



ugr | Universidad
de Granada

TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA

Clasificación automática de criterios morfológicos
para estimación de la edad a partir de modelos 3D
de la sínfisis del pubis

Autor

Valentino Glauco Lugli

Directores

Sergio Damas Arroyo
Pablo Mesejo Santiago



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

—
Granada, Septiembre de 2022

Clasificación automática de criterios morfológicos para estimación de la edad a partir de modelos 3D de la sínfisis del pubis

Valentino Glauco Lugli

Palabras clave: aprendizaje automático, aprendizaje profundo, visión por computador, antropología forense, estimación del perfil biológico, estimación de edad, clasificación, malla 3D.

Resumen

La estimación de la edad es una de las tareas más importantes en la antropología forense, formando parte del perfil biológico, contribuyendo a la identificación de personas vivas o muertas cuando no es posible o práctico otro tipo de técnicas (como podrían ser las huellas dactilares o el ADN). Se utiliza en casos de personas desparecidas, crisis migratorias, guerras, catástrofes naturales o crímenes sin resolver. Para lograr este objetivo, se utiliza comúnmente el hueso de la sínfisis del pubis, al que se le aplica el método de Thomas Win-gate Todd propuesto en 1921. Consiste en el análisis visual de la superficie de la sínfisis para identificar 9 características, altamente subjetivas, por lo que la estimación correcta de la edad depende mucho de la interpretación personal del experto. En este sentido, cabe mencionar que existe una falta de automatización y de técnicas objetivas en esta área de estudio.

Este TFG trata del desarrollo de un sistema capaz de clasificar automáticamente las características presentes en un modelo 3D de la sínfisis del pubis, para asistir en la estimación de la edad de una persona fallecida. La propuesta presenta un enfoque totalmente novedoso y nunca antes visto: un sistema que identifica y clasifica las características haciendo uso de un modelo de aprendizaje profundo para procesar directamente las mallas poligonales 3D de la sínfisis del pubis. De las 9 características del método de Todd, se estudió la conocida como “nódulo óseo” por recomendación de los expertos, debido a su facilidad de detección y relevancia. Para los experimentos, se utilizan 98 mallas 3D de la sínfisis del pubis, 49 con presencia y 49 ausencia de nódulo. Aún siendo de un conjunto de datos muy reducido, se logró clasificar por medio de una red neuronal convolucional la existencia del “nódulo óseo” en los datos de test con un 70 % de *accuracy* y una métrica F1 de 0.7, mostrando el gran potencial que posee de esta línea de investigación.

Automatic classification of morphological criteria for age estimation using 3D scans of the pubic symphysis

Valentino Glauco Lugli

Keywords: machine learning, deep learning, computer vision, forensic anthropology, biological profile estimation, age estimation, classification, 3D mesh

Abstract

Age estimation is one of the most important tasks in forensic anthropology, being part of the biological profile, which helps to identify either alive or deceased individuals when it's not possible or practical to use other techniques (such as fingerprints or DNA). It's used in missing person cases, migratory crises, wars, natural disasters or unsolved crimes. To this end, the pubic symphysis bone is used by applying a method first proposed by Thomas Wingate Todd in 1921. The pubic symphysis is analyzed visually to identify 9 highly subjective characteristics on its surface, which means that a correct age estimate depends highly on the personal interpretation of the forensic expert. There is a lack of automation and objective techniques in this field of study.

This Bachelor's Thesis proposes a system capable of automatic classification of the characteristics present in a 3D model of the pubic symphysis, to assist in the age estimation process of a deceased individual. This proposal presents a new and never-before-seen approach: a system that can identify and classify the characteristics using a deep learning model to directly process the 3D meshes of the pubic symphysis. From the 9 characteristics used in Todd's Method, the forensic experts recommended the study of the "bony nodule" characteristic since it's easy to detect and also quite relevant. For the experiments, a total of 98 3D meshes of pubic symphyses are used, 49 have the bony nodule present and 49 don't. Although it's a small sample size, a convolutional neural network was able to classify the existence of the nodule with 70% accuracy and a F1 metric of 0.7, showing that this line of research has great potential.

Yo, **Valentino Glauco Lugli**, alumno de la titulación Grado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con pasaporte YB0819879, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Valentino Glauco Lugli

Granada a 6 de septiembre de 2022.

D. **Sergio Damas Arroyo**, Profesor del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada.

D. **Pablo Mesejo Santiago**, Profesor del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado *Clasificación automática de criterios morfológicos para estimación de la edad a partir de modelos 3D de la sínfisis del pubis*, ha sido realizado bajo su supervisión por **Valentino Glauco Lugli**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 6 de septiembre de 2022.

Los directores:

Sergio Damas Arroyo

Pablo Mesejo Santiago

Agradecimientos

En primer lugar, quiero agradecer a mis tutores, Sergio Damas y Pablo Mesejo, por darme la oportunidad de trabajar en esta línea de investigación tan importante y fascinante. Agradezco mucho su comprensión y paciencia infinita al resolver las dudas y problemas que me surgieron en el transcurso de este trabajo, así como también su muy buen trato, sumamente amigable y alentador, que permitió la realización de un trabajo satisfactorio y novedoso.

Quiero agradecer a mis padres, Elisa Ventresca y Glauco Lugli, quienes han hecho un esfuerzo enorme para que yo lograse finalizar mis estudios en España, siempre apoyándome de cualquier forma y motivándome para seguir adelante con mis estudios. Agradezco a todos mis amigos, tanto de España como de Venezuela, quienes me han sido de grata compañía durante el transcurso de mi carrera.

Hago también una mención muy especial a la futura doctora veterinaria María Belén Corredor, quien me ha acompañado desde hace años, coincidiendo con el inicio de mis estudios universitarios; siempre apoyándome, escuchándome, regañándome para que no perdiera la esperanza y siguiera adelante con todos los retos que tuve que enfrentarme, tanto fuera como dentro de la universidad. También ha tenido la valiosa oportunidad de leer esta obra cuando estaba en edición y ofrecer un punto de vista externo para el entendimiento de conceptos tan avanzados, así como múltiples correcciones de forma y fondo. Te agradezco por esto y miles de cosas más.

Finalmente, agradezco a la propia Universidad de Granada por haberme dado la oportunidad de continuar con mis estudios universitarios y poderlos finalizar en tan distinguida casa de estudios.

Índice general

1. Introducción	7
1.1. Definición del Problema	7
1.2. Motivación	10
1.3. Objetivos	11
1.4. Planificación del proyecto	12
2. Fundamentos Teóricos	15
2.1. Aprendizaje Automático	15
2.2. Aprendizaje Profundo	16
2.2.1. Redes Neuronales Convolucionales	19
2.2.2. Capa convolucional	19
2.2.3. Capa de <i>pooling</i>	21
2.2.4. Capa totalmente conectada	22
2.2.5. Regularización	22
2.3. Representaciones 3D en <i>Deep Learning</i>	24
2.3.1. Datos en bruto	25
2.3.2. Sólidos	26
2.3.3. Estructuras de alto nivel	27
2.3.4. Superficies	28
2.3.5. Múltiples vistas	29
3. Estado del Arte	31
3.1. Estimación de la edad	31
3.1.1. Métodos tradicionales para restos óseos	32
3.1.2. Métodos tradicionales utilizando la síntesis del pubis .	33
3.1.3. Métodos automáticos utilizando la síntesis del pubis .	34
3.2. Representaciones 3D en <i>Deep Learning</i>	36
3.2.1. Mallas poligonales	37
4. Materiales y Métodos	39
4.1. Materiales	39
4.2. Métodos	42

4.2.1. MeshCNN	42
4.2.2. MedMeshCNN y MeshCNN+	47
5. Implementación y Experimentos	49
5.1. Detalles técnicos de la implementación	49
5.1.1. Entorno de desarrollo	49
5.1.2. Obtención de los modelos 3D	50
5.1.3. Preprocesado de los datos	50
5.1.4. Generación de <i>datasets</i>	53
5.1.5. Estadísticas sobre los datos	55
5.1.6. Entrenamiento del modelo	55
5.2. Experimentos	56
5.2.1. Protocolo de validación experimental	56
5.2.2. Métricas	57
5.2.3. Experimentos preliminares	58
5.2.4. Experimentos con el Nódulo Óseo	65
6. Conclusiones y Trabajos Futuros	79
7. Bibliografía	83
A. Nodule-98-30K: Parámetros y Resultados Completos	91

Índice de figuras

1.1.	Proceso de identificación forense a partir de restos óseos	8
2.1.	Ejemplos de la estructura de una ANN	17
2.2.	Ejemplo de una neurona artificial	18
2.3.	Estructura básica de una red neuronal convolucional	19
2.4.	Ejemplo del operador de convolución	20
2.5.	Ejemplo del operador de pooling	21
2.6.	Tipos de normalización	22
2.7.	Taxonomía de las representaciones 3D para Deep Learning . .	25
3.1.	Publicaciones por año de AF, AF+IA y AF+IA+3D en Scopus	32
4.1.	Distribución de los datos por cada rango de edad	40
4.2.	Distribución de los datos por cada característica del método de Todd	42
4.3.	Síntesis del pubis con presencia de nódulo óseo.	43
4.4.	Síntesis del pubis con ausencia de nódulo óseo.	43
4.5.	MeshCNN: Vecindario local de una arista	44
4.6.	MeshCNN: Características de una arista	45
4.7.	MeshCNN: Ejemplo visual de pooling	46
5.1.	Estructura del archivo de datos	51
5.2.	Detalle del operador Booleano de Blender	52
5.3.	Resultado del sellado del modelo 3D	53
5.4.	Proceso de sellado de la malla 3D de la síntesis del pubis . .	54
5.5.	Esquema de división de datos	56
5.6.	SHREC16: Curvas de aprendizaje	59
5.7.	SphereCubes: Colapso de aristas	61
5.8.	SphereCubes: Mapas de activación	63
5.9.	Nodule-50-30K-5: Curvas de aprendizaje	67
5.10.	Nodule-98-30K-5rev: Curvas de aprendizaje	68
5.11.	Nodule-98-30K-GS-25: Curvas de aprendizaje	71
5.12.	Orientación, mapas de activación	72
5.13.	Nódulo óseo presente, mapas de activación	74

5.14. Nódulo óseo presente mal clasificado, mapas de activación	75
5.15. Nódulo óseo ausente, mapas de activación	76
5.16. Nódulo óseo ausente mal clasificado, mapas de activación	77

Índice de tablas

1.1.	Método de Todd: Características para determinación de edad	9
1.2.	Método de Todd: Ejemplo de características	9
1.3.	Planificación temporal inicial del proyecto	13
1.4.	Planificación temporal final del proyecto	13
1.5.	Estimación de coste del proyecto	14
3.1.	Rangos de edad del Método de Todd	33
3.2.	Rangos de edad del Método de Suchey-Brooks	34
5.1.	SHREC16: Hiperparámetros	59
5.2.	SHREC16: Aumento de datos	59
5.3.	SHREC16: Arquitectura detallada	60
5.4.	SHREC16: Arquitectura resumida	61
5.5.	SphereCubes, Arquitectura	61
5.6.	Distancia Hausdorff entre las mallas originales y las mallas reducidas	64
5.7.	Resoluciones de pooling para cada framework	65
5.8.	Comparativa de frameworks respecto a tiempo y espacio	65
5.9.	Nodule-50-30K-*: Arquitectura	66
5.10.	Nodule-50-30K-*: Resultados en test	66
5.11.	Nodule-50-30K-5: Matriz de confusión	67
5.12.	Nodule-98-30K-GS-*: Hiperparámetros	68
5.13.	Nodule-98-30K-GS-*: Arquitecturas	69
5.14.	Nodule-98-30K-GS-*: Aumento de datos	69
5.15.	Nodule-98-30K-GS-*: Mejores hiperparámetros	70
5.16.	Nodule-98-30K-GS-*: Mejores resultados en test	70
5.17.	Nodule-98-30K-GS-2, 7, 16: Matriz de confusión	71
5.18.	Nodule-98-30K-GS-25: Matriz de confusión	71
A.1.	Nodule-98-30K-GS-*: Todos los hiperparámetros	91
A.2.	Nodule-98-30K-GS-*: Todos los resultados de test	92

Capítulo 1

Introducción

El presente Trabajo de Fin de Grado (TFG) se ocupa de resolver una problemática real dentro del campo de la identificación humana. En concreto, se hace uso de métodos de aprendizaje automático y visión por computador para automatizar una técnica de antropología forense empleada en tareas de estimación del perfil biológico, más específicamente, la estimación de la edad de personas fallecidas a partir de restos óseos.

Este capítulo introductorio se centra en presentar el problema en detalle, la motivación que nos lleva a enfrentarnos a él y el objetivo principal abordado en este TFG.

1.1. Definición del Problema

La antropología forense (AF) se puede definir como el área de conocimiento enfocado al análisis basado en información ósea de restos humanos con objetivos médico-legales [1, 2]. Los expertos en esta disciplina, con sus conocimientos de antropología física¹ y ramas afines, examinan dichos restos óseos para extraer la mayor información posible e identificar la persona asociada a esos restos, así como las circunstancias de su muerte, para ser aplicadas en un marco legal.

Esta área de conocimiento tiene 5 objetivos principales:

1. Determinar la ascendencia y características morfológicas de la persona fallecida.
2. Identificar las circunstancias y manera en que murió una persona.

¹La antropología física estudia el proceso evolutivo de la especie humana, así como las condiciones de vida y salud de poblaciones pasadas y actuales por medio de estudios osteológicos y somatológicos, información que complementa al considerar aspectos sociales, culturales y de comportamiento [3].

3. Determinar el tiempo que lleva fallecida la persona.
4. Asistir y recuperar restos superficiales o enterrados relevantes a la investigación forense.
5. Proveer información útil para la identificación de personas fallecidas, gracias a que existen características morfológicas presentes en los huesos de todos los humanos.
6. Estudio del esqueleto de personas vivas con motivos médico-legales, por ejemplo, para la identificación de migrantes o menores desaparecidos.

La estimación del perfil biológico (PB) es una de las áreas clave de AF y se centra principalmente en la determinación de la edad, el sexo, la altura, el origen poblacional o ascendencia, así como cualquier otra característica que permita individualizar los restos total o parcialmente esqueletizados. Esto permite la búsqueda de la identidad de la persona desaparecida, y la eventual identificación positiva de la misma. Un esquema ilustrativo del proceso de identificación forense puede apreciarse en la Figura 1.1.

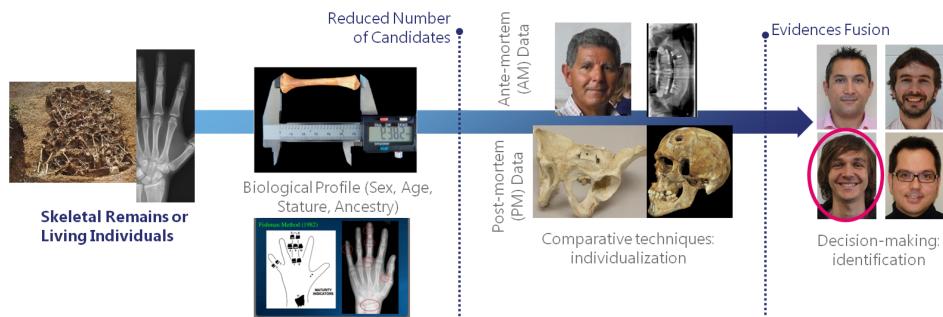


Figura 1.1: Proceso de identificación forense a partir de restos óseos [4].

Una de las tareas más importantes del PB es la determinación de la edad de los restos. Para la misma se utilizan las suturas craneales [5], las costillas [6], la cara auricular del ilion [7] y la síntesis del pubis, ambos ubicados en la pelvis. En cualquier caso, se analiza el desgaste que posee en hueso en el momento de la muerte para poder estimar la edad del individuo [8].

El hueso más común para la estimación de la edad es la síntesis del pubis, siendo preferido por un 95 % de los antropólogos forenses [9]. El método más usado se basa en el trabajo pionero de Thomas Wingate Todd [10], que describió en 1921 las alteraciones que ocurren en la síntesis con el paso del tiempo, y cómo éstas pueden ser utilizadas para estimar un rango de edad de la persona al morir. Todd propone usar un sistema de etapas o fases de envejecimiento, del cual se han realizado numerosas revisiones, la más conocida publicada en 1990 bajo el nombre de método de Suchey-Brooks

[11]. Todd propone estudiar nueve características de la sínfisis del pubis. En función del estado de cada una de ellas, se pueden asociar algunos atributos categóricos a las mismas basándose en el nivel de erosión del hueso en distintas partes, para así obtener un rango estimado de la edad del fallecido. Dichas características pueden observarse en la Tabla 1.1, con un ejemplo visual de cada característica en la Tabla 1.2.

Característica	Atributo	
<i>Crestas y Surcos</i>	Porosidad Regular	Muy Definidas
	Poco Profundas	Restos de Surcos
	No hay Surcos	
<i>Porosidad Irregular</i>	No	Mediana
	Sí	
<i>Borde Superior</i>	No Definido	Definido
<i>Nódulo Óseo</i>	Ausente	Presente
<i>Borde Inferior</i>	No Definido	Definido
<i>Borde Dorsal</i>	No Definido	Definido
<i>Plataforma Dorsal</i>	Ausente	Presente
<i>Bisel Ventral</i>	Ausente	En Formación
	Presente	
	Ausente	En Formación
<i>Borde Ventral</i>	Formado, Sin Excreencias	Formado, Pocas Excreencias
	Formado, Muchas Excreencias	

Tabla 1.1: Características utilizadas para la determinación de la edad según el método de Todd [10] y sus derivados.

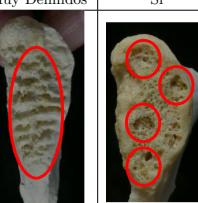
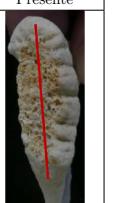
Característica	<i>Crestas y Surcos</i>	<i>Porosidad Irregular</i>	<i>Borde Superior</i>	<i>Nódulo Óseo</i>	<i>Borde Inferior</i>	<i>Borde Dorsal</i>	<i>Plataforma Dorsal</i>
Atributo	Muy Definidos	Sí	Definido	Presente	Definido	Definido	Presente
Ejemplo							

Tabla 1.2: Ejemplo de las características del método de Todd y sus derivados.

Cabe resaltar que la metodología utilizada en este método, así como gran parte de las metodologías en AF, depende mucho del criterio subjetivo del experto. Esto provoca, como consecuencia, la existencia de errores intraevaluador e interevaluador en la AF, pues el uso de criterios que son subjetivos y descriptivos siempre introducirá limitaciones por las diversas interpretaciones interevaluador [8]. Esto reduce la confianza y la validez de los resultados obtenidos, lo que finalmente hace perder credibilidad a los estudios forenses a la hora de ser presentados como evidencia en un juicio. Todo ello, justifica la búsqueda de herramientas y metodologías que permitan al menos reducir estas limitaciones. En este contexto, disciplinas científicas como la inteligen-

cia artificial (IA), y en concreto el aprendizaje automático (ML, *machine learning*) [12, 13], el aprendizaje profundo (DL, *deep learning*) [14, 15] y la visión por computador (CV, *computer vision*) pueden asistir, automatizar y acelerar las tareas forenses de manera que se puedan eliminar sesgos y errores.

Teniendo en cuenta todas estas consideraciones, el presente TFG aborda la clasificación automática de las características morfológicas de la sínfisis del pubis para estimar la edad a partir de modelos 3D utilizando técnicas de IA.

1.2. Motivación

En las últimas décadas, la IA ha permitido la automatización de tareas repetitivas o tediosas para los humanos, así como la superación de la capacidad humana en tareas complejas, particularmente en el área de ML y CV. En estas áreas ha habido grandes avances en lo que respecta a la detección, generación y restauración de imágenes [16]. Los beneficios que aportan estas técnicas han sido aprovechados por numerosas áreas, incluida la medicina, donde la IA ha provisto herramientas sumamente útiles para los expertos del área clínica. Por ello resulta sorprendente que hoy en día la AF siga presentando una baja sofisticación tecnológica [4]. Por lo tanto, una de las motivaciones de este trabajo es la modernización y automatización, desde el punto de vista tecnológico, del área de AF, y en esta obra en particular, las técnicas utilizadas para la estimación de edad.

También, y como se ha mencionado anteriormente, la subjetividad característica del área de AF es un punto de crítica a los ojos de la ley, pues no siempre se posee una base científica para los análisis dado el criterio de Daubert [17], que indica si es admisible el testimonio experto en un juicio. El criterio se satisface cuando (1) el resultado es reproducible y ha sido verificado por terceros, (2) posee tasas de error conocidas y (3) es aceptado por la comunidad científica forense. En este sentido, la aplicación de IA a AF permite reducir la subjetividad de las identificaciones y errores humanos, automatizar muchas tareas, permitir la obtención de conocimiento nuevo, y modelizar y estructurar el conocimiento experto humano. Lo que puede contribuir al cumplimiento del criterio de Daubert dado que ahora se proveería de una mayor base científica para los métodos que, además de ser reproducibles, se es capaz de conocer las tasas de error de los modelos entrenados.

Si nos centramos en el más amplio contexto de la identificación humana, la estimación del PB por medio de técnicas de AF cobra mayor importancia, dado que las herramientas existentes de mayor precisión y sofisticación (como pudieran ser el ADN y las huellas dactilares) presentan serias limita-

ciones. El ADN presenta un mayor coste y un tiempo de respuesta alto y al igual que las huellas dactilares, depende de la existencia de datos tanto antemortem como post-mortem. Junto con lo anterior, ambas técnicas dependen del estado de los tejidos blandos, que son justamente los más afectados por la descomposición tanto natural como artificial (daños por quemaduras, por agua, por químicos, etc.). Por lo tanto, estas técnicas no se pueden aplicar en muchos casos donde sí es posible aplicar las técnicas de AF. Esto es debido a que el tejido óseo es, en general, más resistente a los factores mencionados y, en muchos casos, es lo único que queda tras la descomposición total del tejido blando. Por ello, las técnicas basadas en AF resultan de mayor utilidad en:

- Identificación masiva de víctimas de desastres naturales, accidentes o ataques terroristas.
- Identificación de víctimas de guerras o actos de lesa humanidad, donde puede que los restos hayan sido desmembrados, desfigurados, quemados y/o mezclados.
- Individuos en fosas comunes donde los restos óseos puede que hayan sido mezclados entre sí.
- Identificación de personas desaparecidas no relacionadas con algún desastre o guerra en la que las condiciones del cadáver se han deteriorado al punto que no se pueden aplicar las otras técnicas mencionadas [1].

Para dar una escala al número de víctimas que se enfrentan los antropólogos forenses, solamente en el año 2019, 20 329 personas murieron por causas ligadas al terrorismo, teniendo además un promedio de 24 000 muertes por terrorismo anuales en la última década [18]. Así mismo, los desastres naturales acaban con 45 000 vidas de media al año [19]. En Ucrania, a la fecha de escritura de esta obra, se encuentran más de 1 600 cuerpos sin identificación, de los cuales una gran parte han sido quemados [20], y en España aún se tienen que recuperar alrededor de 20 000 víctimas de la Guerra Civil, que se encuentran en fosas y cunetas, donde apenas un tercio podrá identificarse por medio de ADN [21]. Estos datos sirven para ilustrar la necesidad y motivación del uso de la informática en este campo. Disponer de técnicas automatizadas supondría un ahorro en tiempo y dinero para poder detectar las características que permiten determinar la edad de las víctimas en situaciones donde el número de individuos a identificar es elevado y donde, además, no es posible aplicar otras técnicas.

1.3. Objetivos

Una vez descrito el problema y su motivación, el objetivo principal de este TFG es el desarrollo y validación de un modelo de aprendizaje profundo que

emplee modelos 3D de la sínfisis pélvica, de cara a extraer características morfológicas de dicho hueso, y contribuir a la automatización de la estimación de la edad en antropología forense.

Este objetivo principal se descompone en los siguientes objetivos parciales:

1. Estudio pormenorizado de la literatura relativa a la estimación de edad a partir de restos óseos, y al procesado de modelos 3D por medio de redes neuronales profundas.
2. Análisis y discusión de los modelos existentes, y selección razonada de los candidatos más prometedores para el problema actual.
3. Creación de un prototipo y validación experimental con modelos 3D de la sínfisis del pubis.
4. Extracción de una o varias de las más relevantes características para poder llevar a cabo la estimación de la edad.

1.4. Planificación del proyecto

El proyecto presente consiste, en esencia, en el diseño e implementación de un software de investigación. Dicho software posee unos requisitos y objetivos claros, por lo que no se espera que existan grandes cambios en su desarrollo. De cara a planificar el proyecto, resulta fundamental constatar que la asignatura del TFG consta de 12 créditos ECTS, siendo un crédito igual a 25 horas de trabajo. Esto implica que en total, se requieren aproximadamente 300 horas para la realización del mismo. Como el segundo cuatrimestre cuenta con 20 semanas aproximadamente, la realización del TFG requerirá 15 horas semanales (equivalente a 3 horas diarias, 5 días a la semana).

