262,05	260,55	12,70	9,55	8,00	1,41	2,21	12,47	10,97
212	45	2018	10	15	257,03	254,16	253,51	253,13	262,49	260,80	271,51	269,29	2,87	3,52	3,90	5,46	3,77	14,48	12,26
										MEDIA	17,79	17,11	16,36	12,30	12,17	12,14	11,65		
										DESVIACION	15,04	14,49	13,29	9,76	9,86	8,18	7,96		