Tomando esto en cuenta, se observa que es un proyecto de complejidad pequeña a mediana respecto a los requisitos y objetivos, por lo que se opta por utilizar la metodología de desarrollo de software en cascada [22]. Esta metodología posee las fases de análisis, diseño, codificación, pruebas y mantenimiento. Dicho esto, el modelo en cascada rara vez se utiliza de forma estricta, pues implica el no poder retroceder a una fase anterior, lo que necesita un previo y absoluto conocimiento de los requisitos, la no volatilidad de los mismos y que las etapas subsiguientes no posean errores. Por eso se utiliza el modelo en cascada con retroalimentación, que permite volver a fases anteriores para realizar pequeños ajustes, sea por errores detectados, ambigüedades, o bien porque los propios requisitos hayan cambiado.

Las fases del ciclo de vida se adaptaron al proyecto de la siguiente forma:

- Análisis de Requisitos: Consistió en las reuniones iniciales con el cliente, que en este caso, serían los directores del TFG junto con los an-

tropólogos forenses. Se realiza también en esta fase una revisión bibliográfica extensa en el ámbito de la AF como la combinación de AF con técnicas automáticas dentro del área de la IA con la finalidad de establecer los objetivos del trabajo.

- Diseño: Consistió en la investigación y selección de las técnicas aplicables al problema, lo que incluye modelos, métricas, datos y protocolo de validación experimental. Aquí también se incluyen las diversas pruebas preliminares de los modelos.
- Implementación: Consintió en la adaptación del código de los modelos investigados, implementación de funcionalidades, así como la implementación de software de soporte en forma de scripts, tanto para la generación, el preprocesado de datos y obtención de diferentes métricas o estadísticas.
- Pruebas: Consiste mayoritariamente en la realización de diversos experimentos con los modelos y datos seleccionados.

Dicho esto, se puede observar la planificación inicial del proyecto en la Tabla 1.3, donde se mantuvo un mes adicional para posibles imprevistos o retrasos.

Tarea	Semanas - Horas	Febrero 21 28	Marzo 07 14 21 28	Abril 04 11 18 25	Mayo 02 09 16 23	Junio 30 06 13 20 27	Julio 04 11 18 25
Análisis de Requisitos	4 - 60						
Diseño	4 - 60						
Implementación	6 - 90						
Pruebas	6 - 90						

Tabla 1.3: Planificación temporal inicial del proyecto

Se realizó la planificación inicial con plazos relajados, tomando en cuenta que el autor estaba también realizando las Prácticas de Empresa, junto con estar cursando tres asignaturas. Aún así, ocurrieron retrasos significativos. Una causa fue el método seleccionado ya que, por una parte, es un método muy reciente por lo que la documentación del mismo es escasa y también porque el método utiliza librerías anteriormente desconocidas por el autor, lo que implicó más tiempo del planificado para estudiar y comprender su funcionamiento. Otra causa fueron los datos, pues ocurrió un retraso significativo en la obtención de los mismos, así como tiempo adicional que se utilizó para preprocesarlos, pues por diversas razones esto no se pudo automatizar completamente. Todo ello conllevó una modificación de la planificación, que puede observarse en la Tabla 1.4.

Tarea	Semanas - Horas	Febrero 21 28	Marzo 07 14 21 28	Abril 04 11 18 25	Mayo 02 09 16 23	Junio 30 06 13 20 27	Julio 04 11 18 25
Análisis de Requisitos	5 - 75						
Diseño	4 - 60						
Implementación	8 - 120						
Pruebas	5 - 75						

Tabla 1.4: Planificación temporal final del proyecto

Para el coste estimado, se asume un salario de 35€/hora para un responsable I+D de una empresa tecnológica o un investigador senior. Se añade también el costo de los materiales, de los que resalta: el coste del portátil utilizado para el desarrollo del TFG, el coste de dispositivos de almacenamiento masivo, el coste de usar un servidor GPU de altas prestaciones, y el coste acumulado de una suscripción a Google Colab Pro por la duración del proyecto, junto a otros gastos misceláneos. Se puede observar el desglose de costes en la Tabla 1.5.

Respecto al servidor GPU, se valúa con un coste de 15 000€. Se asume una amortización con 2 años de duración, lo que implica un pago diario de 20.55€, lo que se traduce en 3 164.70€ sobre la duración del proyecto. De la misma forma, la suscripción a Google Colab Pro, que cuesta 9.25€ al mes, se traduce a un coste de 55.50€ en total.

Fecha inicio	21/02/2022
Fecha fin	25/07/2022
Duración	154 días, 110 laborables

Item	Costo
Salario	11 550.00€
Portátil de Altas Prestaciones	800.00€
Google Colab Pro	55.50€
Servidor GPU	3 164.70€
Almacenamiento	150.00€
Otros	300.00€
Total	16 020.20€

Tabla 1.5: Estimación de coste del proyecto

Capítulo 2

Fundamentos Teóricos

Este capítulo tiene como objetivo introducir y explicar los fundamentos teóricos en los que se basan los métodos empleados en el trabajo, así como de su relevancia para la resolución del problema planteado.

2.1. Aprendizaje Automático

El Aprendizaje Automático o *Machine Learning* (ML) [12, 13, 23] es una rama dentro de la IA centrada en desarrollar programas informáticos para resolver tareas complejas en donde no existe una solución analítica. Es decir, no es posible describir un algoritmo que dados los datos de entrada a dicho problema, los transforme a los datos de salida esperados. En esta situación, en muchas ocasiones se carece de detallados conocimientos de descripción del problema, que se pueden intentar compensar con datos relacionados al mismo. Dichos datos pueden ser utilizados para obtener una solución empírica, es decir, se está “aprendiendo” de los datos. De ellos se extraen patrones o reglas para construir un algoritmo aproximado, conocido como un modelo, capaz de resolver dicha tarea incluso al recibir datos nunca vistos. Se puede definir más formalmente que un programa aprende de la experiencia E con respecto a alguna clase de tareas T y una métrica de rendimiento P si su rendimiento en las tareas T , medido con P , mejora con la experiencia E . Dicho aprendizaje se puede dividir en dos grandes grupos: supervisado y no supervisado. En el primero se poseen datos de entrada y su correspondiente salida correcta, mientras que en el segundo, solo se poseen los datos de entrada y se espera que el programa encuentre patrones dentro de los mismos.

En general, los problemas a los que es práctico aplicar ML son aquellos donde (a) se poseen vastas bases de datos en las que se desea obtener patrones en los propios datos, lo que se conoce como minería de datos o *data mining*

[24], (b) aquellos problemas cuyos dominios no están bien entendidos o bien donde un humano no es capaz de expresarlo de una manera descriptiva para poder desarrollar un algoritmo, por ejemplo, problemas relacionados con detección de objetos en una imagen o (c) dominios donde el programa debe adaptarse dinámicamente a condiciones cambiantes [13].

Puede observarse que, dada estas descripciones, al problema presente es posible aplicarle ML: se poseen datos de entrada (los huesos de la sínfisis del pubis) y se tienen unos datos de salida (que serían los atributos que posee dicho hueso dentro de las 9 categorías del método de Todd). Además los antropólogos forenses poseen el conocimiento experto para identificar dichos patrones pero no pueden expresar dicho conocimiento de forma analítica. Es un problema de aprendizaje supervisado, específicamente un problema de clasificación, pues efectivamente se tiene que clasificar cada hueso en diferentes atributos de las categorías presentadas por el método de Todd.

2.2. Aprendizaje Profundo

El Aprendizaje Profundo o *Deep Learning* (DL) [14, 25, 26] es una técnica perteneciente a ML en la que las características relevantes de los datos del problema son aprendidas y extraídas automáticamente por el propio modelo. Esto a diferencia de otras áreas de ML donde las características son obtenidas a mano o *handcrafted*, en la que un experto humano se encarga de obtenerlas utilizando su conocimiento del problema. Se ha visto que para problemas excepcionalmente complejos, las características *handcrafted* tienden a ser más complejas y poseen un peor desempeño que las características extraídas automáticamente.

El modelo más utilizado en DL se conoce como red neuronal artificial o *artificial neural network* (ANN) [27, 28], en donde se poseen nodos de cómputo que se denominan “neuronas”¹ conectadas entre sí en capas. La red posee una capa de entrada, la cual recibe los datos en bruto y se conecta a una o varias capas “ocultas” que a su vez se conectan con la capa de salida. Esta serie de capas son las que permiten extraer y aprender las características relevantes de los datos, en las capas iniciales se extraen características de bajo nivel y con cada siguiente capa se va abstrayendo más y más la representación de los datos para obtener características de alto nivel que facilitan la resolución del problema propuesto. El número de capas que posee una ANN indica su profundidad, de aquí proviene el nombre de aprendizaje *profundo*, al utilizar más capas, por lo general se puede aprender mejor. Un ejemplo de una ANN clásica puede visualizarse en la Figura 2.1, donde también se puede observar la distinción entre una red superficial y una profunda.

¹Su origen es bioinspirado en la corteza visual del cerebro, aunque no son modelos que simulen como tal su funcionamiento.

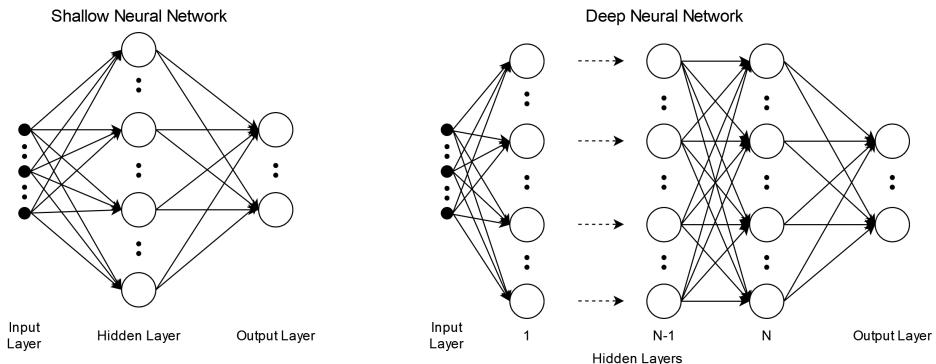


Figura 2.1: Ejemplos típicos de la estructura de una ANN, en este caso se tiene una red superficial o *shallow* y una profunda o *deep* [29].

Cada neurona, como se ha visto, tiene varios valores de entrada y un valor de salida que es utilizado como entrada para la siguiente capa, también posee una función no lineal² o de activación que transforma los datos de entrada en el valor de salida transmitido. La función de activación puede verse también como una especie de umbral, pues al ser no lineal, permite que la red pueda amplificar ciertos datos e ignorar otros dependiendo de lo que se desea aprender. Esto se realiza por medio de los pesos, que son valores asociados a cada entrada de cada neurona, así como un valor adicional conocido como sesgo o *bias*. Variar los pesos y el sesgo permite amplificar o reducir la señal que poseen los datos al ser procesados por la función de activación, un ejemplo visual de una neurona puede observarse en la Figura 2.2.

Existen multitud de funciones de activación, pero las más comunes y utilizadas son la función sigmoidal o logística, la función de la tangente hiperbólica y la ReLU o *Rectified Linear Unit*, la decisión de utilizar alguna función en particular dependerá del problema a resolver.

El aprendizaje de una ANN es en esencia un ajuste de los pesos y el sesgo de cada neurona para que, una vez sean transformados estos datos por la función de activación, se puedan extraer y transformar las características relevantes de los mismos. Para ello, primero se procesan los datos hacia delante, es decir, se procesan los datos desde la capa de entrada, pasando por las capas ocultas hasta llegar a la capa de salida. Esto se conoce como *forward propagation*, dónde se calculan los valores de salida de cada neurona y eventualmente un valor de salida de la red entera. Aquí es donde se utiliza

²De ser una función lineal, la red entera se simplificaría a una sola neurona, obteniendo un modelo lineal, esto se debe a que combinaciones de funciones lineales dan como resultado otra función lineal.

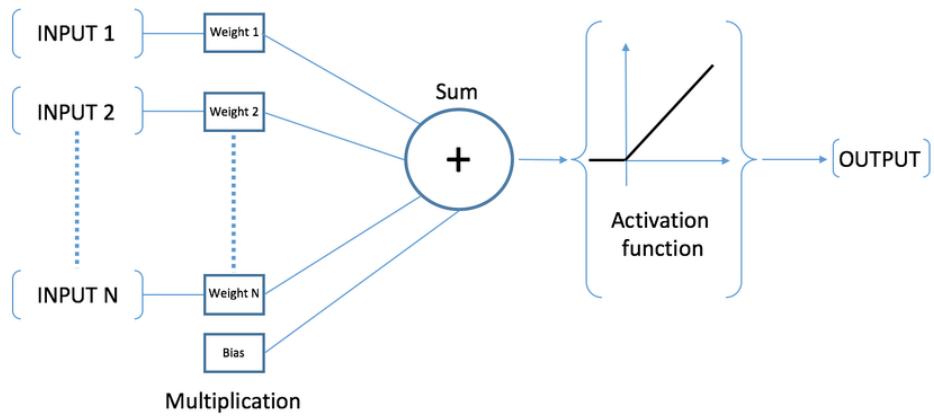


Figura 2.2: Ejemplo de una neurona artificial, los datos de entrada son multiplicados por los pesos y su resultado, junto con el sesgo (o *bias*), son combinados linealmente. A continuación, son transformados por la función no lineal para proporcionar el valor salida de la neurona [30].

una función adicional e igualmente importante: la función de pérdida o error, esta función toma los valores de salida de la red y calcula un valor de error, que se puede interpretar como lo bien que la red ha aprendido de los datos. Existen múltiples funciones de pérdida dependiendo del tipo de aprendizaje y la clase de problemas que se están tratando. Utilizando este valor de error, se aplica el algoritmo de *backpropagation* que obtiene las derivadas de los pesos y sesgo para cada neurona con respecto a ese error. Se recalculan los valores de los pesos haciendo uso de otro algoritmo, conocido como el optimizador u *optimizer*, que usando los valores de las derivadas, calcula los nuevos valores para cada uno de los pesos de la red con el objetivo de minimizar todo lo posible el error obtenido.

Este proceso iterativo se conoce como entrenamiento, y es lo que permite que los pesos se vayan ajustando a los datos de tal forma que aprendan “a fuerza” las características que mejor reducen el valor de error. Es importante mencionar que, dada esta forma de aprendizaje, es posible incurrir en un problema muy típico de ML que se conoce como sobreentrenamiento u *overfitting*. Este fenómeno ocurre cuando el modelo se ajusta demasiado bien a los datos de entrenamiento y no es capaz de generalizar el conocimiento adquirido para aplicarlo a datos nunca antes vistos. es decir, que en los datos de entrenamiento se obtiene un valor de error muy bajo y en datos nunca vistos se obtiene un error substancialmente más alto.

2.2.1. Redes Neuronales Convolucionales

Las redes neuronales convolucionales o *convolutional neural networks* (CNN) [31, 32] son un tipo de ANN más comúnmente utilizadas en el procesamiento, clasificación y segmentación de imágenes, pero también se han descubierto que son aplicables para el procesado de texto, sonidos y, de manera más reciente, en superficies tridimensionales. A diferencia de una ANN clásica, donde todas las neuronas están totalmente conectadas entre sí, una CNN posee dos tipos de capas adicionales: capas convolucionales y capas de *pooling*. La estructura básica de este modelo puede visualizarse en la Figura 2.3, donde se puede apreciar que una CNN se encuentra dividida en dos partes: una que posee solamente capas convolucionales y de *pooling* utilizada para la extracción de características y otra que se asemeja a una ANN clásica, donde se aprenden combinaciones no lineales que ayudan a la red a clasificar los datos.

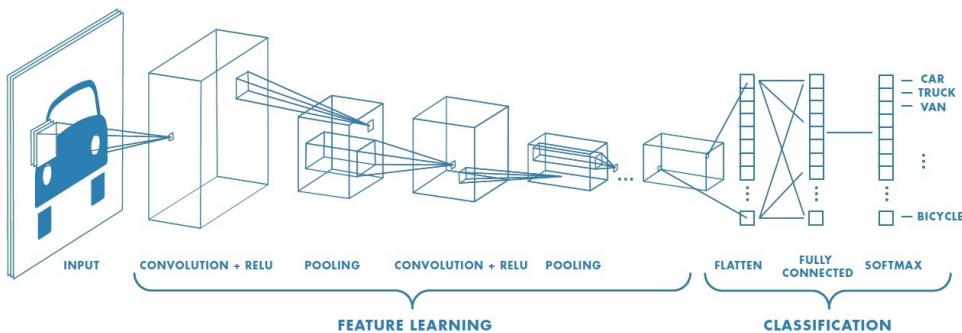


Figura 2.3: Estructura básica de una CNN donde se pueden apreciar las diferentes capas que la componen [33].

2.2.2. Capa convolucional

La capa convolucional es lo que caracteriza a una CNN, en estas capas cada neurona en vez de estar conectada con todas las neuronas de las capas vecinas, está solamente conectada con un vecindario local de neuronas, esto es posible al usar la operación de convolución para procesar las imágenes de entrada³.

Una convolución es el producto punto entre dos matrices: una es el conjunto de pesos que la red puede aprender y modificar que se denomina como el *kernel*, filtro o núcleo de la convolución y la otra es un trozo de la imagen denominado como campo receptivo o *receptive field* que son aquellos píxeles con los que el *kernel* se puede multiplicar en ese momento (véase Figura 2.4 para un ejemplo visual). El *kernel* se va desplazando por la imagen,

³De aquí en adelante, por comodidad, se explicará la CNN clásica que procesa imágenes, pero su funcionamiento es similar al procesar otros tipos de datos.

comienza en una esquina, moviéndose hacia la otra de fila en fila hasta recorrer la imagen entera. La matriz resultante es luego procesada por la función no lineal, al igual que en una neurona clásica y al resultado se le denomina como mapa de características o de activación y servirá de entrada para la siguiente capa de la red. Por medio de este proceso, la red es capaz de capturar dependencias temporales y espaciales en los datos con la aplicación de los filtros relevantes, ya que, al aprender los valores que deben de aplicarse a los filtros, la red será capaz de extraer las características relevantes de la imagen. De forma análoga a las ANNs clásicas, las primeras capas obtienen características de bajo nivel que serán utilizadas por las siguientes capas convolucionales para poder extraer características de más alto nivel.

Cabe mencionar que aplicar el operador de convolución directamente sobre la imagen, por la propia naturaleza del operador, resulta en una reducción del tamaño del mapa de activación. Esto no siempre es deseable y por lo tanto, se puede añadir relleno o *padding* a la imagen de entrada utilizando información ya presente en la misma, para que el mapa de activación posea la misma dimensionalidad que la imagen de entrada. También es posible reducir aún más la salida modificando los saltos o *strides* que da el filtro de convolución al recorrer la imagen.

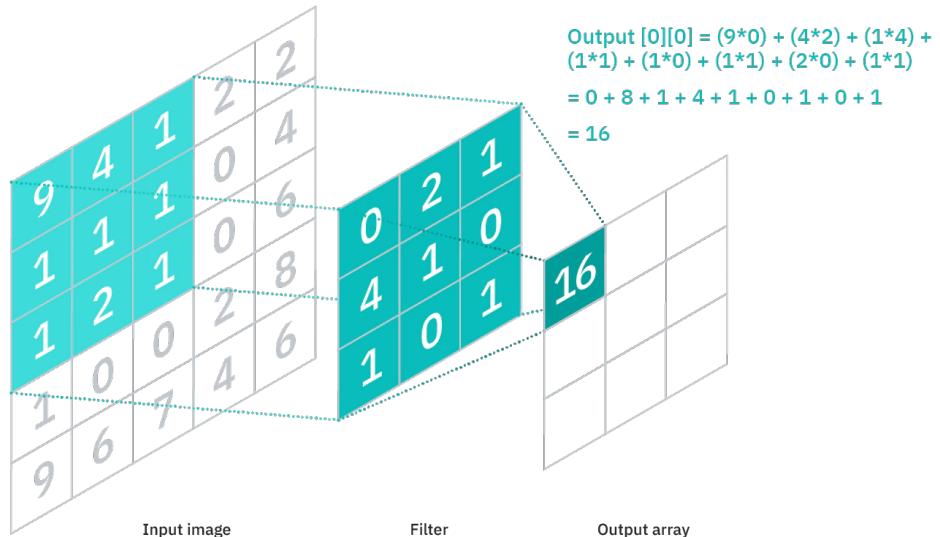


Figura 2.4: Ejemplo de una convolución 2D con un *kernel* 3×3 . El campo receptivo para la neurona actual es la submatriz 3×3 de la imagen original con la que se está multiplicando el *kernel* [34].

2.2.3. Capa de *pooling*

Las capas de *pooling* tienen como único objetivo reducir la dimensionalidad del mapa de activación de las capas convolucionales, y por lo tanto, se añaden justamente después de las mismas. Si bien es cierto que la propia convolución permite una reducción de la salida, se prefiere el uso de capas de *pooling* para esta tarea, pues es una manera más controlada de reducir el tamaño y además provee de beneficios a la extracción de características.

El *pooling*, al igual que la convolución, posee un filtro o ventana que recorre los datos dado un salto o *stride* al moverse por los mismos, con la diferencia que, en vez de aplicar una convolución, se pueden aplicar operaciones para obtener ya sea el valor promedio de los datos sobre el filtro, *average pooling* o el valor máximo, *max pooling*. En ambos casos se genera un efecto de invarianza ligera a la traslación y reducción de ruido, siendo más fuerte en la operación de *max pooling*. Ejemplos de estas operaciones se pueden apreciar mejor en la Figura 2.5.

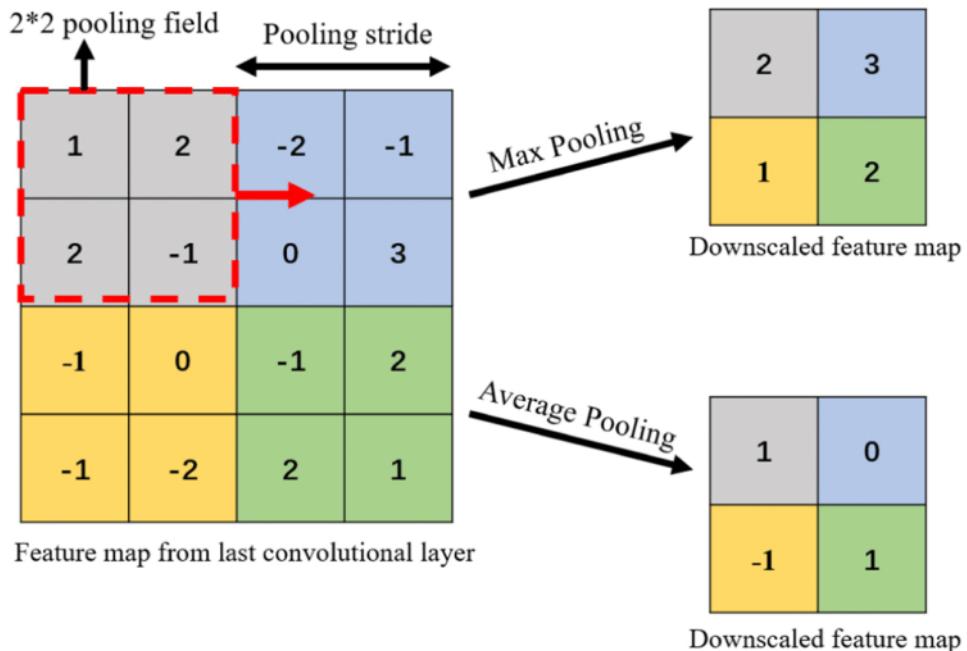


Figura 2.5: Ejemplo de las operaciones de *pooling* con un filtro 2×2 y *stride* 2 [35].

Las capas convolucionales y las capas de *pooling* juntas conforman las capas de procesado y extracción de características de las CNNs, y naturalmente, dependiendo de la complejidad del problema, el número de dichas capas será ampliado o reducido para así lograr extraer las características necesarias para el aprendizaje.

2.2.4. Capa totalmente conectada

Las capas totalmente conectadas o *fully connected*, también llamadas capas densas o *dense* aparecen después de todas las capas de convolución y *pooling*, y en esencia, estas capas forman una ANN clásica donde todas las neuronas están conectadas con las neuronas de las capas vecinas, que toma de entrada las características extraídas de los datos y de las mismas aprenderá combinaciones no lineales, para poder cumplir con el objetivo de la red, ya sea de clasificar o de estimar un valor numérico, por ejemplo.

La salida de la última capa densa, siendo la salida de la red entera, es donde se evaluará la función de pérdida elegida, y al igual que en una red neuronal clásica, se utilizará este valor para ajustar los pesos de toda la red mientras se encuentre en entrenamiento, por medio del *backpropagation* y el algoritmo optimizador seleccionado.

2.2.5. Regularización

Como se ha comentado anteriormente, tanto las ANNs como las CNNs son propensas al fenómeno de sobreentrenamiento u *overfitting*. La regularización es un método utilizado para combatir este fenómeno y en esencia lo que se quiere es controlar la complejidad del modelo, ya sea alterando los datos, el número de parámetros o el funcionamiento de la red. En general, la regularización es cualquier modificación que se puede realizar al modelo para que generalice mejor.

Se compone de múltiples técnicas diferentes, pero las más relevantes y utilizadas en CNNs son: La normalización, *data augmentation*, la inicialización de los pesos y *dropout*.

Normalización

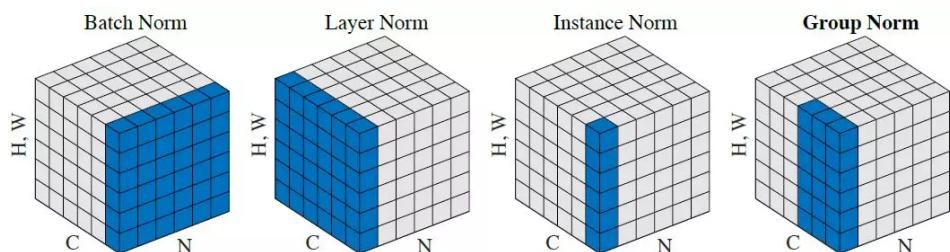


Figura 2.6: Diferentes tipos de normalización. **H,W** indican la altura y anchura de la imagen, **C** indica los canales y **N** el número de lotes [36].

La normalización es una técnica que estandariza los datos de manera que el valor medio de los mismos sea cercano a 0, con una desviación estándar cercana a 1, empíricamente se ha demostrado que esto mejora el rendimiento

de las redes, pues evita que los pesos posean valores muy grandes, lo que afecta el cálculo de gradientes.

Por lo general la normalización se aplica a las capas convolucionales de la red y la manera típica de utilizarla es haciendo uso de la normalización por lotes o *batch normalization*, en donde se aplica la estandarización a una característica i -ésima de entrada calculando la media y desviación típica de todas las características i -ésimas del lote. Existe también la normalización por capa o *layer normalization* que aplica la media y desviación típica por cada capa independiente del lote, la normalización por grupo o *group normalization* que aplica la normalización a un grupo de canales pero no toda la capa entera. Por último se tiene la normalización por instancia o *instance normalization* que normaliza cada canal por separado. El uso de un tipo u otro de normalización depende de la tarea a cumplir, pues se sabe que empíricamente diferentes tipos producen mejores modelos en diferentes problemas. Un ejemplo visual de todos los tipos de normalización se puede observar en la Figura 2.6.

Data Augmentation

El aumento de datos o *data augmentation*, es una técnica para aumentar la cantidad de datos que se poseen de manera artificial, significa que, se generan nuevos datos de los ya presentes mediante la aplicación de transformaciones aleatorias pero realistas. Por ejemplo, escalados no uniformes de la imagen, rotaciones, traslaciones, adición de ruido o transformaciones de perspectiva. De esta manera se poseen más muestras de entrenamiento diferentes, lo que reduce la posibilidad de *overfitting* porque la red tendrá más datos con los que trabajar y las modificaciones realizadas ayudan a la red a tener una idea más general de lo que se está aprendiendo.

Inicialización de los pesos

Como ha sido mencionado, las ANNs y CNNs utilizan pesos y sesgos en cada neurona, que son modificados en el entrenamiento por un algoritmo de optimización. Estos algoritmos necesitan un valor inicial que sea diferente de 0 al comienzo de dicho entrenamiento para funcionar, y la elección de estos valores afecta de gran forma al entrenamiento de la red, por lo tanto, se considera también como una técnica de regularización.

Se pueden inicializar los valores de forma aleatoria utilizando una distribución normal o uniforme que no toma en cuenta ningún parámetro de la red. Aún así, existen diversas heurísticas que se han desarrollado sobre los años que se ha comprobado mejoran el entrenamiento de los modelos. Por ejemplo, se tiene la inicialización Xavier [37] que parte de una distribución uniforme pero que toma en cuenta la cantidad de entradas que posee cada

neurona, por lo que la inicialización está parcialmente guiada por la densidad de las capas. Otra inicialización heurística se denomina Kaiming [38] y genera los valores por medio de una distribución normal acotada por el número de entradas de la capa.

Dropout

La técnica de abandono o *dropout* consiste en apagar o desactivar temporalmente cierta cantidad de neuronas en las capas totalmente conectadas de forma aleatoria durante el proceso de entrenamiento. La aplicación del *dropout* hace que la red generalice mejor, pues las neuronas en una capa totalmente conectada tienden a generar una codependencia entre ellas, es decir, que ciertas neuronas se adaptan para contrarrestar los errores de otras neuronas y debido a que estos errores dependen de los datos de entrenamiento, no se generalizará bien para nuevos datos. Por lo tanto el *dropout* permite evitar estas codependencias e impulsar el poder individual de cada neurona, lo cual aumenta el poder de generalización de la red.

2.3. Representaciones 3D en *Deep Learning*

Debido a que las CNNs fueron originalmente diseñadas para trabajar con imágenes, datos bidimensionales regulares, no existe todavía un consenso para la mejor representación de modelos 3D, ya que se tiene la complicada tarea de adaptar estos métodos y técnicas a datos tridimensionales irregulares. Adicionalmente, se trata de un área de investigación bastante reciente.

Aún así, con el auge de dispositivos más accesibles para la digitalización y generación de modelos 3D a partir de objetos físicos, la abundancia de modelos generados por ordenador y del surgimiento del concurso anual SHREC [39] (*3D Shape Retrieval Challenge*, Reto de Recuperación de Modelos 3D) existen hoy en día múltiples maneras en las que se han logrado representar la información 3D para ser procesada por una CNN con diferentes puntos a favor y en contra.

Actualmente se distinguen 5 categorías de representaciones tridimensionales, véase la Figura 2.7: Datos en bruto, sólidos, superficies, estructuras de alto nivel y datos de múltiples vistas, adicionalmente a ello, estas representaciones pueden a su vez clasificarse en datos euclídeos y no euclídeos, es decir, si los datos poseen o no una estructura euclídea subyacente, poseen una parametrización global o un sistema de coordenadas común. Esta taxonomía fue descrita por primera vez por Ahmed et al. [40] y ampliada por Gezawa et al. [41].

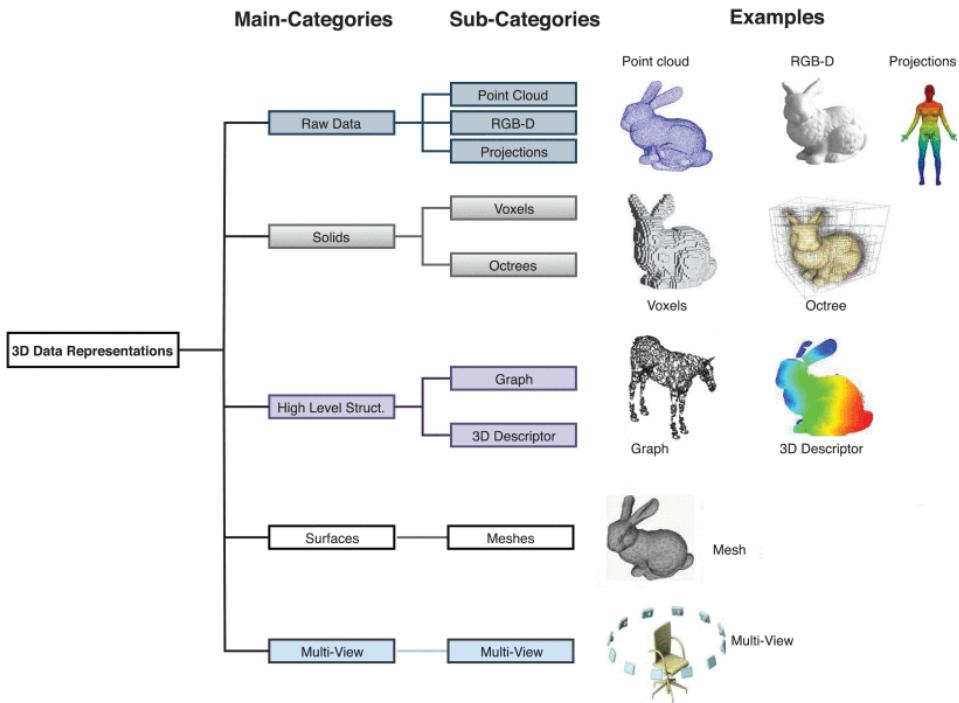


Figura 2.7: Taxonomía de las diferentes técnicas actuales en DL utilizadas para representar datos tridimensionales [41].

2.3.1. Datos en bruto

Los datos en bruto son aquellos obtenidos directamente por métodos de escaneo, por ejemplo, utilizando dispositivos como el Microsoft Kinect o con un escáner de luz estructurada, aplicando ninguna o muy pocas transformaciones a dichos datos.

Nube de Puntos

Son un conjunto de puntos sin estructura representados por coordenadas tridimensionales ya sea en coordenadas cartesianas u otro sistema, que simbolizan la geometría 3D de un objeto. Tiene su origen en fotogrametría y en tiempos más recientes, LIDAR.

Son fáciles de obtener y trabajar, siendo una de las representaciones más sencillas y cercanas a los datos en bruto, aunque su procesamiento puede ser un reto ya que se carece de información de conectividad entre puntos, por lo que pueden existir ambigüedades respecto a la forma real del objeto o bien pueden haber datos incompletos o espúreos que son difíciles de diferenciar de los datos de interés.

Datos RGB-D

Representación popularizada por el Microsoft Kinect, los objetos 3D se caracterizan con su información cromática en RGB a la que se le adiciona un canal nuevo, D, que indica la profundidad de cada píxel detectado, obteniendo información 2.5D del objeto.

Su mayor ventaja es la facilidad de adquisición y de procesado, por lo tanto existen multitud de *datasets* actualmente. El mayor problema que presentan es que son datos limitantes, puesto que la información que contienen no es suficiente para aprender la geometría entera de un objeto 3D.

Proyecciones

Son una manera de mapear los puntos 3D a planos 2D, se realiza utilizando proyecciones imaginarias que permiten trasladar características cruciales del objeto 3D a su proyección 2D. Las proyecciones cilíndricas y esféricas son las más populares, siendo invariante a rotaciones sobre el eje principal de dicha proyección.

Su mayor ventaja es que los datos contienen las características más importantes de la superficie 3D proyectada, además, que al trabajar en el dominio 2D se pueden aplicar fácilmente modelos ya conocidos para imágenes, sin embargo, no se recomienda para tareas más complicadas sobre las superficies tridimensionales, pues mucha información de la topología se pierde al realizar la proyección.

2.3.2. Sólidos

Las representaciones de sólidos para modelos 3D proporcionan la información sobre el espacio que ocupa el objeto, es decir, la información indica si cierto espacio tridimensional está ocupado o no por el mismo.

Vóxeles

La caracterización por véxeles se puede pensar como una rejilla regular en tres dimensiones en la cual el modelo 3D se encuentra distribuido. La información del punto de vista puede codificarse también, categorizando los véxeles como visibles para el punto de vista u ocultos.

Si bien ofrecen una representación total del modelo, el hecho de que se tienen que representar las partes del volumen que se encuentran ocupadas y vacías, conlleva a una demanda enorme e innecesaria de memoria, por lo que, no es factible utilizar esta representación para modelos de alta resolución.

Árbol octal

Se trata de una representación de véxeles más eficiente, a diferencia de utilizar una rejilla regular, el tamaño de los véxeles es variable. Los árboles octales modelan la información 3D como una estructura de datos jerárquica en forma de árbol que modela la ocupación del objeto en la escena 3D. Se basa en la descomposición recursiva de la escena en cubos que tienen cada uno 8 hijos, que pueden estar dentro o fuera del objeto.

Su principal ventaja es que es una representación más eficiente que los véxeles, además que, también pueden preservar mejor los detalles, pudiendo generar representaciones de alta resolución. Aún así, su mayor desventaja es la inhabilidad de mantener la geometría exacta de ciertos objetos 3D como lo sería mantener la suavidad de la superficie.

2.3.3. Estructuras de alto nivel

En las tareas de clasificación y en recuperación de formas 3D existe la necesidad de tener representaciones concisas, pero aún así, detalladas de los modelos 3D que se utilizan para describir un objeto como representativo de alguna categoría.

Descriptor 3D

En general, los descriptores 3D son representaciones simplificadas de los modelos 3D que describen las características geométricas o topológicas del mismo. Estos descriptores pueden obtenerse de la geometría, topología, superficie, textura, cualquier otra característica o una mezcla de todo lo anterior. Se pueden ver como una “firma” que caracteriza a un modelo 3D.

Su principal ventaja es que facilitan el procesamiento y cómputo de los modelos 3D, funcionan bastante bien en las tareas de comparación, análisis y recuperación de formas tridimensionales, particularmente en aprendizaje no supervisado. Sin embargo, una de sus mayores desventajas es que la utilidad se degrada al intentar utilizarlos en aprendizaje supervisado, ya que el descriptor extrae características de los datos en bruto, lo que es una abstracción de los datos del modelo 3D. Si se utiliza un modelo de aprendizaje supervisado se estarían aprendiendo abstracciones de una abstracción, lo que puede conllevar a la pérdida de información del descriptor si la representación es muy simple o abstracta.

Grafos

Esta representación permite resumir la información tridimensional conectando las diferentes formas por medio de grafos, esto es, los nodos del grafo

corresponden a los vértices del modelo y las aristas representan la conectividad entre los vértices. Los grafos pueden ser dirigidos o no dirigidos.

Las redes neuronales de grafos pueden clasificarse como métodos de filtrado espectral y métodos de filtrado espacial. Los métodos de filtrado espectral utilizan la descomposición en valores y vectores propios de la laplaciana del grafo para definir un operador similar a la convolución. Por otro lado, los métodos de filtrado espacial utilizan filtros de paso alto y paso bajo como combinaciones lineales de cada capa de la red. El aprendizaje está basado en el vecindario local de cada vértice, donde se aplica una función no lineal a cada nodo del grafo.

Este tipo de representación posee como ventaja que, en primer lugar es posible aplicar todos los modelos mencionados en representaciones de mallas 3D, adicionalmente, estos métodos han logrado resultados prometedores, pero son computacionalmente costosos y dependientes del grafo base utilizado, queriendo decir que la generalización entre distintos dominios es inconsistente.

2.3.4. Superficies

Este tipo de representación describe la superficie que cubre las partes internas de un objeto 3D como un conjunto de polígonos. Poseen la ventaja de ser simples, de fácil procesado y dibujado puesto que todas las superficies pueden ser caracterizadas con ecuaciones lineales. Existen muchos métodos de representación por superficies, como las subdivisiones, mallas paramétricas e implícitas aunque la representación más popular y utilizada para DL es la malla poligonal, particularmente, la malla triangular.

Malla 3D

Las mallas 3D consisten en una combinación de vértices, aristas y caras. Cada vértice posee una lista de conectividad que indica como están conectados entre sí, esta lista puede interpretarse como el conjunto de aristas, que a su vez, describen las caras de la malla.

Su principal ventaja es su amplia utilización e importancia en los gráficos por ordenador tanto para almacenar descripciones de modelos 3D así como su dibujado, aunque por su irregularidad y complejidad el estudio de su aplicación para tareas de DL no había podido ser abordado satisfactoriamente hasta hace pocos años. Hoy en día se trata de una de las ramas más nuevas, con modelos que han logrado resultados satisfactorios, aunque sufren de alto uso de recursos computacionales.

2.3.5. Múltiples vistas

Consiste en la representación de un modelo 3D como un conjunto de imágenes tomadas del modelo desde diferentes puntos de vista, utilizando técnicas típicas de dibujado de gráficos por ordenador. Estas imágenes luego se utilizan como entrada para una CNN convencional.

El mayor beneficio de esta representación es que se pueden aprovechar todos los métodos y técnicas ya existentes para las CNNs comunes basadas en imágenes, además, que esto permite utilizar modelos de alta resolución. Aún así, su mayor desventaja es determinar el número de puntos de vista a utilizar junto con la pérdida de información causada por modelos 3D que poseen partes que se solapan entre sí, ocultándolas del punto de vista. Adicionalmente, este tipo de representación no almacena las propiedades geométricas intrínsecas del modelo 3D y el uso de muchas vistas también tiene un coste computacional muy alto.

Capítulo 3

Estado del Arte

3.1. Estimación de la edad

La estimación de la edad, al ser un componente sumamente importante a la hora de obtener el PB, ha tomado mayor interés de los investigadores a comienzos del siglo actual y teniendo una tendencia creciente en el número de publicaciones. Puede observarse en la Figura 3.1 la cantidad de publicaciones existentes en la base de datos *Scopus* que hacen referencia tanto a AF como a la estimación de edad¹, desde 1977 se tienen 1144 publicaciones registradas.

Por otro lado, y afirmando lo mencionado sobre la precariedad tecnológica que posee esta área de conocimiento, existe un número muy limitado de publicaciones que hagan alguna mención no solamente de DL o ML sino de técnicas de IA en general², es decir, que incluso relajando los términos de búsqueda para englobar toda el área de IA, se tienen solamente 30 publicaciones que han surgido mayoritariamente a mediados de la década pasada. Finalmente, solo hay cinco artículos que combinan IA con modelos tridimensionales para la estimación de edad por medio de AF³. Esto se interpreta como un indicador de lo novedoso y pionero de este trabajo.

¹Consulta realizada: (TITLE-ABS-KEY (forensic AND anthropology AND age AND estimation)).

²Consulta realizada: (TITLE-ABS-KEY (((deep AND learning) OR (machine AND learning) OR (soft AND computing) OR (artificial AND intelligence) OR (data AND mining)) AND forensic AND anthropology AND age AND estimation)).

³Consulta realizada: (TITLE-ABS-KEY (((deep AND learning) OR (machine AND learning) OR (soft AND computing) OR (artificial intelligence) OR (data AND mining)) AND forensic AND anthropology AND age AND estimation AND 3d)).

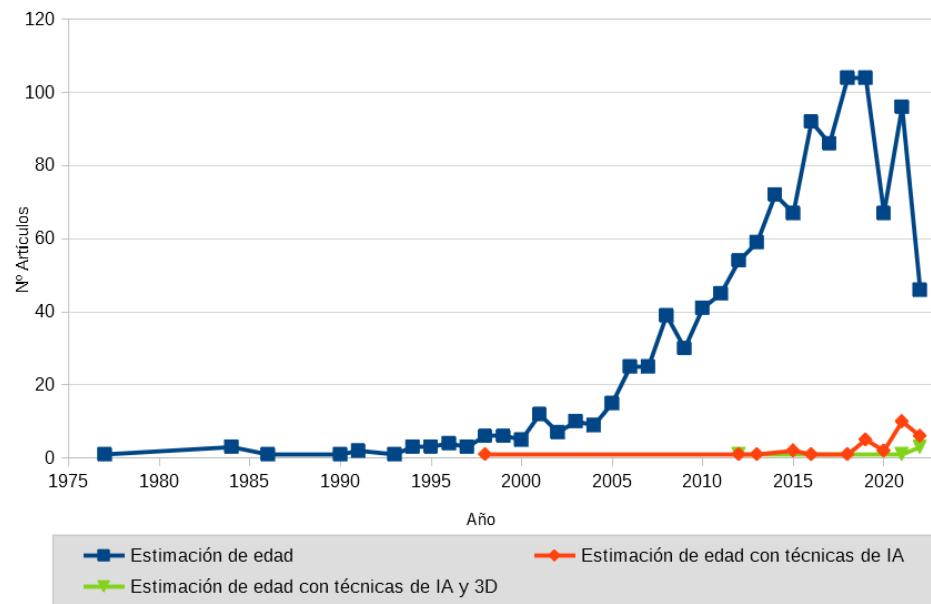


Figura 3.1: Número de publicaciones en *Scopus* en función del año, búsqueda realizada el 15/07/2022. En **azul** se muestran las publicaciones que mencionan AF y la estimación de la edad (1144 publicaciones), en **rojo** aquellas que adicionalmente mencionan alguna técnica de IA (30 publicaciones) y en **verde** aquellas que también mencionan el uso de 3D (5 publicaciones).

A continuación se presenta el estado del arte de tanto en los métodos tradicionales utilizando la síntesis del pubis y, brevemente, aquellos utilizados en otras estructuras óseas. Luego se presentan los métodos de estimación automáticos centrándose en la síntesis del pubis con el uso de DL y modelos 3D.

3.1.1. Métodos tradicionales para restos óseos

Por parte del cráneo, los métodos más utilizados son el método de Meindl y Lovejoy [42], el método de Acsádi y Nemeskéri [43], y el método de Mann [44]. Estos métodos estudian la osificación de las articulaciones fibrosas que conectan los diferentes huesos del cráneo, denominadas suturas craneales, para determinar la edad. Según una revisión reciente [45], los métodos producen resultados erráticos con baja precisión, por lo que, se aconseja solamente utilizarlo como apoyo a métodos de otras estructuras óseas, aunque el mismo estudio comenta que la incorporación de nuevas tecnologías, como la tomografía axial computarizada ha permitido reducir los errores de estimación de los métodos, véase [46, 47].

Por parte de las costillas, el método más utilizado es el de Íşcan y Loth [6,

48] que se centra en el desgaste por el paso de los años del extremo ventral de la cuarta costilla para la estimación de la edad. El método está limitado por sesgos, poca reproducibilidad en diferentes poblaciones y errores intraevaluador e interevaluador moderados [49, 50]. Hoy en día el método también incluye el uso limitado de tomografías axiales computarizadas [51].

Por parte de la cara auricular del ilion, los métodos utilizados son el de Lovejoy [52] y el de Buckberry y Chamberlain [7], sufren de igual manera de baja precisión y sesgos a la hora de estimar la edad del fallecido [53, 54]. Hoy en día el método puede aumentarse con el uso de tomografías [55, 56], así como la adición de métodos analíticos bayesianos con resultados mixtos [57].

3.1.2. Métodos tradicionales utilizando la sínfisis del pubis

La sínfisis del pubis es el hueso más utilizado para la estimación de edad, como ha sido mencionado en la Sección 1.1. Hoy en día se utiliza el método de Suchey-Brooks [11], variante del método de Todd. Estos métodos estiman la edad del fallecido dentro de rangos o intervalos y dependiendo de las características (Tabla 1.1) que posea la sínfisis del pubis, se clasifican dentro de un rango u otro de edad. El método original de Todd clasifica los huesos en 10 rangos mientras que el método de Suchey-Brooks lo realiza en 6 rangos nuevos que poseen solapamiento entre cada uno. Pueden observarse estos rangos en las Tablas 3.1 y 3.2.

Etapa	Rango de Edad
I	18-19
II	20-21
III	22-24
IV	25-26
V	27-30
VI	30-35
VII	35-39
VIII	39-44
IX	45-50
X	50+

Tabla 3.1: Rangos de edad del Método de Todd

Según un metaanálisis [58], el método de Suchey-Brooks es uno de los métodos con mejor exactitud al momento de determinar la edad, aún así, se recomienda su aplicación con cautela [59]. Actualmente, pueden aprovecharse las imágenes de tomografías computarizadas para aplicar directamente el método sobre los modelos volumétricos como si se tratase del hueso real. Otros investigadores [60-63] han hecho uso de esta tecnología para obser-

Etapa	Rango de Edad
I	15-23
II	19-34
III	21-46
IV	23-57
V	27-66
VI	34-86

Tabla 3.2: Rangos de edad del Método de Suchey-Brooks

var los cambios de densidad ósea que suceden internamente en el hueso al envejecer para refinar el método. Sin embargo, pese al uso de estas nuevas tecnologías de manera limitada, seguir utilizando características subjetivas conlleva a que los expertos basen las estimaciones finales de la edad más en su experiencia que en la información provista directamente por el método, lo que reduce su efectividad y credibilidad [9].

3.1.3. Métodos automáticos utilizando la sínfisis del pubis

Los primeros intentos para reemplazar los métodos subjetivos surgen a los inicios de la década del 2010 con el estudio realizado por Biwasaka et al [64]. Se calcula de forma analítica la curvatura media de la cara de la sínfisis del pubis escaneada en 3D y se examina qué tan cóncavo o convexo es el hueso respecto al intervalo de edad que se encuentra. El estudio utilizó 145 huesos y concluye que existe una relación entre cada fase del método de Suchey-Brooks y la curvatura media, aunque el estudio no reporta ninguna métrica estadística.

En Villa et al. [65] se amplía el estudio anterior al generar 5 variables del análisis de curvatura del hueso: la media del valor absoluto de la curva, el 10 % de valores de curvatura más altos, el 10 % de los valores más bajos, el porcentaje de superficies con curvaturas mayores que cero (superficies convexas) y el porcentaje de superficies con curvatura entre -0.01 y 0.01 (superficies planas). El estudio se realizó con dos colecciones, una de 24 huesos que obtuvo una correlación Spearman⁴ moderada a fuerte ($\rho = 0.60$ – 0.93) y otra de 98 que obtuvo una correlación débil a moderada ($\rho = 0.29$ – 0.51), valores similares a los obtenidos por técnicas manuales, por lo que los autores concluyen que esta línea de investigación tiene potencial.

En Slice & Algee-Hewitt [66] se utilizan mallas 3D de la sínfisis del pubis para desarrollar una métrica denominada *Slice Algee-Hewitt Score* (SAH-Score) que aprovecha las propiedades de las mallas 3D para realizar un análisis de componentes principales (PCA) de los vértices que forman el hueso. El valor

⁴Coefficiente de correlación de Spearman o Rho de Spearman

obtenido es utilizado como un indicador de la complejidad de la superficie de la sínfisis del pubis, que se utiliza como característica para obtener un modelo de regresión lineal que estima directamente la edad del fallecido. En el estudio se utilizaron 41 huesos y se obtiene una estimación directa de la edad, donde reportan un RMSE⁵ de 17.15 años, además, los valores predichos concuerdan con los intervalos descritos por Suchey-Brooks.

En Stoyanova et al. [67] se utiliza un algoritmo conocido como *Thin Plate Splines* o TPS para caracterizar las mallas 3D de la sínfisis del pubis. El algoritmo calcula la energía de flexión o *bending energy* (BE) que se necesitaría para doblar una hipotética placa infinitamente delgada de metal plana a la forma del hueso, el valor obtenido se utiliza para entrenar un modelo de regresión lineal. Se utilizaron 44 mallas con las que se obtuvo un RMSE de 19 años. Posteriormente en [68] se caracteriza el hueso utilizando la SAH-Score, la BE y también un análisis la curvatura del borde ventral. Se utilizan 93 huesos para entrenar un modelo de regresión lineal multivariante que obtiene un RMSE entre 13.7 y 16.5 años.

En Villar et al. [69] se hace uso de Árboles de Decisiones Difusos o *Fuzzy Decision Trees* (FDT) para inferir reglas que permiten obtener los rangos de edades utilizando las características descritas por Todd de 74 huesos, previamente etiquetados por dos expertos humanos. Se obtiene un MAE⁶ de 1.68 respecto al intervalo de edad del método de Todd, aunque por la baja cantidad de datos no se lograron obtener reglas de inferencia para todos los intervalos.

En Gámez-Granados et al. [70] se diseña un sistema explicable y basado en reglas. Se utiliza un algoritmo de clasificación ordinal llamado NSLVOrd [71] para clasificar 892 sínfisis del pubis dentro de los 10 intervalos de Todd, donde se utilizaron las 9 características del método previamente etiquetadas por expertos. Si bien el modelo es de clasificación, puede adaptarse para obtener el valor directo de la edad, en el estudio se reporta un RMSE de 12.34 años y un MAE de 10.38 años.

En Kotěrová et al. [72] se proponen nueve métodos de regresión automáticos utilizando características extraídas por un experto de la sínfisis del pubis. Se utilizaron enfoques de regresión clásica, así como K-vecinos cercanos (KNN), modelos Bayesianos, árboles de regresión y ANNs. La regresión multilínea obtuvo los mejores resultados para los 941 huesos utilizados en el estudio, con un RMSE de 12.1 años y MAE de 9.7 años. El estudio se amplía en [73] donde se utilizan modelos 3D escaneados y se propone un modelo de regresión multivariante, donde se utiliza la energía de Dirichlet para caracterizar la superficie del hueso y una CNN que tiene como entrada múltiples

⁵Error cuadrático medio

⁶Error absoluto medio

proyecciones 2D de la malla 3D. Se reporta que se obtiene un MAE de 11.7 y 10.6 años respectivamente, siendo los mejores resultados obtenidos de la literatura consultada.

Los tres últimos estudios mencionados son los más similares a este TFG. En [73] se utiliza una CNN para extraer automáticamente las características de la sínfisis del pubis de un modelo 3D, aunque es utilizado para estimar directamente la edad, a diferencia del objetivo de este proyecto, que es poder clasificar el hueso dentro de las características de Todd. En este sentido, los trabajos que hacen uso de dichas características directamente son [69, 70] y se utilizan para poder obtener reglas explicables para clasificar un hueso dentro de los intervalos de edad, no permiten identificar directamente de la sínfisis del pubis las características que posee. Puede verse que este trabajo es pionero en esta área, pues no existe según la literatura consultada, otro estudio que utilice técnicas de DL para extraer las características de Todd automáticamente y utilizando modelos 3D.

3.2. Representaciones 3D en Deep Learning

Como se observó en la Sección 2.3, existen múltiples maneras de representar los datos tridimensionales para ser utilizados en DL. Se realizará ahora un análisis de la representación más adecuada para la extracción automática de las características morfológicas para la estimación de la edad y se mostrarán los últimos avances en dicha representación.

La representación de nube de puntos es la más sencilla de todas, pero, debido a su simplicidad el concepto de vecindario local de puntos y de conectividad no está bien definido. Esto hace que la aplicación de las operaciones típicas de una CNN sea un trabajo no trivial y conlleva a ambigüedades, lo cual reduce su efectividad en modelos con gran detalle. Debido a esto, se descarta como representación a utilizar.

Por parte de la representación RGB-D, debido a que son representaciones 2.5D, no tienen el poder de representar fielmente la información que posee un modelo 3D complejo, por lo tanto, queda descartado como representación para este TFG.

Las proyecciones y las múltiples vistas, aunque poseen la ventaja de poder utilizar CNNs clásicas al transformar el objeto 3D en una o varias imágenes 2D, al igual que RGB-D pierden información valiosa de la topología al realizar la proyección. Adicionalmente, resulta un problema en superficies complejas puesto que partes del modelo pueden tapar u ocultar otras partes del mismo, perdiendo más información todavía. Por lo tanto, estas dos representaciones también son descartadas.

Los descriptores 3D simplifican el modelo 3D a una especie de “firma”,

teniendo en cuenta que se desean analizar las características que posee la síntesis del pubis y esto se traduce a que se desea analizar directamente la topología del modelo 3D, esta representación no resulta útil, al igual que los grafos, puesto que deben de transformar el modelo 3D. Adicionalmente, los descriptores poseen mejor desempeño en tareas de aprendizaje no supervisado que en supervisado, otra razón más por la que descartarlo.

Las representaciones volumétricas, como ha sido mencionado, no permiten representar exactamente los detalles que posee la superficie del modelo 3D. Además, poseen el problema de realizar operaciones en los espacios vacíos, lo que aumenta mucho el requerimiento de memoria. Por estas razones, son también descartadas.

La opción seleccionada son las mallas poligonales tridimensionales, puesto que son ubicuas en informática gráfica, lo que conlleva a la existencia de multitud de métodos para transformarlas y tratarlas. Adicionalmente, la mayoría de escáneres 3D utilizados por los expertos en AF directamente producen esta representación, lo que permite trabajar con los datos con comodidad. Son una representación eficiente y fiel de los datos, en las superficies lisas y planas se utilizan menos polígonos, mientras que en las superficies complejas existe mayor concentración para representar los detalles intrincados. A nivel de procesamiento por una CNN, poseen la ventaja que la representación provee también información de conectividad entre vértices y sus vecindarios locales.

3.2.1. Mallas poligonales

En el trabajo pionero de Feng et al. [74] se define MeshNet como el primer *framework* de CNNs que procesan directamente las mallas 3D. La unidad básica de procesado es la cara triangular de la malla, porque provee la conectividad necesaria para poder realizar la convolución, siendo equivalente a un píxel en una imagen. Lo que caracteriza dicha cara se divide en características espaciales y características estructurales, esto siendo el análogo al valor RGB de cada píxel. La característica espacial toma como entrada la coordenada que se encuentra en el centro de cada triángulo, mientras que las características estructurales se dividen en dos: Una que captura la estructura interna de cada cara para obtener la información de la topología y otra que captura la estructura externa que explora los vecindarios locales de las caras. De igual forma se modifica la convolución para que se realicen concordando con las características, tiene dos partes: la combinación de características espaciales y estructurales y la agregación de características estructurales que generan respectivamente dos características nuevas para el siguiente bloque de la red. Se utilizó el *dataset* de mallas conocido como ModelNet40 [75] para las pruebas, donde se reporta un 91.9 % de *accuracy* de clasificación, obteniendo el mejor resultado contra otros modelos que utilizan nubes de

puntos, volúmenes o múltiples vistas.

En Hanocka et al. [76] se define otro *framework* denominado MeshCNN, el cual surge casi en paralelo con MeshNet, siendo ambos trabajos publicados con meses de diferencia. A diferencia de MeshNet, MeshCNN hace uso de las aristas que conectan cada vértice de las caras poligonales como la unidad convolucional, es decir, el equivalente al píxel. En vez de utilizar dos tipos de descriptores para caracterizar una cara, se utiliza un vector de 5 componentes que caracterizan una arista utilizando diferentes atributos que son únicos a cada arista y por lo tanto, son invariantes a traslación y rotación. De esta forma se puede aplicar de una forma más directa una convolución similar a la utilizada en imágenes. También, implementa la operación de *pooling* de forma que la propia red aprende qué partes de la malla puede simplificar y cuáles no, para poder lograr el objetivo del aprendizaje. Los experimentos fueron realizados en diferentes *datasets*, se obtuvo un 98.6 % de *accuracy* en SHREC30 [77] y un 92.16 % en un *dataset* de cubos grabados con distintas formas [78]. En ambos *datasets* fue el modelo con mejores resultados comparados con técnicas de nubes de puntos y volumétricas.

En Schneider et al. [79] se introduce MedMeshCNN como una ampliación para MeshCNN puesto que la implementación original posee un alto consumo de memoria. Esto se mitiga en este trabajo modificando la operación de *pooling*, aunque el estudio se centra más en explorar las capacidades de segmentación del *framework* para mallas 3D médicas. Para los experimentos se utilizaron 94 mallas de aneurismas intracraneales, 65 provenientes *dataset* AneuRisk65 [80] y el resto provistas por el Hospital Universitario de Magdeburgo, Alemania. Se reporta un índice de Jaccard promedio de 63.24 % para todas las clases, con 71.4 % para la segmentación de las aneurismas en la malla. Los autores concluyen que el *framework* posee potencial para el área médica.

En [81], el autor también mitiga el alto uso de memoria de MeshCNN, creando MeshCNN+ que implementa una modificación del *pooling* y también permite que el entrenamiento pueda ser distribuido en diferentes ordenadores. Nuevamente centrándose en la habilidad del *framework* de segmentar modelos, se utilizó un *dataset* denominado ABC [82] que contiene un millón de mallas generadas por software de Diseño Asistido por Computadora o *Computer Aided Design* (CAD) reportando un *accuracy* de 86.2 %, aunque notando que el modelo no logra segmentar las clases poco representadas en la superficie de las mallas.

Capítulo 4

Materiales y Métodos

4.1. Materiales

Para este TFG se tiene disponible una colección de modelos 3D de la sínfisis del pubis izquierda y derecha de 571 fallecidos en formato de malla tridimensional. Estos fueron escaneados por el personal del laboratorio de Antropología Física del Departamento de Medicina Legal, Toxicología y Antropología Física de la Universidad de Granada. Además de la digitalización 3D, el personal de laboratorio examinó estas muestras y etiquetó el atributo asociado a cada una de las características que se podían observar en el modelo 3D de la sínfisis del pubis.

La edad de fallecimiento de estos 571 individuos se conoce y está representada gráficamente en la Figura 4.1. Como se observa en dicha figura, hay muchas más muestras de individuos de edad avanzada que de jóvenes (como es lógico también desde un punto de vista demográfico). De estos individuos se decide estudiar la sínfisis del pubis izquierda para evitar la variabilidad intrapaciente, también ignorando aquellas muestras con anomalías físicas. Existe un claro desbalanceo en los datos relacionado con el desbalanceo de la distribución de edad en la muestra. En la Figura 4.2 se observa que cada una de las características poseen muchas más muestras en un atributo que en otro, dado que la aparición de ciertas características y atributos de las mismas es más probable conforme aumenta la edad del individuo.

Con el tiempo que se posee para este trabajo, se decide estudiar un conjunto reducido de las características. Se consultaron varios expertos antropológicos para determinar cuáles eran relevantes pero al mismo tiempo fáciles de observar, de forma que se pudiera validar con mayor facilidad si el modelo está aprendiendo. Los expertos recomiendan las características de “Nódulo Óseo”, “Crestas y Surcos”, “Borde Superior” y “Borde Inferior” como las más sencillas y relevantes [70], aunque entre los mismos expertos no había

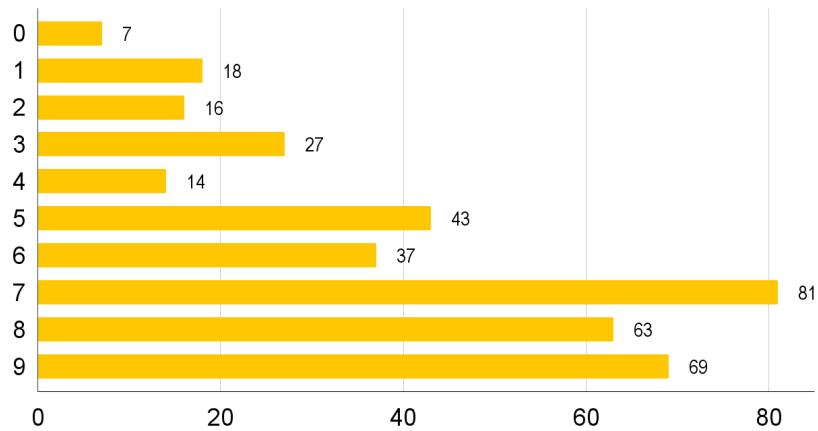


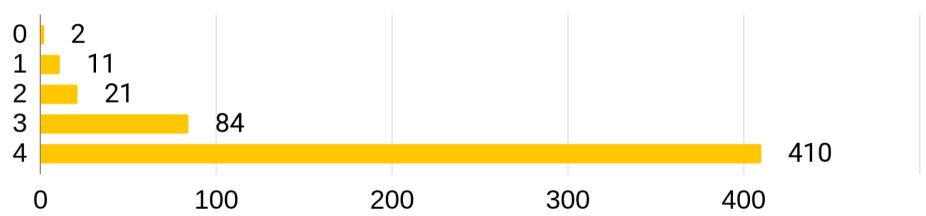
Figura 4.1: Distribución de los datos por cada rango de edad del método de Todd. **0:** 18-19 años, **1:** 20-21 años, **2:** 22-24 años, **3:** 25-26 años, **4:** 27-30 años, **5:** 30-35 años, **6:** 35-39 años, **7:** 39-44 años, **8:** 45-50 años, **9:** 50+ años.

un consenso, nuevamente demostrando la subjetividad que posee el método manual.

La característica de “Crestas y Surcos” (Subfigura 4.2a) se descarta por recomendación de los expertos, pues indicaron que depende mucho de detalles que posiblemente el escáner no pudiese detectar. Adicionalmente, posee un desbalanceo fuerte hacia la clase 4, que contiene la gran mayoría de las muestras, mientras que la clase 0 posee apenas 2. Las características restantes, “Borde Superior” (4.2c), “Nódulo Óseo” (4.2d) y “Borde Inferior” (4.2e) poseen un desbalanceo similar.

Debido a los retrasos en la obtención de los datos así como el tiempo que tardó el preprocesado (véase Sección 5.1.2 para detalles técnicos), los datos poseen geometría espúrea y huecos en las mallas que los métodos seleccionados no son capaces de soportar. Por lo tanto, se tuvo que restringir el estudio a la característica “Nódulo Óseo”. Se utilizan las 49 pubis que tienen presente el nódulo óseo y las primeras 49 que no lo tienen presente, para tener datos balanceados aunque reducidos.

Estas 98 sínfisis poseen una resolución promedio de 970 611.59 triángulos o de forma equivalente, 1 457 184.4825 aristas. Si bien un enfoque es utilizar la resolución nativa, se tiene que considerar el hardware que se tiene a disposición y el tiempo de cómputo requerido para procesar cada hueso. Se observó en el experimento 5.2.3 que aplicando operaciones de simplificación de mallas, descritas en 5.1.4, se mantiene la calidad incluso al reducir los datos a 30 000 aristas, teniendo una diferencia media de 0.0067 mm, con una diferencia máxima de 0.0878 mm entre las mallas originales y las reducidas.



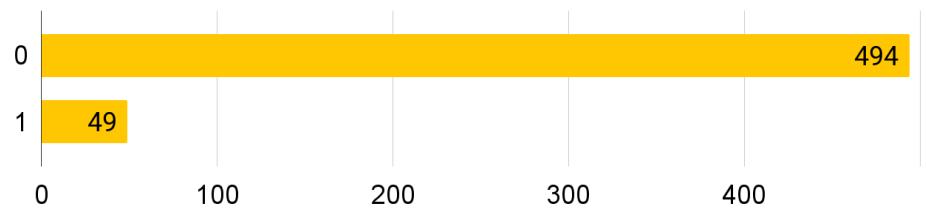
(a) **Crestas y Surcos.** 0: Porosidad Regular, 1: Muy Definida, 2: Poco Profundas, 3: Restos de Surcos, 4: No hay Surcos



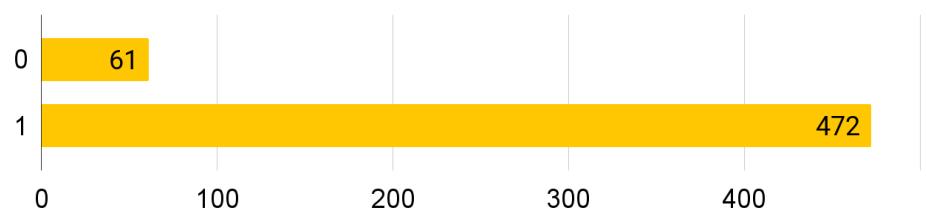
(b) **Porosidad Irregular.** 0: No, 1: Mediana, 2: Sí



(c) **Borde Superior.** 0: No Definido, 1: Definido



(d) **Nódulo Óseo.** 0: Ausente, 1: Presente



(e) **Borde Inferior.** 0: No Definido, 1: Definido



(f) **Borde Dorsal.** 0: No Definido, 1: Definido

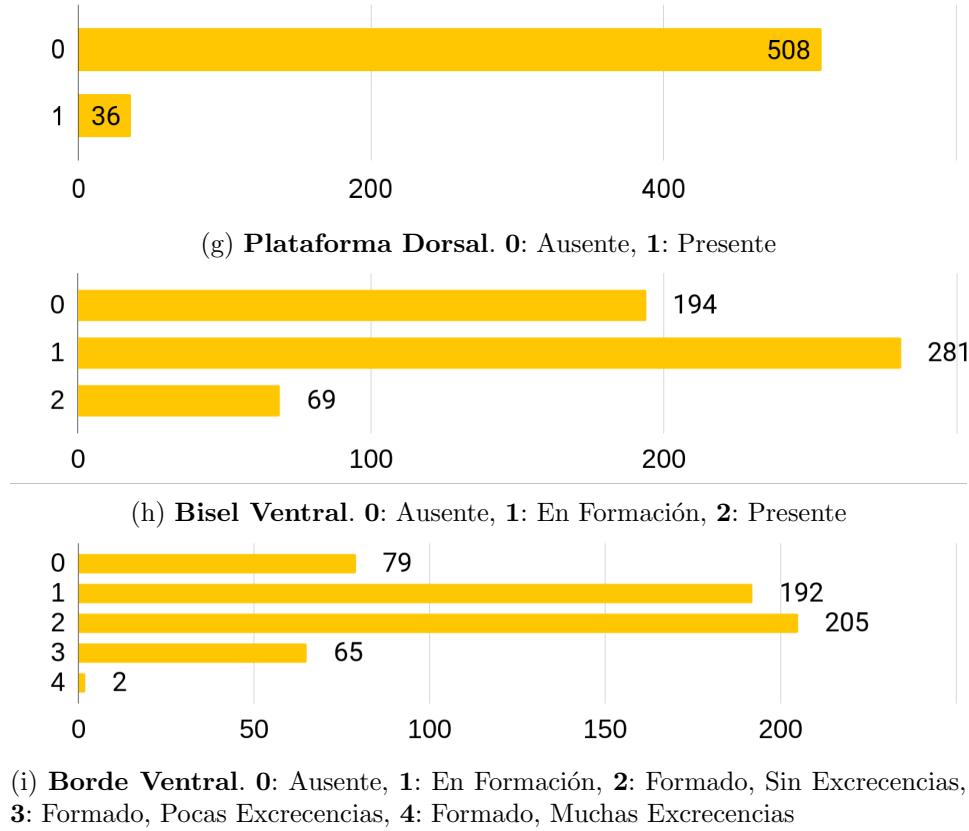


Figura 4.2: Distribución de los datos por cada característica del método de Todd

Se puede observar un ejemplo de las mallas 3D que posee el *dataset* en la Figuras 4.3 y 4.4, donde también se resalta la ausencia o presencia del nódulo óseo en la superficie de la sínfisis.

4.2. Métodos

4.2.1. MeshCNN

MeshCNN [76] es un *framework* para diseñar CNNs que procesan mallas poligonales en vez de imágenes, para ello, los autores tuvieron que realizar ciertas modificaciones a las capas convolucionales y de *pooling*, las cuales se explican a continuación.

Capa convolucional

Una representación por cuadrículas (por ejemplo, una imagen) es conveniente porque provee tanto información sobre conectividad (vecindarios locales



Figura 4.3: Síntesis del pubis con presencia de nódulo óseo.



Figura 4.4: Síntesis del pubis con ausencia de nódulo óseo.

de píxeles) así como las características que posee (valor RGB de un píxel) en una sola matriz. Sin embargo, como las mallas son irregulares no es posible utilizar este formato, se tienen que definir las características de forma separada a la conectividad.

Una malla 3D se define como un par (V, E) donde $V = \{v_1, v_2, \dots, v_n\}$, los vértices, son conjunto de coordenadas cartesianas en \mathbb{R}^3 . Mientras que $E = \{e_1, e_2, \dots, e_m\}$, las aristas, definen la conectividad por medio de duplas de vértices $e_k = (v_i, v_j)$. En el caso de mallas triangulares, un conjunto de aristas (e_i, e_j, e_k) formarían una cara triangular. En esta representación, asumiendo que las mallas son herméticas¹, se garantiza que cada arista puede formar parte de dos triángulos como máximo y, por lo tanto, es adyacente a cuatro aristas vecinas. Esto provee la información sobre la conectividad de la malla de manera similar a como lo hace un píxel con los píxeles a su alrededor en una imagen. Por ello, la convolución se decide aplicar sobre las aristas.

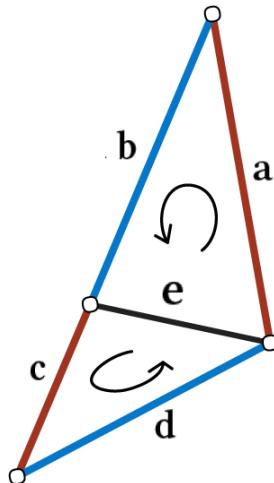


Figura 4.5: Vecindario local de una arista e

En informática gráfica los vértices, y por extensión las aristas, que forman una cara, se ordenan de forma antihoraria, si se aplica la convolución directamente produciría un campo receptivo ambiguo. Por ejemplo, si se observa la Figura 4.5, la convolución de la arista e y su vecindario podría aplicarse en el orden (a, b, c, d) o (c, d, b, a) , lo que afectaría el aprendizaje de los filtros. Entonces, para garantizar la invarianza, se aplica la operación 4.1 que permite obtener los mismos resultados independientemente del orden de las

¹Esto es, que la superficie de la malla proporciona una clara división entre lo que conforma el espacio interno del objeto y lo que conforma el espacio externo, donde no se permiten huecos que permitan pasar del espacio externo al interno sin tener que atravesar la superficie de la malla

aristas.

$$(e^1, e^2, e^3, e^4) = (|a - c|, a + c, |b - d|, b + d) \quad (4.1)$$

Los valores que se convolucionan son, dada una arista e , un vector de 5 componentes que contiene:

- El ángulo diedro entre los dos triángulos que conforma una arista, ϕ .
- Los ángulos internos perpendiculares a la arista, $\{\alpha_1, \alpha_2\}$.
- La relación entre la longitud de la arista y la longitud perpendicular de los triángulos, $\left\{ \frac{|e|}{|h_1|}, \frac{|e|}{|h_2|} \right\}$.

Visualmente, se puede observar esto en la Figura 4.6. Estas características son todas relativas a la arista, lo que quiere decir que es invariante a la rotación, traslación y escalado uniforme. Es el equivalente al valor RGB de un píxel en una imagen.

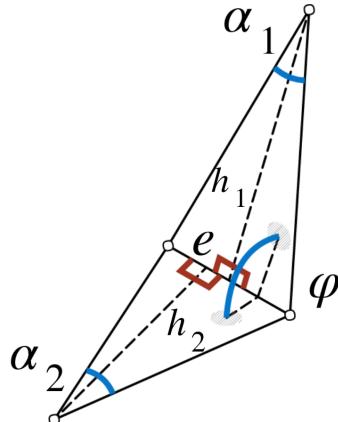


Figura 4.6: Características de entrada de una arista

Capa de *pooling*

Para extender la operación de *pooling* a las mallas 3D se observa que la esencia general del *pooling* consiste en: (1) tener una región que procesar dada por adyacencia, (2) combinar las características de cada región donde se aplica el *pooling* y (3) redefinir la adyacencia de las características combinadas. En imágenes, la adyacencia está inherente en los datos al ser una matriz y por lo tanto, la región del *pooling* depende del tamaño del *kernel* elegido. Las características, una vez combinadas, generan otra cuadrícula uniformemente espaciada, por lo que, la adyacencia se redefine de forma inherente. Con esto se puede observar que el *pooling* convencional es un caso especial del proceso general.

El *pooling* para mallas también es otro caso especial, donde la adyacencia se determina por la topología. A diferencia de las imágenes, que poseen un factor de reducción natural dado un *kernel*, se define el *pooling* como una serie de operaciones de colapso de aristas, en el cual cada operación de colapso convierte cinco aristas en dos, como se observa en la Figura 4.7. Por lo tanto, se puede controlar la resolución deseada de la malla después de cada operación de *pooling* con un hiperparámetro que controla la cantidad de aristas que se desean obtener.

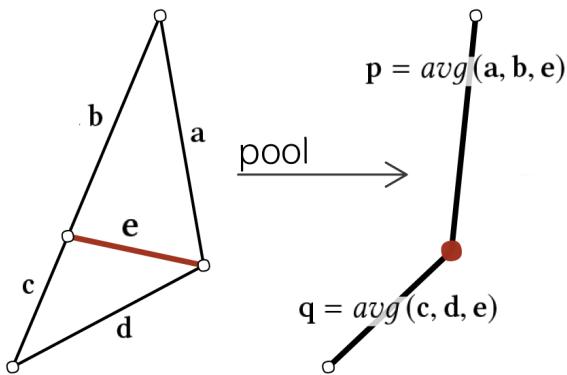


Figura 4.7: Ejemplo visual del *average pooling* por colapso de aristas.

Las aristas se encuentran en una cola de prioridad ordenada de menor a mayor por la norma ℓ^2 de cada una, lo que permite a la red colapsar de manera no uniforme las aristas con menor activación que posee la malla, y seleccionar aquellas que sí son relevantes para resolver el problema de aprendizaje. Las características de las dos aristas resultantes de la operación de *pooling* se calculan tomando el valor promedio de la arista central y las dos aristas que se combinarán en una, como se observa en 4.7, que p y q se calculan como el promedio de a, b, e y c, d, e respectivamente. Finalmente, notar que en MeshCNN no es posible que se colapsen todas las aristas, un colapso de aristas que haga que la malla no sea hermética está prohibido puesto que no sería posible aplicar luego el operador de convolución debido a los huecos generados.

Data Augmentation

MeshCNN también posee implementado un aumento de datos nativo para las mallas 3D. Realizar simples rotaciones, traslaciones o escalados uniformes no suponen nuevos datos, debido a la invarianza que poseen las aristas. Aún así, sí se pueden crear nuevos datos al aplicar un escalado no uniforme a las mallas, ya que eso sí afecta las características. También se pueden mover los vértices por la superficie de la malla o bien, se pueden rotar o voltear ciertas aristas de la superficie para generar datos adicionales. Esto se puede

controlar con diferentes parámetros que acepta el *framework*.

4.2.2. MedMeshCNN y MeshCNN+

Luego de la publicación de MeshCNN, surgen otras dos propuestas para ampliar y mejorar el *framework* original, tanto añadiendo nuevas funcionalidades como intentando mitigar el principal problema de la implementación, su altísimo consumo de memoria. Estas han sido consideradas para este proyecto también.

La implementación del *pooling* hace uso de una matriz cuadrada para almacenar en el entrenamiento las aristas que son colapsadas, esto con la intención de poder restaurar la malla a su estado antes del *pooling* para las siguientes iteraciones, ya que el *pooling*, como ha sido comentado, no es uniforme. El mayor problema es que la matriz está en su mayoría llena de ceros, y además, posee una complejidad en espacio de $\mathcal{O}(\text{aristas}^2)$ por cada capa que se añada, lo se traduce en un consumo alto de memoria de vídeo, por ejemplo, una sola capa que toma de entrada una malla de 50 000 aristas asignará aproximadamente 18 GB de espacio en el peor caso [81].

Schneider et al. introduce MedMeshCNN [79] como una ampliación que taclea el problema que posee la operación de *pooling* en MeshCNN haciendo uso de las matrices dispersas, esto le permite poder procesar mallas de hasta 170 000 aristas en una GPU con 12GB de VRAM. Esta ampliación se centra más en explorar las posibilidades del *framework* original respecto a la tarea de segmentación de modelos, por lo que, añaden métricas *Intersection over Union* o IoU, conocida también como el índice de Jaccard; así como el uso de una función de pérdida que trabaja con clases desbalanceadas. Aún así, en comunicación con los autores, se confirmó que esta implementación es igual de capaz de realizar clasificación con unas ligeras modificaciones.

En [81] se introduce MeshCNN+, que parte de la misma premisa de MedMeshCNN e intenta también reducir el consumo de memoria de la implementación original. En este caso, además de utilizar matrices dispersas, se hace uso del entrenamiento distribuido entre varios CPUs, dado que se hace notar que además del gasto de memoria de la matriz en la capa de *pooling*, existe también un cuello de botella respecto a la operación del colapso de aristas, que se hace secuencialmente y utiliza más la CPU que la GPU, por lo que sin el cómputo distribuido se desaprovecha la potencia de la GPU y aumenta el tiempo de entrenamiento. De igual manera que MedMeshCNN, este estudio se centró en la segmentación de mallas 3D, pero es posible utilizarla para clasificación.

Capítulo 5

Implementación y Experimentos

5.1. Detalles técnicos de la implementación

5.1.1. Entorno de desarrollo

Todo el proyecto ha sido desarrollado utilizando Python como lenguaje de programación principal, esto debido a que el método seleccionado, MeshCNN y sus ampliaciones, están desarrollados en el mismo. Los métodos también utilizan la librería de DL PyTorch junto con las librerías CUDA para poder ejecutar los modelos en tarjetas gráficas NVIDIA, acelerando así su ejecución. También se hizo uso de la librería Numpy para cálculos numéricos y de Pandas para el manejo cómodo de datos así como la librería scikit-learn para métricas adicionales. Se ha gestionado el uso de estas librerías junto con muchas otras de soporte por medio del gestor de entornos Anaconda. Finalmente, para mandar trabajos a un entorno de ejecución remoto se utilizó Slurm.

Para la escritura de scripts se utilizó Python junto a Bash y para el control de versiones se utilizó Git y Github. El repositorio de este proyecto se puede acceder por la siguiente dirección: <https://github.com/RhinoBlindado/tfg> y se encuentra estructurado en diferentes carpetas:

- **condaEnvs**, donde se almacenan los distintos entornos de Anaconda utilizados.
- **data**, donde se encuentran los datos a utilizar, se divide en **datasets**, en donde se encuentran los datos ya procesados y listos para utilizar por el modelo y, **original** donde se encuentran los datos originales. Por motivos de confidencialidad, no se encuentran disponibles los datos

de las sínfisis del pubis.

- **docs**, donde se encuentran los distintos documentos relacionados con el proyecto, incluído este mismo documento.
- **networks**, donde se encuentran las implementaciones de los distintos *frameworks* mencionados anteriormente: MeshCNN, MedMeshCNN y MeshCNNPlus.
- **output**, donde se encuentran todos los datos de salida de los modelos, incluidas las salidas obtenidas de Slurm.
- **programs**, donde se encuentran todos los programas compilados de terceros que han sido utilizados de alguna manera para el desarrollo del proyecto. Se encuentran los ejecutables de *Blender* y *Meshlab*.
- **scripts**, donde se encuentran todos los scripts desarrollados para el proyecto. En **dataPrep** se poseen aquellos que generan o alteran los datos, en **dataStats** aquellos que obtienen alguna métrica de los datos, en **colab** los *notebooks* utilizados en Google Colab y en **slurm** los scripts utilizados para slurm.

5.1.2. Obtención de los modelos 3D

Los datos se encuentran en una carpeta del servicio UGRDrive, que provee almacenamiento en la nube para investigadores. Los modelos de la sínfisis del pubis izquierda y derecha se encuentran dentro de una carpeta numerada por cada individuo que posee información generada por el proceso de digitalización y el software utilizado. Por ejemplo, se poseen los materiales, texturas, históricos, mediciones y otros archivos pertenecientes al software de escaneo. Un archivo descargado de la unidad posee la estructura dispuesta en la Figura 5.1 en donde lo que interesa son los archivos ***.obj**, ya que estos son los que poseen la malla 3D del hueso. Adicionalmente a esto se proporcionó un fichero **.csv** con las 9 características morfológicas de cada hueso por individuo.

Se desarrolló un script denominado **genBoneCSV.py** que dado el fichero **.csv** general genera otro **.csv** contenido solamente los huesos que poseen cierta característica, de esta manera se puede aislar distintas características para poder ser estudiadas individualmente, adicional a esto se tiene otro script, **extractBoneModel.py** que se encarga de aislar la malla 3D de la lateralidad deseada o ambas lateralidades del resto de información para cada individuo.

5.1.3. Preprocesado de los datos

Finalizados los procedimientos descritos en la Sección anterior, se tienen las mallas 3D de las sínfisis del pubis en bruto aisladas dada una o varias carac-

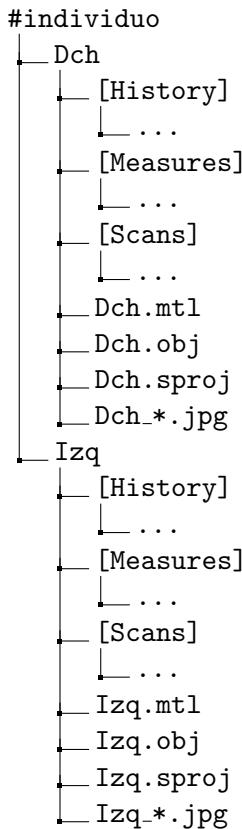


Figura 5.1: Estructura del archivo que contiene las síntesis del pubis escaneadas.

terísticas morfológicas a estudiar. Como ha sido comentado anteriormente, dado que el modelo de DL elegido posee limitaciones respecto a la cantidad máxima de aristas que es capaz de procesar por su alto consumo de memoria, y el requerimiento de que las mallas sean herméticas o *watertight*, esto requiere que las mismas deban de ser selladas y reducidas en número de aristas perdiendo la mínima información topológica posible.

Debido a que los modelos 3D no se encuentran perfectamente alineados con respecto a la cara del hueso que es relevante para el estudio, el trabajo de sellar las mallas tuvo que realizarse en su mayoría de forma manual. No existe una forma analítica de detectar la ubicación de la cara relevante del hueso si esta no se encuentra ubicada en un mismo rango de coordenadas para todos los huesos. Para lograrlo se tendría que volver a escanear, lo cual resulta inabordable. Se utilizó para esto el software de edición y modelado 3D gratis y de código libre llamado *Blender* [83] porque permite la simplificación de las mallas, la limpieza de geometría espúrea y posee un modificador denominado Booleao o *Boolean*, que dada dos mallas 3D en la escena, permite generar

nueva geometría aplicando operaciones de unión, intersección o diferencia de la geometría de las mallas originales.

El procedimiento para el sellado es el siguiente: (a) la malla 3D de la síntesis del pubis se importa a *Blender*. Para sellar la malla 3D y hacerla hermética, (b) se utiliza un cubo lo suficientemente grande para que pueda ofuscar la parte que no interesa de la geometría de la síntesis, se utiliza el modificador booleano realizando una diferencia entre el cubo y la síntesis del pubis (Figura 5.2) para que solamente se obtenga de resultado aquella parte del hueso que no esté dentro del cubo, automáticamente se elimina la geometría oculta y se calculan nuevas caras para sellar la malla. (c) Una vez finalizado esto, se eliminan las pequeñas superficies desconexas que puedan existir y se exporta la malla modificada. Se puede observar visualmente este proceso en la Figura 5.4 y la comparación de una malla antes y después de procesar en la Figura 5.3.

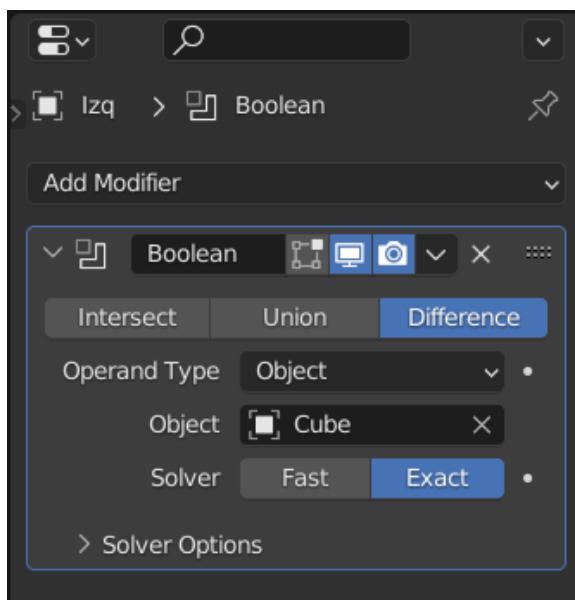


Figura 5.2: Detalle del operador Booleano de *Blender* con los parámetros utilizados para sellar los modelos.

Este procedimiento, además, tiene el beneficio de reducir significativamente la cantidad de aristas que se tienen por modelo sin perder calidad de la zona de interés. Permite retener más detalles una vez que se aplique la simplificación de las mallas, esto se debe a que si se utilizara el modelo original para la simplificación se estarían colapsando más aristas en toda la malla para llegar al número deseado, potencialmente perdiendo más información de la zona de interés.

Con la malla sellada, el siguiente paso ocurre dentro del script `meshCollapse`

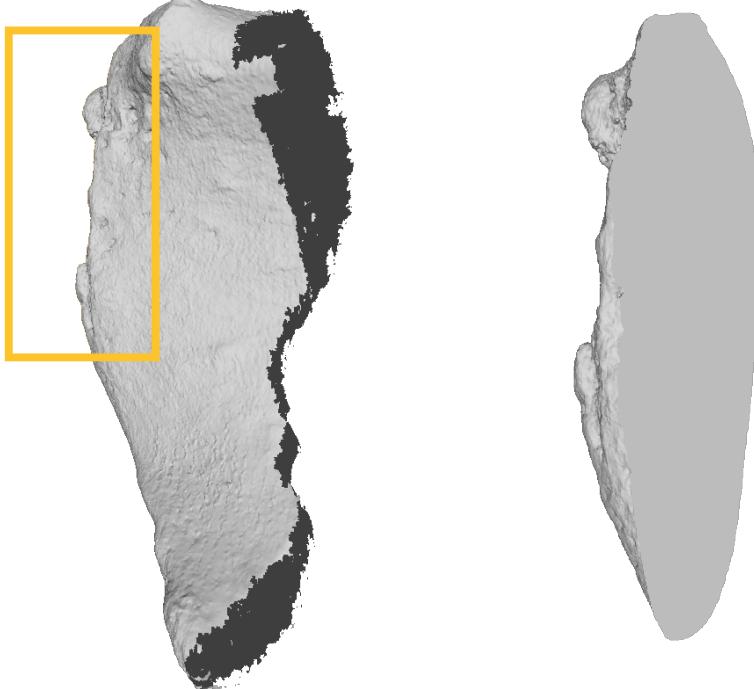


Figura 5.3: Resultado del sellado del modelo 3D. Se observa el modelo por detrás de la cara de la sínfisis del pubis. A la izquierda se observa el modelo original hueco y con estructuras no relevantes, a la derecha el mismo hueso luego del proceso de sellado.

.py que invoca a *Blender* en segundo plano y se encarga de reducir la malla a un valor deseado de aristas. También se encarga de eliminar los triángulos degenerados de la superficie general de la malla de la sínfisis del pubis, puesto que esta geometría espúrea produce errores al ser procesada por el modelo de DL.

5.1.4. Generación de *datasets*

Una vez que los modelos se encuentran ya preprocesados, para la generación de los conjuntos de datos se utilizaron dos scripts, denominados `genDatasetFolders.py` y `genDatasetCSV.py`. El primer script permiten generar los conjuntos de entrenamiento y pruebas, típico del área de ML, de forma que los datos están separados por carpetas; mientras que el segundo genera ficheros .csv que indican a qué conjunto pertenecen basándose en la estructura de ficheros. Cabe destacar que, originalmente la implementación del código de MeshCNN no poseía la capacidad para cargar los datos por medio de un fichero en formato .csv y que además, el uso de los mismos ha permitido añadir el conjunto de validación para poder tener más certeza de

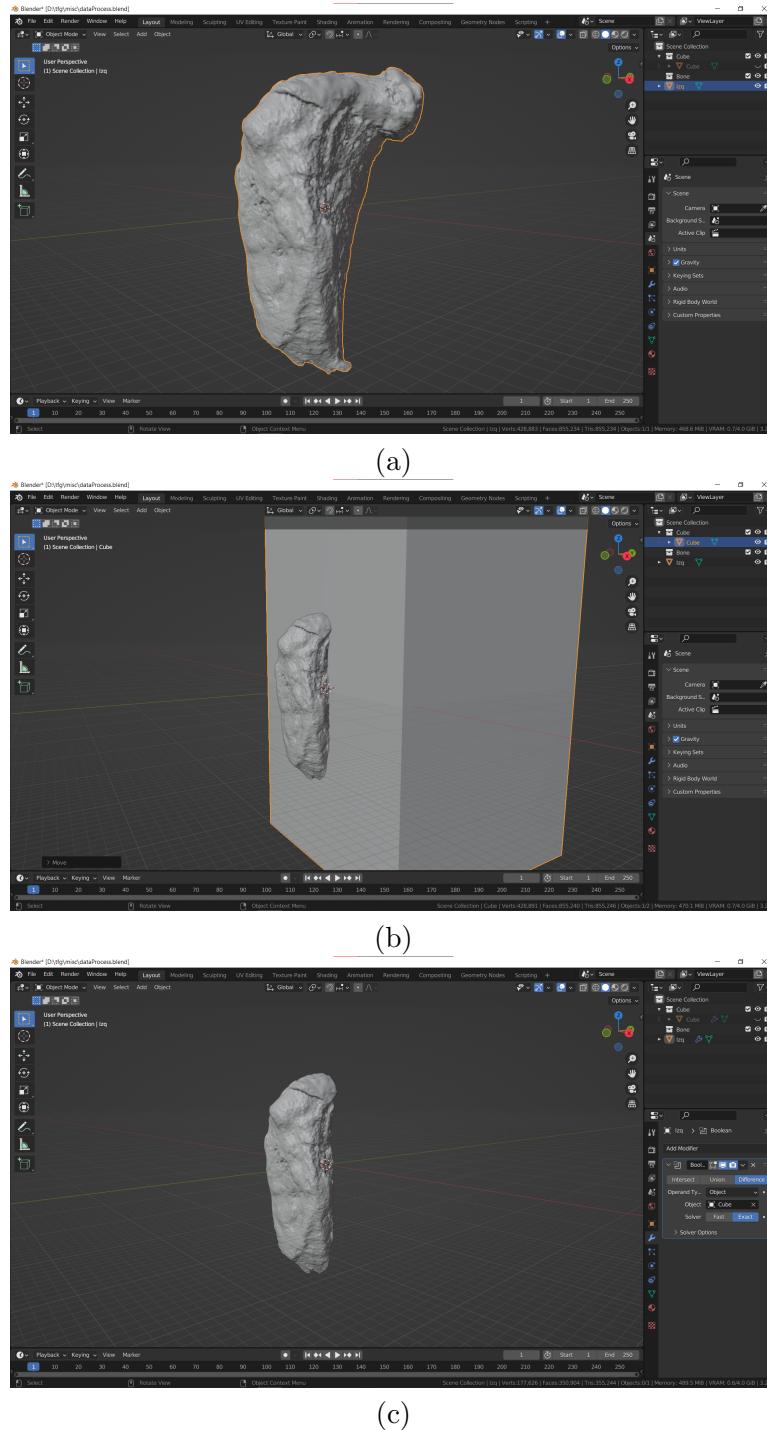


Figura 5.4: Proceso de sellado del modelo tridimensional de la síntesis del pubis. En (a) se observa el modelo original cargarlo a *Blender*, en (b) se utiliza el operador Booleano para calcular una superficie que sella el modelo y en (c) se observa el resultado obtenido.

la capacidades de aprendizaje de la red.

Por otro lado y opcionalmente dentro de `genDatasetFolders.py` es posible también realizar parte del preprocesado, se posee una opción que habilita la invocación de `meshCollapse.py` para que mientras se genera el conjunto de datos, también se reduzca la cantidad de aristas.

5.1.5. Estadísticas sobre los datos

Se escribieron un total de 7 scripts que obtienen de alguna manera u otra información relevante de los datos. Con `checkBoneExists.csv` se verifica la existencia de los huesos en alguna carpeta dado un `.csv`, en `checkHausdorff` se obtiene la distancia Hausdorff de las mallas poligonales, utilizado en un experimento; con `genLossPlot.csv` se generan los gráficos de las curvas de aprendizaje, con `getOBJStats.py` se obtienen estadísticas de los ficheros `.obj`, con `getParamDistr.py` se obtiene la distribución de datos de los huesos y finalmente con `memWatch.sh` se obtuvo el uso de memoria de VRAM durante un experimento.

5.1.6. Entrenamiento del modelo

Para el entrenamiento del modelo se debe de invocar al script `train.py` el cual recibe múltiples parámetros de entrada, donde se definen las cantidad de capas convolucionales, su densidad, la reducción del *pooling*, las capas totalmente conectadas así como regularización, la forma en que la red debe de inicializarse, dónde debe de leer los datos, etc. Una vez finaliza el entrenamiento, para obtener las métricas con el conjunto de test se invoca al script `test.py`.

Para la ejecución se utilizaron tres sistemas distintos. En las pruebas preliminares se utilizó un ordenador portátil ASUS FX505DT con una CPU AMD Ryzen 7 3750H, 16 GB de RAM y una NVIDIA GeForce GTX 1650 que posee 4 GB de VRAM. Como fue comentado en la sección 4.2.2, el alto uso de memoria que posee MeshCNN al momento de procesar mallas más complejas conlleva a utilizar el entorno de ejecución de Google Colab Pro, que posee un CPU Intel Xeon CPU E5-2699, 16 GB de RAM y una NVIDIA Tesla P100 con 16 GB de VRAM. Tras varias pruebas se concluyó que, aún con el aumento de las prestaciones, poder entrenar un modelo suficientemente complejo requeriría de mayor potencia de cómputo. Por lo tanto, el tercer y final entorno de ejecución se encuentra en el clúster de servidores GPU de la Universidad de Granada denominado NGPU ubicado en el CPD Santa Lucía, al que se accedió por SSH y se gestionó el entrenamiento con Slurm. Se utilizó con preferencia el nodo “Dionisio” que posee dos Intel Xeon Silver 4216, 512 GB de RAM y una NVIDIA Quadro RTX 8000 con 48 GB de VRAM.

Aunque los entrenamientos más complejos fueron realizados en NGPU, se utilizó también el entorno de Google Colab para realizar pruebas preliminares así como el ordenador ASUS que tuvo una función similar, además, de funcionar como el entorno para desarrollar más el *framework* descrito por MeshCNN.

5.2. Experimentos

5.2.1. Protocolo de validación experimental

Para el entrenamiento de los modelos se utilizó la técnica de *hold-out*: se divide el conjunto de datos en dos: El conjunto de entrenamiento y el de test, dentro del conjunto de entrenamiento a su vez se subdivide para obtener el conjunto de validación. Al momento de entrenar el modelo, se utiliza este último conjunto para verificar época a época la calidad del entrenamiento comparándolo con las métricas que se obtienen del conjunto de entrenamiento. Una vez se finaliza el entrenamiento el modelo es probado con el conjunto nunca antes visto de test para poder obtener una valoración final de la calidad del aprendizaje. Véase la Figura 5.5.

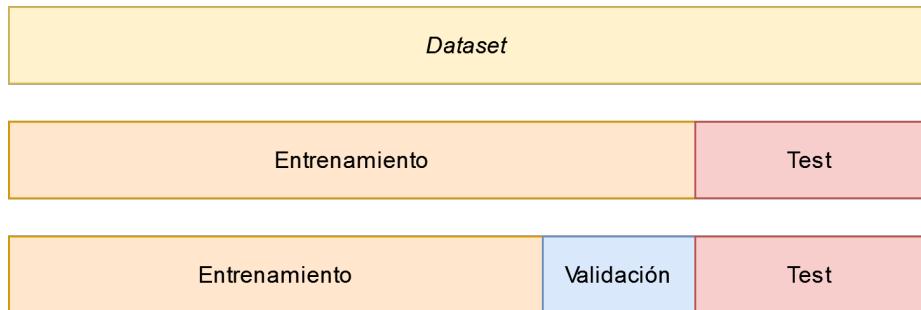


Figura 5.5: Esquema de división del conjunto de datos o *dataset* en los subconjunto de entrenamiento, validación y test.

Se generaron dos *datasets* denominados Nodule-50-30K y Nodule-98-30K que poseen respectivamente 50 (25 con presencia y 25 con ausencia de nódulo óseo) y 98 (49 con presencia y 49 con ausencia) muestras en total. En ambos casos se utiliza el 80 % de los datos para entrenamiento, lo que se traduce a un 40 y 78 muestras del total, con 10 y 20 muestras en test. En Nodule-50-30K, se utiliza el 15 % de las muestras de entrenamiento como validación, lo que se traduce en 6 muestras. Mientras que en Nodule-98-30K se utiliza un 12 %, es decir, 10 muestras del total.

5.2.2. Métricas

Debido a que se trata de un problema de clasificación binaria, se utiliza la típica función de pérdida de entropía cruzada o *cross entropy loss* para ir ajustando los pesos y sesgo de las neuronas del modelo, como fue explicado en la Sección 2.2. La función se define de la siguiente manera:

$$L_{CE} = - \sum_i^n t_i \log(p_i) \quad (5.1)$$

Para n clases donde t_i es la etiqueta verdadera y p_i es la probabilidad de la i -ésima clase. Esta función evalúa la certeza que tiene el modelo para obtener una probabilidad alta en la clase verdadera y baja en el resto de clases para los datos que se están utilizando en el entrenamiento. Si se tiene una certeza alta, el valor de la función tenderá a cero, en caso de baja certeza podría tender a infinito positivo.

Si bien esta es la métrica principal al momento de seleccionar el mejor modelo, no permite observar directamente que cómo de bien está clasificando las muestras. Para ello se utiliza la métrica de *accuracy* que permite observar la proporción de muestras que se están clasificando en la etiqueta verdadera correspondiente y por lo tanto es mucho más intuitiva. La variante binaria se define como:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.2)$$

Donde TP y TN indican verdadero positivo y verdadero negativo, esto es, las muestras clasificadas correctamente, mientras que FP y FN indican falso positivo y falso negativo, las muestras mal clasificadas. El *accuracy* lo que obtiene es la proporción de muestras clasificadas correctamente del total de muestras, aunque no se distinguen entre clases, por lo que se pueden obtener valores sesgados si existen clases desbalanceadas.

Para mitigar esto se utilizan las métricas de *precision* y *recall* que se definen como:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5.3)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5.4)$$

Precision indica que porcentaje de las predicciones positivas son correctas, mientras que *Recall* indica qué porcentaje de muestras que son realmente positivas fueron correctamente clasificadas. Esto permite observar si el modelo posee algún sesgo obteniendo más falsos positivos o falsos negativos.

Estos dos valores se juntan para obtener el valor $F1$ que permite comparar de forma más cómoda los diferentes modelos, se define como:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5.5)$$

Se utiliza también la matriz de confusión para visualizar directamente los valores TP , TN , FN y FP , pues resulta útil para obtener una intuición de las métricas mencionadas anteriormente.

Finalmente, se incluye una métrica adicional denominada distancia Hausdorff. Esta métrica, a diferencia del resto, no es utilizada para evaluar el modelo entrenado; se utiliza para evaluar las mallas poligonales. De formal, la distancia Hausdorff se define como:

$$d_H = \max \left\{ \sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y) \right\} \quad (5.6)$$

Esta es la distancia máxima entre dos subconjuntos que se encuentran en espacio métrico. En el contexto de las mallas 3D [84], se aplica esta métrica por cada triángulo de una malla buscando el triángulo más cercano en otra, permitiendo determinar qué tan diferente es la superficie de una malla respecto de otra. Se utiliza el software gratis y de código abierto *MeshLab* [85] para calcular la distancia. Al aplicarse se obtiene la distancia media y máxima de diferencia entre las dos mallas en unidades absolutas que utilicen las mismas (metros, centímetros, etc) o como un porcentaje de diferencia relativo entre ellas.

5.2.3. Experimentos preliminares

Previo a tratar directamente con los datos proporcionados, se realizaron unos experimentos con el fin de familiarizarse con el *framework* y sus derivados, el flujo de trabajo y las limitaciones frente a los datos del estudio.

Verificando funcionalidad base con SHREC16

Para familiarizarse con el *framework* y validar su funcionamiento correcto, se repite el experimento hecho por Hanocka et al. [76] utilizando el *dataset* de SHREC16 [77], el cual posee un total 600 mallas repartidas en 30 categorías balanceadas con una resolución de 750 aristas. Se utiliza la misma partición de datos de 80 % entrenamiento y 20 % test. Adicionalmente se incluye el conjunto de validación que hace uso de un 20 % del conjunto de entrenamiento. Se ha utilizado la misma estructura de red e hiperparámetros que los utilizados por los autores. En la Tabla 5.1 se pueden apreciar los hiperparámetros más relevantes, en la Tabla 5.2 el aumento de datos utilizado y en la Tabla 5.3 la arquitectura detallada del modelo.

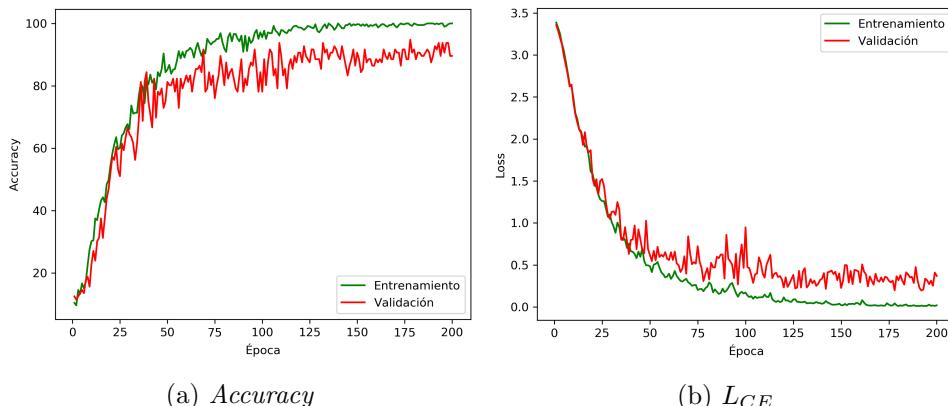
Hiperparámetro	Valor
Optimizador	<code>optimizer</code>
Aristas de entrada	<code>ninput_edges</code>
Tamaño de Batch	<code>batch_size</code>
Épocas	<code>niter</code>
Épocas con decaimiento de LR	<code>niter_decay</code>
LR Inicial	<code>lr</code>
Tipo de inicialización	<code>init_type</code>

Tabla 5.1: SHREC16, Hiperparámetros.

Modificación	Valor
Nro. Mallas aumentadas	<code>num_aug</code>
Proporción aristas a invertir	<code>flip_edges</code>
Proporción vértices a deslizar por superficie	<code>slide_verts</code>
Escalado no uniforme aleatorio de vértices	<code>scale_verts</code>

Tabla 5.2: SHREC16, Aumento de datos.

El modelo, que también se denominó **SHREC16**, se logró entrenar satisfactoriamente, además que permitió familiarizarse con el flujo de trabajo y realizar adiciones pertinentes al código. Se puede observar en la Figura 5.6 que las curvas de entrenamiento y validación se encuentran juntas y con la misma tendencia, tanto en *accuracy* como en el valor de error, por lo que se valida que el modelo utilizando el *framework* de MeshCNN sí aprende y se obtiene un resultado similar al de los autores: un 97,5 % de *accuracy* en el conjunto de test con un error de 0,126.

Figura 5.6: Modelo SHREC16. Curvas de aprendizaje de *accuracy* y error.

Dado que **SHREC16** es el único modelo conocido que resuelve un problema

#	Capa	Dimensiones		Canales	
		Entrada	Salida	Entrada	Salida
1	<i>MeshConv</i> (5×5) + ReLU	750	750	5	64
2	<i>GroupNorm</i>	750	750	64	64
3	<i>MeshPool</i>	750	600	64	64
4	<i>MeshConv</i> (5×5) + ReLU	600	600	64	128
5	<i>GroupNorm</i>	600	600	128	128
6	<i>MeshPool</i>	600	450	128	128
7	<i>MeshConv</i> (5×5) + ReLU	450	450	128	256
8	<i>GroupNorm</i>	450	450	256	256
9	<i>MeshPool</i>	450	300	256	256
10	<i>MeshConv</i> (5×5) + ReLU	300	300	256	256
11	<i>GroupNorm</i>	300	300	256	256
12	<i>MeshPool</i>	300	180	256	256
13	<i>GlobalAvgPooling</i>	300	256	256	256
14	<i>Dense</i>	-	-	256	100
15	<i>SoftMax</i>	-	-	100	30

Tabla 5.3: SHREC16, Arquitectura detallada del modelo. *MeshConv*: Convolución de malla, *GroupNorm*: Normalización de grupo, *MeshPool*: *Pooling* de malla, *GlobalAvgPooling*: *Pooling* global promedio, *SoftMax*: Capa totalmente conectada con función de activación *SoftMax*.

similar al de este TFG utilizando mallas poligonales, se eligió como la base de los modelos que tratarán con las síntesis del pubis. Debido a esto, partes de la arquitectura actual serán reutilizadas sin modificación: se mantienen el tamaño de los filtros de convolución con *padding*, se mantiene el *pooling* global que conecta los bloques convolucionales a las capas totalmente conectadas y se mantiene la última capa con la función de activación *SoftMax*. Para mayor claridad, en los siguientes experimentos, se adoptará una forma condensada de representar la arquitectura de la red, asumiendo que los detalles anteriormente mencionados se mantienen iguales. Se puede observar en la Tabla 5.4 la representación resumida de SHREC16, siendo equivalente a la representación detallada en la Tabla 5.3.

Explorando la interpretabilidad de los modelos

Con la finalidad de comprobar las posibilidades de interpretar el modelo, o de forma más coloquial, tener una idea de lo que se está fijando la red para aprender, se pueden utilizar las mallas intermedias que se producen luego de cada operación de *pooling*. Como se ha explicado en la Sección 4.2.1, este *framework* realiza un *pooling* donde se mantiene detalle en las partes de la malla 3D que la red utiliza para clasificar una muestra. Se desea obtener una

Capa	Cantidad y/o densidad
Convolución + ReLU	<code>nfc</code>
Normalización	<code>norm</code>
<i>Pooling</i>	<code>pool_res</code>
Densa	<code>fc_n</code>
<i>Dropout</i>	<code>dropout</code>
	0.0

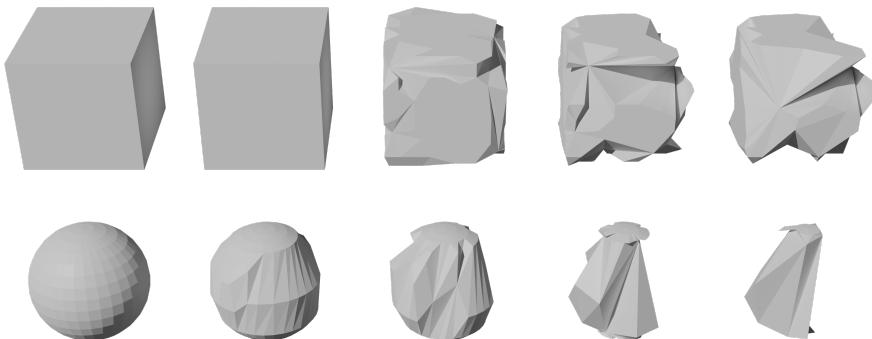
Tabla 5.4: SHREC16, Arquitectura resumida del modelo

intuición sobre esta funcionalidad para permitir la interpretabilidad humana de las decisiones que toma el modelo.

Capa	Cantidad y/o Densidad
Convolución	<code>nfc</code>
Normalización	<code>norm</code>
Pooling	<code>pool_res</code>
Densa	<code>fc_n</code>
<i>Dropout</i>	0 %

Tabla 5.5: SphereCubes, Arquitectura.

Para esto, se generó un *dataset* simple que contiene 20 objetos con un máximo de 1440 aristas, 10 esferas y 10 cubos con diferentes grados de deformaciones. Se adaptan las capas de *pooling* y de entrada de SHREC16 manteniendo el resto de hiperparámetros iguales. Se denomina el modelo como **SphereCubes** con la nueva arquitectura definida en la Tabla 5.5. Siendo datos muy sencillos, en apenas 10 épocas ya se tiene un 100 % de *accuracy* en validación y test con un L_{CE} de 0.4812 y 0.533 respectivamente.

Figura 5.7: SphereCubes, colapso de aristas luego de cada operación de *pooling* para el cubo `cube0.obj` y la esfera `sphere0.obj`.

En la Figura 5.7 se observa las mallas resultantes luego de cada operación de

pooling para dos muestras del conjunto de test, `cube0.obj` y `sphere0.obj`. Para cada objeto, la red está colapsando distintas aristas, esto se nota en la geometría que tienen las mallas en la última capa de *pooling*. Se puede pensar que esto es una especie de mapa de activación 3D, pues muestra las partes de la malla que han sufrido menos pérdida de detalle y por lo tanto, son útiles para la red al momento de determinar si se trata de un cubo o una esfera. Para mostrar mejor las partes de la malla original que se han mantenido o perdido, se utiliza la distancia Hausdorff, midiendo la distancia entre los vértices de la malla original y la malla resultante de la última capa de *pooling*. Para los objetos mencionados, se puede observar el mapa de activación obtenido en la Figuras 5.8a y 5.8b, pintándose de color rojo los vértices con menor distancia entre sí, mientras que en azul se pintan los que poseen mayor distancia.

Se puede observar que para el cubo, la red mantuvo más detalles en las tapas laterales así como la tapa superior pues se muestran con color rojo, mientras que las puntas han importado menos, estando coloreadas de azul. Por parte de la esfera, la superficie que la caracteriza son las tapas superior e inferior, que puede verse hasta en el modelo procesado, que se ha mantenido esa parte intacta y en efecto aparece coloreada de rojo, mientras que el centro de la esfera no ha sido de mayor importancia, estando coloreada en tonos verdes y azules. Este patrón se mantiene para otras muestras, por ejemplo, en las Figuras 5.8c y 5.8d se obtiene la distancia entre otros dos objetos y en efecto, la red se fija en zonas similares de la superficie para decidir si es un cubo o una esfera, por lo que se puede concluir que las mallas obtenidas luego del *pooling* permiten dar interpretabilidad a las decisiones que toma el modelo para clasificar objetos.

Analizando pérdida de calidad al reducir la calidad de las mallas

Debido a la gran resolución de los datos obtenidos, se desea realizar un análisis para determinar si es posible utilizar mallas simplificadas para acelerar el procesado por los modelos con la mínima pérdida de información respecto a la superficie de la sínfisis del pubis.

En primer lugar, se generan tres conjuntos de datos: de 30 000, 50 000 y 100 000 aristas; partiendo de los datos originales. Se realiza un análisis de la distancia Hausdorff para comprobar que tanta calidad se está perdiendo al realizar el colapso de aristas más y más fuerte, pues la pérdida de calidad de la superficie de la sínfisis puede reducir también la calidad del modelo entrenado. Se puede observar en la Tabla 5.6 los resultados de comparar las 98 muestras con calidad original y las mismas muestras reducidas a dichas resoluciones. Se utiliza la diagonal caja delimitadora promedio, de 51.6024 mm para obtener el porcentaje de error.

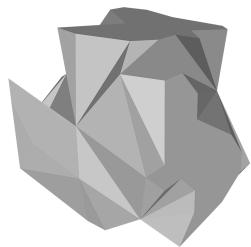
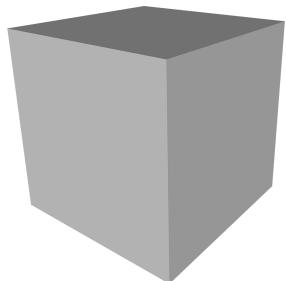
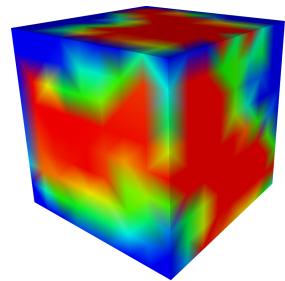
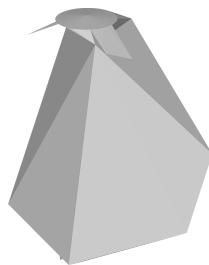
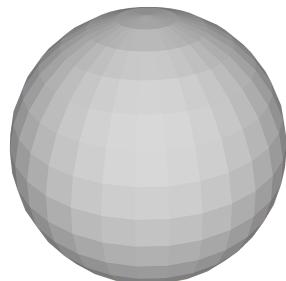
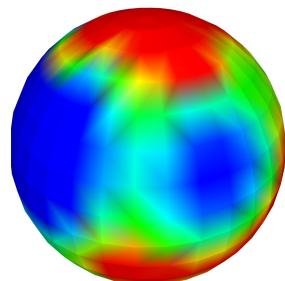
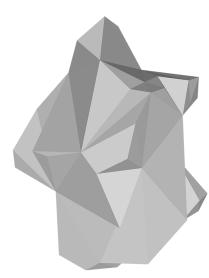
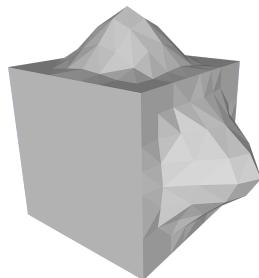
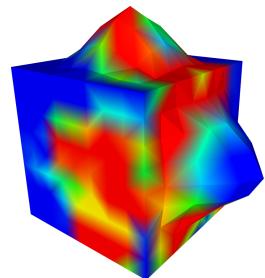
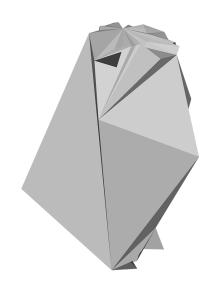
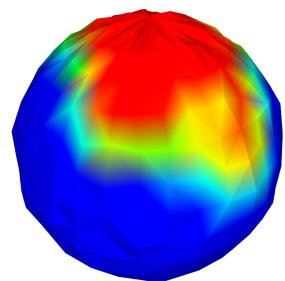
(a) `cube0.obj`(b) `sphere0.obj`(c) `cube1.obj`(d) `sphere1.obj`

Figura 5.8: **SphereCubes**, mapas de activación calculados con la distancia Hausdorff. Se pinta la malla con un gradiente desde el color rojo (menor distancia) al color azul (mayor distancia).

Nº Aristas	Distancia máxima media	Distancia media	Error máximo medio	Error medio
30K	0.0878 mm	0.0067 mm	0.1701 %	0.0130 %
50K	0.0593 mm	0.0045 mm	0.1149 %	0.0087 %
100K	0.0377 mm	0.0024 mm	0.0073 %	0.0047 %

Tabla 5.6: Distancia Hausdorff entre las mallas originales y las mallas reducidas, medidas en unidades absolutas y relativas.

Se observa que la diferencia entre cada reducción con los datos originales es mínima en general. Lógicamente, a mayor resolución la distancia entre las superficies variará menos entre sí, pero igual es válido notar que incluso con 30 000 aristas se posee una distancia máxima de diferencia entre mallas de menos de un milímetro, lo que corresponde con un 0.1701 % de diferencia relativa al cuadro delimitador de la malla. Esto quiere decir que las mallas son muy parecidas entre sí aún siendo reducidas drásticamente, lo que permite también la posibilidad de utilizar muestras de mucha menor resolución sin miedo a una gran pérdida de información.

Estimando tiempo de procesado de datos

Habiendo ya comprobado las funcionalidades base del *framework*, la interpretabilidad que provee el mismo y la factibilidad de utilizar mallas con menor resolución, ahora se desea obtener una métrica de comparación entre MeshCNN, MedMeshCNN y MeshCNN+ respecto al tiempo de procesado de una malla dependiendo de la resolución de la misma y la variante del *framework* que se está utilizando.

Nuevamente partiendo de la arquitectura de SHREC16, se adaptan las capas de *Pooling* para aceptar una malla 3D de una síntesis del pubis de 30 000, 50 000 y 100 000 aristas con la finalidad de evaluar el tiempo que toma procesar cada una y si se es capaz de entrenar a esas resoluciones. Se puede observar en la Tabla 5.7 las capas de *pooling* que se utilizaron con la misma proporción de aristas por capa de *pooling* respecto al número de aristas de entrada del modelo. El entrenamiento se realizó en el nodo de cómputo de NGPU “Dionisio” que posee 48 GB de memoria de vídeo, el máximo disponible en el clúster. El tiempo que tomó el modelo y su uso de memoria en cada *framework* se puede observar en la Tabla 5.8.

Como puede observarse, existe una gran diferencia entre el tiempo que toma cada *framework* y la memoria utilizada. MeshCNN es el más rápido pero es el que más memoria gasta. Tanto así que no fue posible estimar el tiempo de procesado de 100 000 aristas porque ocupaba más memoria de la disponible. MeshCNN+ encuentra un balance, siendo más lento pero consumiendo significativamente menos memoria, mientras que MedMeshCNN obtiene el

Aristas	<i>Pooling</i>
30K	24000, 18000, 12000, 7200
50K	40000 30000 20000 12000
100K	80000, 60000, 40000, 24000

Tabla 5.7: Resoluciones de *pooling* utilizadas por cada *framework* para las aristas de entrada.

Aristas	MeshCNN		MedMeshCNN		MeshCNN+	
	Tiempo	VRAM	Tiempo	VRAM	Tiempo	VRAM
30K	8.9040 s	13.5 GB	402.3333 s	2.710 GB	23 s	1.792 GB
50K	15.1664 s	42.982 GB	1100 s	4.488 GB	38 s	2.164 GB
100K	-	> 48 GB	4632.3333 s	8.976 GB	80.3333 s	3.288 GB

Tabla 5.8: Comparativa en tiempo y espacio de los *frameworks* para procesar una malla.

peor desempeño, siendo el *framework* más lento con diferencia y con un uso ligeramente peor de memoria que MeshCNN+.

5.2.4. Experimentos con el Nódulo Óseo

Una vez realizados los experimentos anteriores, se procede a entrenar con los datos provistos. Como comentado, se generaron dos *datasets* para comenzar con las pruebas mientras se terminaban de preprocesar el resto de datos, pues el proceso no puede automatizarse del todo. Así mismo, por cuestiones de planificación, se decide utilizar MeshCNN para los experimentos, puesto que, aunque sea el *framework* que más consume memoria, es el más rápido y se posee cierta flexibilidad respecto al uso de memoria haciendo uso del nodo de NGPU con 30000 aristas.

Nodule-50-30K

Debido a las similaridades entre SHREC16 con el problema a resolver, se realizan distintas pruebas con variaciones de la arquitectura para determinar su desempeño en la detección del nódulo óseo de la síntesis del pubis. Los hiperparámetros se mantienen tal y como están en la Tabla 5.1 a excepción del tamaño del *batch* que se fija a 1, las aristas de entrada a 30000, las épocas se fijan a 50 manteniendo el mismo LR y 10 con decaimiento. Se mantiene el aumento de datos tal como se describe en la Tabla 5.2.

En la Tabla 5.9 se pueden observar los 5 modelos más representativos de las pruebas realizadas. El 1º y 2º modelo utilizan la misma arquitectura, salvo que el segundo incluye saltos residuales entre los bloques convolucionales, añadido siguiendo el razonamiento de modelos como ResNet50 que los em-

Capa		Nodule-50-30K-1	Nodule-50-30K-2	Nodule-50-30K-3
Convolución	<code>nfc</code>	64, 128, 256, 256	64, 128, 256, 256 1 salto residual	16, 16, 32, 64
Normalización	<code>norm</code>	Group, 16		
<i>Pooling</i>	<code>pool_res</code>	24000, 18000, 12000, 7200		20000, 15000, 8000, 7500
Densa	<code>fc_n</code>	100		
<i>Dropout</i>	<code>dropout</code>	0.0		
		Nodule-50-30K-4		Nodule-50-30K-5
Convolución	<code>nfc</code>	16, 16, 32, 32, 64	32, 64, 256, 256	
Normalización	<code>norm</code>	Instancia		
<i>Pooling</i>	<code>pool_res</code>	20000, 15000, 13000, 8000, 7500	20000, 15000, 10000, 5000	
Densa	<code>fc_n</code>	100	2042, 512, 128	
<i>Dropout</i>	<code>dropout</code>	0.0	0.5, 0.5, 0.5	

Tabla 5.9: Modelos Nodule-50-30K-*, Arquitectura.

plean y se sabe que funcionan bien para la detección de imágenes. El 3º modelo incluye una estructura menos densa y con unas capas de *pooling* con mayor reducción de aristas con la idea de concentrar la información que la red necesite para detectar la característica, el 4º modelo utiliza una red con más convoluciones y el 5º modelo utiliza más capas totalmente conectadas y el uso de *dropout* con la idea de potenciar el aprendizaje, estos dos utilizan también normalización de instancia pues obtuvo mejores resultados que la normalización de grupo para la misma estructura y misma semilla.

Modelo	<i>Accuracy</i>	<i>Precision</i>		<i>Recall</i>		<i>F1</i>		<i>L_{CE}</i>
		Presente	Ausente	Presente	Ausente	Presente	Ausente	
1	0.50	0.50	0.50	0.40	0.60	0.44	0.55	1.420
2	0.50	0.50	0.50	0.20	0.80	0.29	0.62	0.697
3	0.30	0.33	0.25	0.40	0.20	0.36	0.22	1.475
4	0.50	0.00	0.50	0.00	1.00	0.00	0.67	0.693
5	0.70	0.75	0.67	0.60	0.80	0.67	0.73	0.621

Tabla 5.10: Modelos Nodule-50-30K-*, resultados en el conjunto de test

En la Tabla 5.10 se pueden observar los resultados de los 5 modelos mencionados, puede verse que los modelos 1 a 4 no lograron aprender, pues se mantuvieron con un *accuracy* de 0.50 o menos, con lo único notable siendo que los modelos tienden a predecir que las muestras tienen el nódulo óseo ausente. El modelo Nodule-50-30K-5 a diferencia, si logró aprender, pues obtiene un *accuracy* de 0.70, adicionalmente no posee una tendencia marcada para clasificar las muestras con nódulo presente o ausente, poseyendo una *precision* de 0.75 y 0.67, *recall* de 0.60 y 0.80 respectivamente, para mayor claridad se puede observar la matriz de confusión en 5.11, donde se aprecia que es una clasificación balanceada. Se puede observar en las curvas de entrenamiento, Figura 5.9, que el aprendizaje ha sido ruidoso pero con tendencia ascendente, aunque se observan indicios de sobreajuste en la gráfica del error, Subfigura 5.9b. Es necesario, notando el buen desempeño que se ha obtenido en las muestras reducidas, hacer uso de todos los datos

disponibles para explorar mejor la calidad del modelo actual.

		Real	
		Presente	Ausente
Predicción	Presente	3	1
	Ausente	2	4

Tabla 5.11: Nodule-50-30K-5, matriz de confusión.

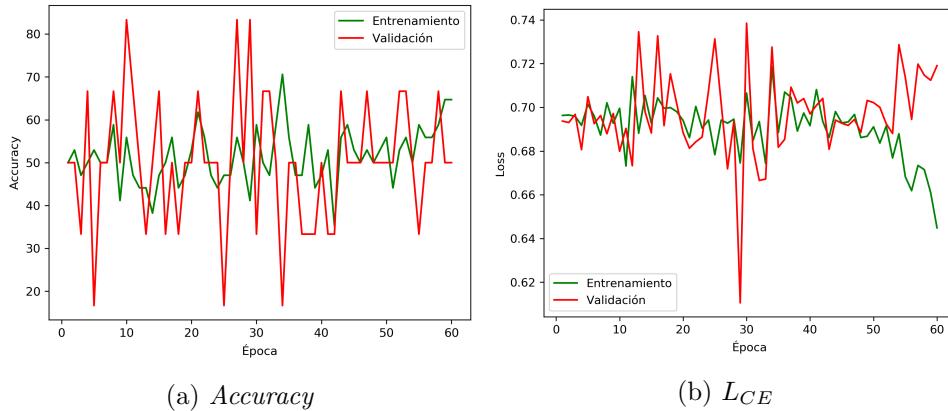


Figura 5.9: Modelo Nodule-50-30K-5. Curvas de aprendizaje de *accuracy* y error.

Nodule-98-30K

Una vez obtenido el modelo Nodule-50-30K-5, el cual para 50 huesos obtuvo la mejor *accuracy* de 70 %, se decide, entonces, ampliar el estudio para utilizar todos los huesos disponibles del *dataset* balanceado. Se reutiliza la misma estructura del modelo, ahora denominándose Nodule-98-30K-5rev.

Una vez realizado el entrenamiento con los mismos parámetros e hiperparámetros utilizados para Nodule-50-30K-5, se obtiene un *accuracy* de 98.4848 % y un error de 0.0665 para los datos de entrenamiento, pero para los datos de validación se obtiene 58.3333 % de *accuracy* y 1.6335 de error. En las curvas de aprendizaje del modelo, Figura 5.10, se puede observar claramente un ejemplo de sobreentrenamiento u *overfitting*, sobre todo en la gráfica de la función de pérdida (5.10b). Las curvas de entrenamiento y validación comienzan juntas pero a partir de la época 20 divergen, con los valores del conjunto de entrenamiento mejorando y los valores del conjunto de validación empeorando. Esto indica que el modelo no está logrando generalizar los datos de entrenamiento y solo los está “memorizando”, lo que desmejora su rendimiento con datos no vistos, en este caso los de validación. No sorprende, entonces, que al procesar los datos de test se obtiene un 55 %

de *accuracy* con un error de 4.248.

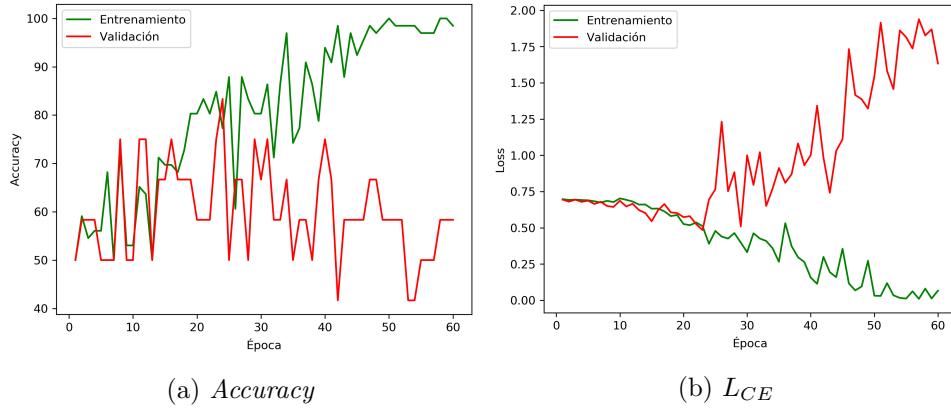


Figura 5.10: Curvas de aprendizaje para el *accuracy* y el error de entropía cruzada del modelo `Nodule-98-30K-5rev` presentando un claro caso de sobreentrenamiento.

Si bien no se obtuvieron resultados satisfactorios en esta primera aproximación, el hecho de que el modelo sobreentrene es una señal positiva pues indica que la red está funcionando, está aprendiendo demasiado bien los datos del conjunto de entrenamiento, lo cual es razonable dado el limitado número de muestras que se poseen.

Como se mencionó en la Sección 2.2.5, para afrontar este fenómeno se tiene que hacer uso de la regularización. Para esto se propone una batería de pruebas, modificando distintos parámetros de la arquitectura original para obtener un modelo que mejore los resultados para el *dataset* completo de nódulos óseos. En la Tabla 5.12 se observan los hiperparámetros considerados.

Hiperparámetro	Valores
Optimizador	optimizer
Aristas de entrada	ninput_edges
Tamaño de Batch	batch_size
Épocas	niter
Épocas con decaimiento de LR	niter_decay
LR Inicial	lr
Tipo de inicialización	init_type

Tabla 5.12: `Nodule-98-30K-GS-*`, hiperparámetros utilizados.

Se utilizará la variante del optimizador Adam llamada AMSGrad que mejora los problemas de convergencia del original. También se ha considerado el optimizador RMSProp pues también hace uso del LR adaptativo. Otro

parámetro que ayuda con el aprendizaje de las redes es la inicialización. Se sabe que al realizarla por medio de una distribución normal puede suceder que los pesos sean extremadamente grandes o, al revés, muy pequeños. Por eso se consideran los métodos Xavier y Kaiming, que intentan mantener un balance en las magnitudes de inicialización tomando en cuenta las capas del modelo. Se reduce también el número de épocas, esto se realiza para reducir el tiempo total de entrenamiento de todos los modelos.

Capa	Cantidad y/o Densidad	
Convolución <code>nfc</code>	32, 64, 256, 256	256, 128, 64, 32
Normalización <code>norm</code>	Instancia	
<i>Pooling</i> <code>pool_res</code>	20000, 15000, 10000, 7500	20000, 15000, 10000, 5000
Densa <code>fc_n</code>	2042, 512, 128	1024, 512, 128
<i>Dropout</i> <code>dropout</code>	0.25, 0.25, 0.25	0.5, 0.25, 0.125

Tabla 5.13: Nodule-98-30K-GS-*, arquitecturas utilizadas.

Respecto a la arquitectura, en la Tabla 5.13, se observan las variaciones utilizadas en el entrenamiento. Se utilizan densidades crecientes y decrecientes de las capas convolucionales. De esta manera, el modelo puede tener más información del nódulo óseo en las primeras capas del modelo o en las últimas. El *pooling* originalmente se mantiene como el modelo Nodule-98-30K-5rev pero se observó que ciertos modelos no podían reducir la malla a 5000 aristas, así que se aumentó hasta 7500. Las capas densas poseen también una versión alternativa con menos números de neuronas, ya que también se sabe que tener muchos parámetros en un modelo conlleva al sobreentrenamiento. Por último, se utilizan dos variantes del *dropout*, se utiliza un *dropout* constante de 0.25 en las tres capas densas, reducido de 0.5 y un *dropout* decreciente, pues se piensa que de esta manera se está regularizando de mejor forma la parte densa del modelo.

Alteración	Valor	
Mallas a aumentar	<code>num_aug</code>	20 40
Proporción de aristas a invertir	<code>flip_edges</code>	20 % 40 %
Escalar vértices no uniformemente	<code>scale_verts</code>	No Sí
Proporción de vértices a mover por superficie	<code>slide_verts</code>	20 % 40 %

Tabla 5.14: Nodule-98-30K-GS-*, aumento de datos utilizado.

Por último, se aplica el aumento de datos descrito en la Tabla 5.14. Se tiene el aumento original así como una versión más agresiva, para permitir una mejor generalización del modelo.

Tomando en cuenta todas las combinaciones posibles, esto es:

- 2 opciones de optimizador.

- 2 opciones de inicialización de pesos.
- 2 opciones de densidad de capas convolucionales.
- 2 opciones de cantidad de neuronas en las capas densas con *dropout*.
- 2 opciones de aumento de datos.

Se tienen entonces, un total de $2^5 = 32$ modelos distintos, los cuales fueron entrenados con el *dataset* en cuestión. De los 32 modelos, 4 obtienen el *accuracy* máximo de 70 % para el conjunto de test: **Nodule-98-30K-2, 7, 16 y 25**. Los parámetros de estos modelos pueden apreciarse en la Tabla 5.15 y sus resultados del conjunto de test en la Tabla 5.16. Los parámetros para los 32 modelos se pueden observar en la Tabla A.1 junto a sus resultados, que están en la Tabla A.2.

Modelo	Optimizador	Inicialización	Convolución	Densas + <i>dropout</i>	Aumento de datos
2	AMSGrad	Xavier	32, 64, 256, 256	1024, 512, 128; 0.5, 0.25, 0.125	20, 0.2, No, 0.2
7	AMSGrad	Xavier	256, 128, 64, 32	1024, 512, 128; 0.5, 0.25, 0.125	40, 0.4, Sí, 0.4
16	RMSProp	Xavier	32, 64, 256, 256	2048, 512, 128; 0.25, 0.25, 0.25	20, 0.2, No, 0.2
25	RMSProp	Kaiming	32, 64, 256, 256	2048, 512, 128; 0.25, 0.25, 0.25	40, 0.4, Sí, 0.4

Tabla 5.15: **Nodule-98-30K-GS-***, hiperparámetros de los mejores modelos

Centrándose en los parámetros, es notable encontrar que dos modelos utilizan AMSGard y otros dos RMSProp, los que usan AMSGard hacen uso de la inicialización Xavier y menos cantidad de neuronas densas aunque un modelo utiliza más cantidad de filtros de convolución y también utiliza más aumento de datos, lo cual tiene sentido para contrarrestar la mayor cantidad de parámetros en dicho modelo. Por parte de los modelos que utilizan RMSProp, ambos utilizan densidades crecientes con el mayor número de neuronas totalmente conectadas pero uno utiliza más aumento de datos que otro, también coincide con el único modelo que utiliza la inicialización Kaiming. De estos resultados se puede concluir que para este problema una red que tiene densidades crecientes de filtros de convolución junto con regularización Xavier y una cantidad de neuronas reducida con *dropout* logra aprender y detectar patrones en la compleja superficie de la síntesis del pubis.

Modelo	<i>Accuracy</i>	<i>Precision</i>		<i>Recall</i>		<i>F1</i>		<i>L_{CE}</i>
		Presente	Ausente	Presente	Ausente	Presente	Ausente	
2	0.70	0.64	0.83	0.90	0.50	0.75	0.62	0.580
7	0.70	0.64	0.83	0.90	0.50	0.75	0.63	0.580
16	0.70	0.64	0.83	0.90	0.50	0.75	0.62	0.616
25	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.676

Tabla 5.16: **Nodule-98-30K-***, mejores resultados en el conjunto de test.

Ahora analizando los resultados en sí con mayor detalle, los modelos que mejor valor de pérdida son el 2 y 7 con $L_{CE} = 0.580$ aunque poseen un sesgo en su clasificación, pues poseen un *recall* de 0.90 respecto a la presencia de

nódulo óseo, es decir, que el 90 % de lo que clasifica como nódulo presente es correcto, aún así sobreestima la existencia del nódulo pues posee un *precision* de 0.64, lo que quiere decir que no logra predecir correctamente la ausencia del nódulo, se puede observar la matriz de confusión en la Tabla 5.17. Este sesgo se comparte también con el modelo 16, lo que indica que en el espacio de soluciones, estos tres modelos han convergido en zonas similares. Por otro lado, el modelo 25 posee un *precision* y *recall* de 0.70, por lo que predice de igual forma la existencia y ausencia de nódulo, lo cual es evidente en su matriz de confusión, que se encuentra en la Tabla 5.18, siendo entonces el mejor modelo de los obtenidos puesto que es el más balanceado de todos en su clasificación. En la Figura 5.11 se puede verificar que, también, se ha logrado mitigar el sobreentrenamiento, ya que las curvas de entrenamiento y validación están juntas, decreciendo en el error y creciendo en el *accuracy*.

		Real	
		Presente	Ausente
Predicción	Presente	9	5
	Ausente	1	5

Tabla 5.17: Nodule-98-30K-GS-[2, 7, 16], matriz de confusión.

		Real	
		Presente	Ausente
Predicción	Presente	7	3
	Ausente	3	7

Tabla 5.18: Nodule-98-30K-GS-25, matriz de confusión.

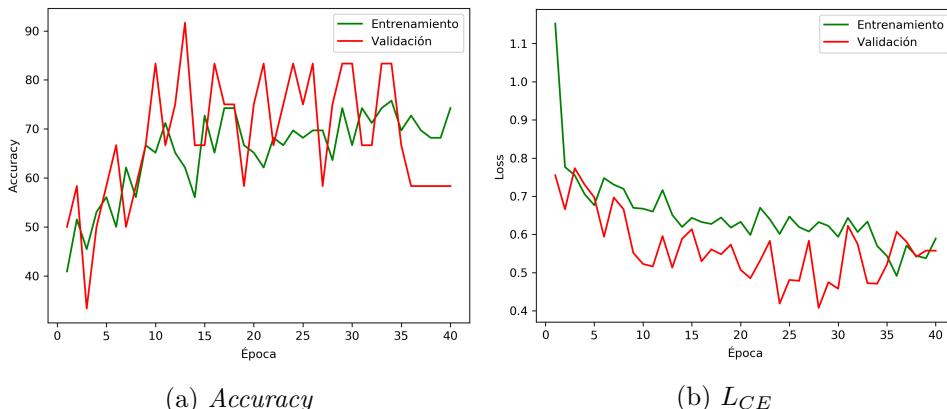


Figura 5.11: Modelo Nodule-98-30K-GS-25. Curvas de aprendizaje de *accuracy* y error.

Respecto a la interpretabilidad, interesa observar las partes de la síntesis

del pubis que utilizó el modelo para caracterizar la superficie. Para mayor claridad, se incluye la Figura 5.12 que muestra la orientación en la que se muestran los mapas de activación. En la Figura 5.13 se observan dos muestras con presencia de nódulo correctamente clasificado, mientras que en la Figura 5.14 se observan dos muestras incorrectamente clasificadas con ausencia de nódulo. De forma equivalente, en las Figuras 5.15 y 5.16 se observan muestras con ausencia de nódulo clasificadas correcta e incorrectamente.

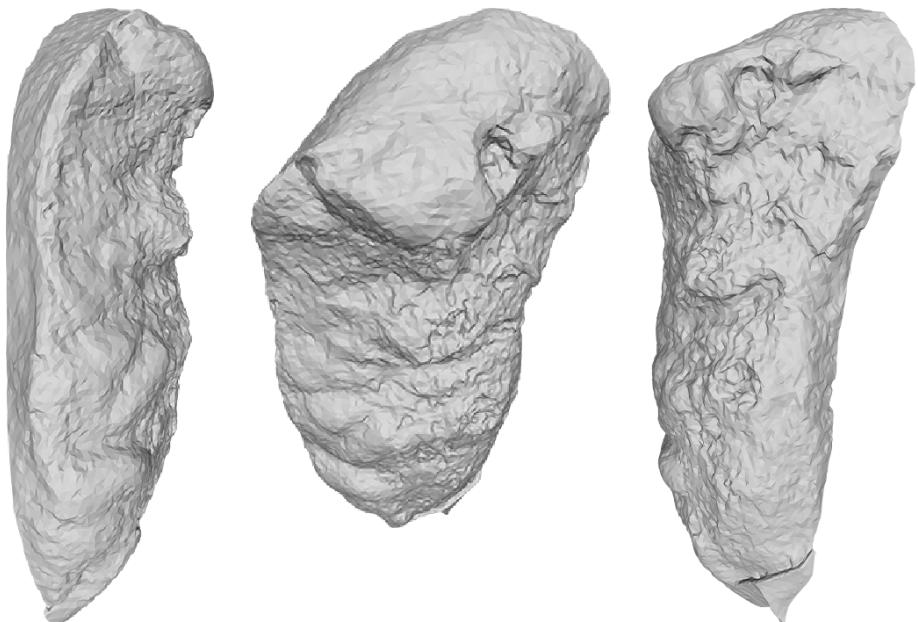


Figura 5.12: Ejemplo de orientación en la que se muestran los mapas de activación. De izquierda a derecha, se muestra el hueso visto desde la izquierda, visto desde el centro-arriba, y desde la derecha.

En general, se observa que los mapas de activación son bastante ruidosos, se encuentran moteados de colores tanto rojos como azules por la totalidad de la superficie, aunque se puede notar que el borde izquierdo de la sínfisis se encuentra por lo general mejor preservado que el borde derecho, que posee mayor gama de tonos azules, así como la parte superior que también se encuentra mejor preservada. Resulta curioso que el nódulo óseo como tal no parece tener preferencia para el modelo, en una gran parte posee colores azules indicando que la malla procesada difiere a la malla original en mayor medida. Observando las Figuras 5.14 y 5.16 se observa que las muestras mal clasificadas poseen activaciones diferentes, las muestras que poseen nódulo pero fueron clasificadas como ausentes poseen una activación similar a las muestras clasificadas correctamente, con el borde izquierdo teniendo una

fuerte influencia; mientras que en las muestras que poseen ausencia de nódulo pero fueron clasificadas como presente poseen unos mapas de activación mucho más ruidosos, donde pareciera que existe ambigüedad en la superficie pues el mapa de activación no parece estar localizado en ninguna zona en particular.

Estos resultados indican que la red no ha tomado preferencia por el nódulo óseo, y esto puede ser explicado en parte por el tamaño reducido de los datos, lo que permite que la red pueda aprender más detalles específicos de las muestras que le permitan obtener un alto *accuracy* inclusive en muestras no vistas anteriormente. Esto también explicaría por qué los mapas de activación son tan ruidosos, no se ha podido generalizar lo suficiente el aprendizaje para que se observen regiones claramente de mayor o menor interés. Otra explicación es que, además de lo mencionado, existan nuevas regiones de la sínfisis del pubis que estén asociadas a la presencia o ausencia del nódulo óseo, esto tiene soporte observando los falsos positivos, que poseen una activación elevada en la zona lateral izquierda de la sínfisis del pubis.

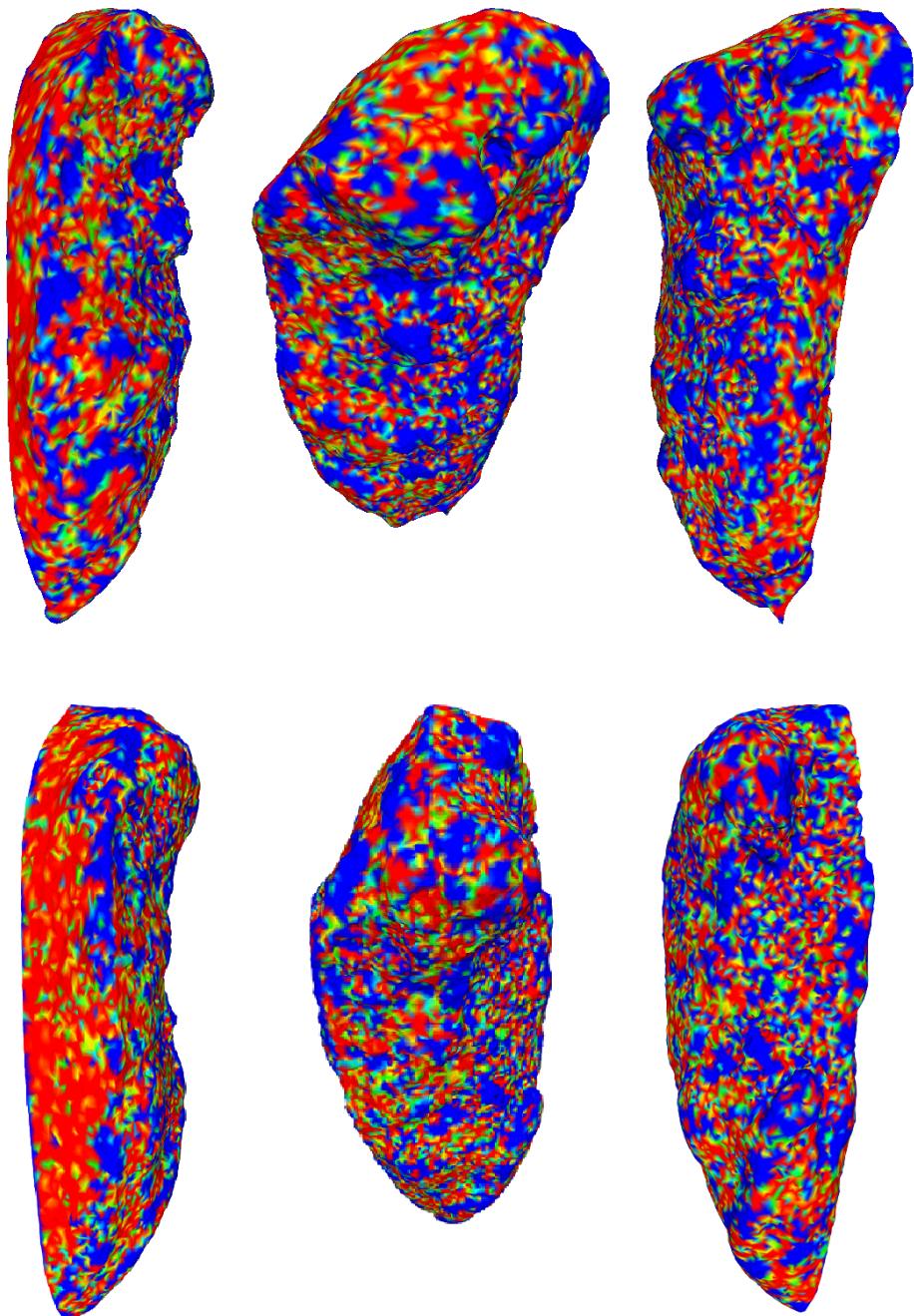


Figura 5.13: Síntesis del pubis con presencia de nódulo óseo clasificado correctamente

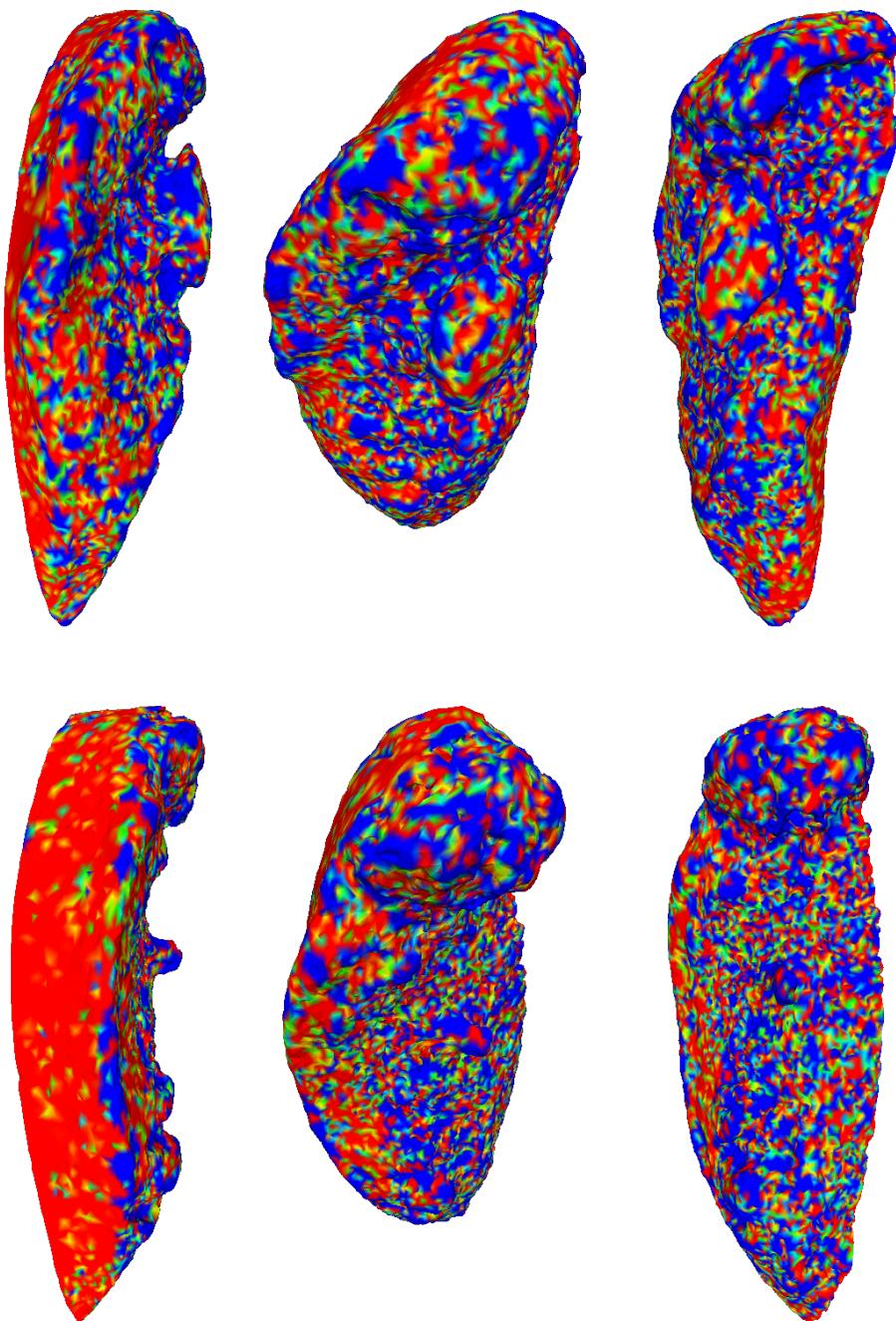


Figura 5.14: Síntesis del pubis con presencia de nódulo óseo clasificado como ausente

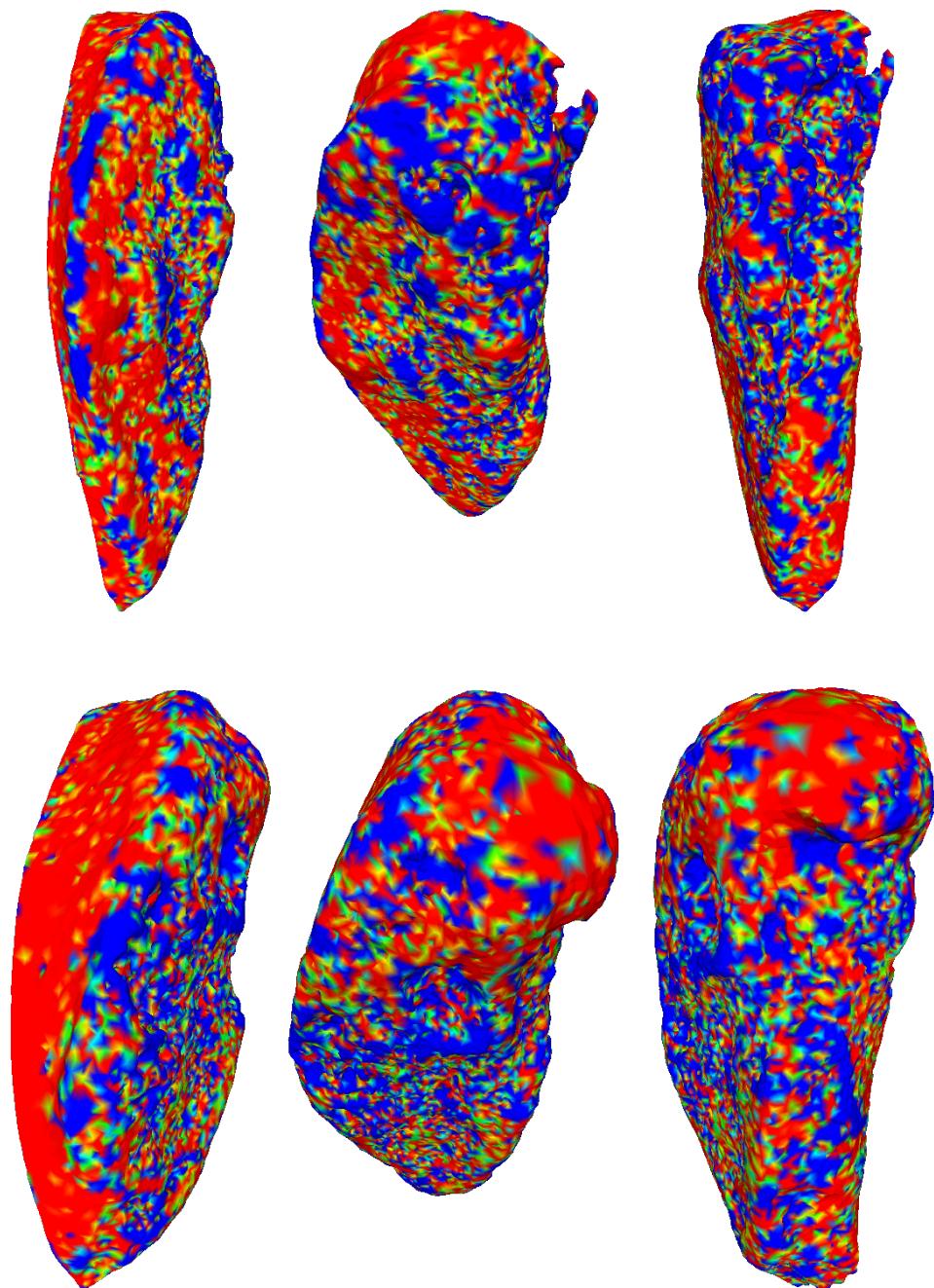


Figura 5.15: Síntesis del pubis con ausencia de nódulo óseo clasificado correctamente

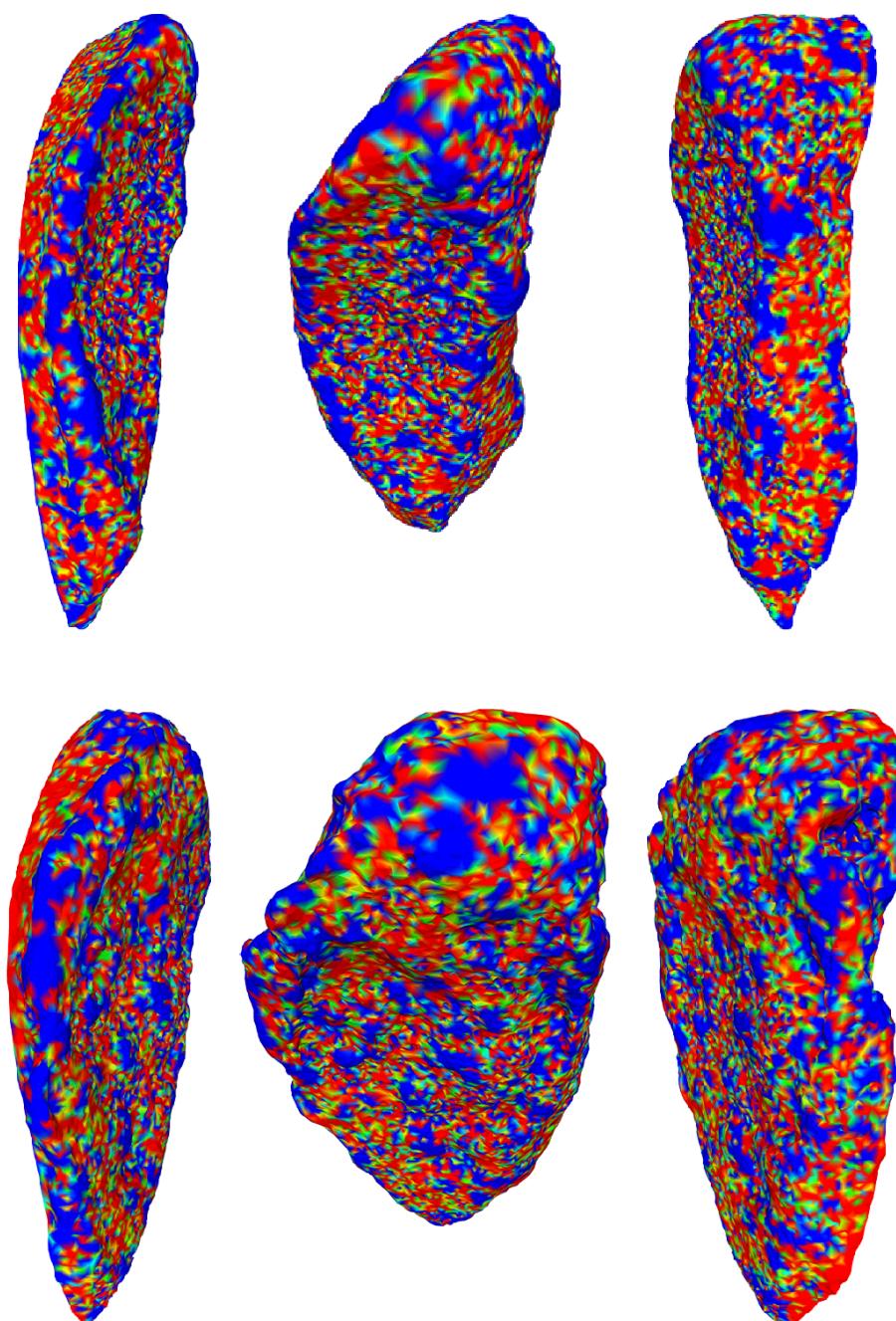


Figura 5.16: Síntesis del pubis con ausencia de nódulo óseo clasificado como presente

Capítulo 6

Conclusiones y Trabajos Futuros

La estimación de la edad es parte esencial del PB y, además, un problema complejo y de gran importancia para el campo de la AF debido a la subjetividad de los métodos manuales empleados actualmente por los expertos actualmente. Este TFG aborda la clasificación automática de los criterios morfológicos utilizados para la estimación de la edad, centrándose en la síntesis del pubis con el objetivo de obtener un modelo capaz de automatizar la extracción de dichos criterios, y poder asistir al experto humano en la toma de decisiones.

En primer lugar, se realizó un estudio pormenorizado de la literatura relativa a la estimación de edad de restos óseos, centrándose en la síntesis del pubis, así como del procesado de modelos 3D por medio de DL. De aquí se pudo observar y concluir que actualmente la AF sigue utilizando mayormente el método de Todd [10] o variantes, con innovaciones tecnológicas limitadas no ligadas a DL. Por parte del procesado de modelos 3D, se observa que no existe un consenso en la mejor forma de representar los datos y, por ende, existen múltiples maneras de representar la información. Para la problemática actual se seleccionan las mallas poligonales puesto que es un formato ubicuo y práctico que es utilizado ampliamente en la informática gráfica y por los propios antropólogos forenses. Por último, se concluye que no existe hasta el momento otra investigación que haya tomado el enfoque novedoso de clasificar automáticamente las características morfológicas del hueso directamente.

Se realizó un análisis de las propuestas metodológicas que hicieran uso de las mallas poligonales. Se trata de una de las representaciones más novedosas por lo que los estudios existentes son limitados. De los mismos, se selecciona MeshCNN y sus variantes, MedMeshCNN y MeshCNN+, como los *frame-*

works a utilizar para la creación de un prototipo que procese las mallas 3D de la síntesis del pubis. En concreto, después de un análisis que tomase en cuenta la complejidad espacial y temporal, se selecciona MeshCNN como el *framework* a utilizar debido a su buen rendimiento en cuanto al tiempo en consecuencia de la planificación temporal de este trabajo y aprovechando que se tuvo acceso a hardware que permite mitigar su alta complejidad espacial.

Centrándose en la creación de prototipos, se necesitó del desarrollo de diversas herramientas para el preprocesado de los datos, que implicó mayoritariamente la reducción de aristas de las mallas 3D con la menor pérdida de información topológica y el sellado de las mismas, tarea que se logró automatizar en gran medida. Tratándose de un prototipo se decidió centrarse en una característica recomendada por los expertos antropólogos por ser relevante y fácil de comprobar a simple vista: el nódulo óseo. Se generaron dos conjuntos balanceados de datos de 50 y 98 muestras en total. Inicialmente, con las 50 muestras se generaron 5 modelos inspirados en una arquitectura utilizada para clasificación de mallas, de estos 5, se obtiene un modelo que posee un 70 % de *accuracy* con una pérdida L_{CE} de 0.621 que tiende a sobreestimar la ausencia del nódulo óseo. Posteriormente, utilizando las 98 muestras se observa que ahora el modelo sobreentrena, obteniendo un 55 % de *accuracy* y 4.428 de L_{CE} . Para mitigar este problema, se proponen 32 modelos con diferentes modificaciones en los hiperparámetros para entrenar con dicho conjunto, lográndose al final, obtener 4 modelos que satisfactoriamente reducen el sobreentrenamiento y obtienen un *accuracy* de 70 %, notándose que ahora, en su mayoría los modelos tienden a sobreestimar la presencia del nódulo óseo, obteniendo un *precision* de 0.64 y un *recall* de 0.90, aunque el mejor modelo obtenido posee un balance que le otorga un F1 de 0.70. La observación de los mapas de activación indican que el modelo no se está centrando del todo en la zona del nódulo óseo, mostrando activaciones por la mayoría del hueso y sobre todo en el costado izquierdo. Se concluye que esto ocurre por una falta de datos o por la existencia de otra característica morfológica asociada a la existencia del nódulo óseo. A pesar de este resultado, puede concluirse que el modelo ha logrado aprender y diferenciar una superficie sumamente compleja, indicando el potencial que tiene este enfoque de cara a resolver el problema.

Por lo tanto, se concluye que se han completado satisfactoriamente los objetivos planteados, obteniéndose un modelo capaz de detectar características morfológicas por medio del uso de mallas poligonales de síntesis del pubis escaneadas. Todo el código se encuentra disponible en el repositorio de GitHub <https://github.com/RhinoBlindado/tfg>, a excepción de los escaneos de la síntesis ya que son datos confidenciales.

Siendo un proyecto utilizando una técnica novedosa y nunca antes utilizada,

existen varias ampliaciones lógicas que se pueden realizar al trabajo realizado. Por un lado, probar el modelo obtenido con las otras características del método de Todd de forma individual así como una aproximación multitarea que busque identificar varias características a la vez.

Por otro lado, se pueden explorar otros métodos que procesen modelos 3D. Recientemente se ha publicado otro *framework* denominado MeshNet++ [86], una ampliación de MeshNet [74] que obtiene resultados comparables o superiores a MeshCNN y, además, mitiga los diversos problemas y restricciones que posee MeshCNN al procesar las mallas.

Capítulo 7

Bibliografía

- [1] S. N. Byers, *Introduction to forensic anthropology*, 5th. ed. Routledge, 2017.
- [2] A. M. Christensen, N. V. Passalacqua y E. J. Bartelink, *Forensic Anthropology*. Elsevier Science & Technology, 2019.
- [3] Escuela Nacional de Antropología e Historia, *Antropología Física*, 2016. URL: <https://www.enah.edu.mx/index.php/pres-af-lic> (visitado 27-07-2022).
- [4] P. Mesejo, R. Martos, Ó. Ibáñez, J. Novo y M. Ortega, “A Survey on Artificial Intelligence Techniques for Biomedical Image Analysis in Skeleton-Based Forensic Human Identification,” *Applied sciences*, vol. 10, n.º 14, pág. 4703, 2020.
- [5] S. Khandare, S. Bhise y A. Shinde, “Age estimation from cranial sutures: CT scan study,” *Indian Journal of Basic and Applied Medical Research*, vol. 3, n.º 4, págs. 203-211, 2014.
- [6] M. Y. Işcan, S. R. Loth y R. K. Wright, “Age estimation from the rib by phase analysis: white males,” *Journal of Forensic Science*, vol. 29, n.º 4, págs. 1094-1104, 1984.
- [7] J. L. Buckberry y A. T. Chamberlain, “Age estimation from the auricular surface of the ilium: a revised method,” *American Journal of Physical Anthropology*, vol. 119, n.º 3, págs. 231-239, 2002.
- [8] S. M. Black y E. Ferguson, *Forensic anthropology: 2000 to 2010*. CRC Press, 2011.
- [9] H. M. Garvin y N. V. Passalacqua, “Current Practices by Forensic Anthropologists in Adult Skeletal Age Estimation,” *Journal of Forensic Sciences*, vol. 57, n.º 2, págs. 427-433, 2012.
- [10] T. W. Todd, “Age changes in the pubic bone,” *American journal of physical anthropology*, vol. 4, n.º 1, 1921.

- [11] S. Brooks y J. M. Suchey, “Skeletal age determination based on the os pubis: A comparison of the Acsádi-Nemeskéri and Suchey-Brooks methods,” *Human evolution*, vol. 5, n.º 3, págs. 227-238, 1990.
- [12] Y. S. Abu-Mostafa, M. Magdon-Ismail y H.-T. Lin, *Learning From Data: A Short Course*, 1.ª ed. AMLbook.com, 2012.
- [13] T. Mitchell, *Machine Learning*, 1.ª ed. McGraw Hill, 1997.
- [14] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*. MIT Press, 2016.
- [15] F. Chollet, *Deep Learning with Python*, 2.ª ed. Manning, 2021.
- [16] A. Krizhevsky, I. Sutskever y G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, n.º 6, págs. 84-90, 2017.
- [17] Legal Information Institute, *Daubert standard*. URL: https://www.law.cornell.edu/wex/daubert_standard (visitado 14-07-2022).
- [18] H. Ritchie, J. Hasell, E. Mathieu, C. Appel y M. Roser, “Terrorism,” *Our World in Data*, 2013. URL: <https://ourworldindata.org/terrorism> (visitado 12-07-2022).
- [19] H. Ritchie y M. Roser. “Natural Disasters.” (jun. de 2014), URL: <https://ourworldindata.org/natural-disasters> (visitado 12-07-2022).
- [20] R. Petrenko, “More than 1,600 unidentified bodies have been counted in Ukraine,” *Ukrayinska Pravda*, 2022. URL: <https://www.pravda.com.ua/eng/news/2022/07/4/7356275/> (visitado 12-07-2022).
- [21] N. Junquera, “Las huellas del horror franquista: 10.000 esqueletos recuperados en 20 años,” *El País*, 2022, Section: España. URL: <https://elpais.com/espana/2022-01-09/las-huellas-del-horror-franquista-10000-esqueletos-recuperados-en-20-anos.html> (visitado 12-07-2022).
- [22] R. S. Pressman, *Software engineering: a practitioner's approach*. Palgrave macmillan, 2005.
- [23] E. Alpaydd, *Introduction to Machine Learning*, 2.ª ed. MIT Press, 2009.
- [24] J. Han, *Data mining: concepts and techniques*, 3rd ed. Morgan Kaufmann, 2012.
- [25] Y. LeCun, Y. Bengio y G. Hinton, “Deep learning,” *Nature*, vol. 521, n.º 7553, págs. 436-444, 2015.
- [26] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, págs. 85-117, 2015.
- [27] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., 1995.

- [28] B. D. Ripley, *Pattern recognition and neural networks*. Cambridge University Press, 2007.
- [29] A. Bakiya, K. Kamalanand, V. Rajinikanth, R. S. Nayak y S. Kadry, “Deep neural network assisted diagnosis of time-frequency transformed electromyograms,” *Multimedia Tools and Applications*, vol. 79, n.º 15, págs. 11 051-11 067, 2020.
- [30] C. Decaro et al., “Machine learning approach for prediction of hematic parameters in hemodialysis patients,” *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 7, págs. 1-8, 2019.
- [31] Y. LeCun et al., “Backpropagation Applied to Handwritten Zip Code Recognition,” *Neural Computation*, vol. 1, n.º 4, págs. 541-551, 1989.
- [32] Y. Lecun, L. Bottou, Y. Bengio y P. Haffner, “Gradient-Based Learning Applied to Document Recognition,” *Proceedings of the IEEE*, vol. 86, n.º 11, págs. 2278-2324, 1998.
- [33] Prabhu, *Understanding of Convolutional Neural Network (CNN) — Deep Learning*, 2019. URL: <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148> (visitado 18-07-2022).
- [34] IBM, *What are Convolutional Neural Networks?* 2021. URL: <https://www.ibm.com/cloud/learn/convolutional-neural-networks> (visitado 17-07-2022).
- [35] L. Minfei, G. Yidong, Z. Chang, W. Zhi, E. Schlangen y B. Šavija, “Microstructure-informed deep convolutional neural network for predicting short-term creep modulus of cement paste,” *Cement and Concrete Research*, vol. 152, pág. 106 681, 2022.
- [36] Y. Wu y K. He, “Group normalization,” en *Proceedings of the European conference on computer vision (ECCV)*, 2018, págs. 3-19.
- [37] X. Glorot e Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” en *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop y Conference Proceedings, 2010, págs. 249-256.
- [38] K. He, X. Zhang, S. Ren y J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” en *Proceedings of the IEEE international conference on computer vision*, 2015, págs. 1026-1034.
- [39] SHREC2022 – 3DOR 2022. URL: <https://www.micc.unifi.it/3dor2022/index.php/shrec-2022/> (visitado 18-07-2022).
- [40] E. Ahmed et al., “A survey on deep learning advances on different 3D data representations,” *arXiv Preprint*, 2018.

- [41] A. S. Gezawa, Y. Zhang, Q. Wang y L. Yunqi, “A Review on Deep Learning Approaches for 3D Data Representations in Retrieval and Classifications,” *IEEE Access*, vol. 8, págs. 57 566-57 593, 2020.
- [42] R. S. Meindl y C. O. Lovejoy, “Ectocranial suture closure: A revised method for the determination of skeletal age at death based on the lateral-anterior sutures,” *American journal of physical anthropology*, vol. 68, n.º 1, págs. 57-66, 1985.
- [43] G. Acsádi, J. Nemeskéri y K. Balás, *History of human life span and mortality*. Akadémiai Kiadó, 1970.
- [44] R. W. Mann, R. L. Jantz, W. M. Bass y P. S. Willey, “Maxillary suture obliteration: a visual method for estimating skeletal age,” *Journal of Forensic Science*, vol. 36, n.º 3, págs. 781-791, 1991.
- [45] S. Ruengdit, D. T. Case y P. Mahakkanukrauh, “Cranial suture closure as an age indicator: a review,” *Forensic science international*, vol. 307, pág. 110 111, 2020.
- [46] F. Chiba et al., “Age estimation by multidetector CT images of the sagittal suture,” *International journal of legal medicine*, vol. 127, n.º 5, págs. 1005-1011, 2013.
- [47] K. L. Boyd, C. Villa y N. Lynnerup, “The use of CT scans in estimating age at death by examining the extent of ectocranial suture closure,” *Journal of forensic sciences*, vol. 60, n.º 2, págs. 363-369, 2015.
- [48] M. Y. İşcan, S. R. Loth y R. K. Wright, “Age estimation from the rib by phase analysis: white females,” *Journal of Forensic Science*, vol. 30, n.º 3, págs. 853-863, 1985.
- [49] L. Fanton, M.-P. Gustin, U. Paultre, B. Schrag y D. Malicier, “Critical study of observation of the sternal end of the right 4th rib,” *Journal of forensic sciences*, vol. 55, n.º 2, págs. 467-472, 2010.
- [50] K. M. Hartnett, “Analysis of age-at-death estimation using data from a new, modern autopsy sample—Part II: Sternal end of the fourth rib,” *Journal of forensic sciences*, vol. 55, n.º 5, págs. 1152-1156, 2010.
- [51] M. Blaszkowska, A. Flavel y D. Franklin, “Validation of the İşcan method in clinical MSCT scans specific to an Australian population,” *International Journal of Legal Medicine*, vol. 133, n.º 6, págs. 1903-1913, 2019.
- [52] C. O. Lovejoy, R. S. Meindl, T. R. Pryzbeck y R. P. Mensforth, “Chronological metamorphosis of the auricular surface of the ilium: a new method for the determination of adult skeletal age at death,” *American Journal of Physical Anthropology*, vol. 68, n.º 1, págs. 15-28, 1985.

- [53] C. G. Falsys, H. Schutkowski y D. A. Weston, “Auricular surface aging: worse than expected? A test of the revised method on a documented historic skeletal assemblage,” *American Journal of Physical Anthropology: The Official Publication of the American Association of Physical Anthropologists*, vol. 130, n.º 4, págs. 508-513, 2006.
- [54] E. Michopoulou, P. Negre, E. Nikita y E. F. Kranioti, “The auricular surface as age indicator in a modern Greek sample: a test of two qualitative methods,” *Forensic science international*, vol. 280, 246-e1, 2017.
- [55] C. Villa, J. Buckberry, C. Cattaneo y N. Lynnerup, “Reliability of suchey-brooks and buckberry-chamberlain methods on 3D visualizations from CT and laser scans,” *American journal of physical anthropology*, vol. 151, n.º 1, págs. 158-163, 2013.
- [56] P. Barrier et al., “Age at death estimation using multislice computed tomography reconstructions of the posterior pelvis,” *Journal of forensic sciences*, vol. 54, n.º 4, págs. 773-778, 2009.
- [57] E. Nikita, P. Xanthopoulou y E. Kranioti, “An evaluation of Bayesian age estimation using the auricular surface in modern Greek material,” *Forensic science international*, vol. 291, págs. 1-11, 2018.
- [58] J. V. Schanandore, M. Wolden y N. Smart, “The accuracy and reliability of the Suchey–Brooks pubic symphysis age estimation method: Systematic review and meta-analysis,” *Journal of Forensic Sciences*, vol. 67, n.º 1, págs. 56-67, 2022.
- [59] E. Priya, “Methods of skeletal age estimation used by forensic anthropologists in adults: a review,” *Forensic Res Criminol Int J*, vol. 4, n.º 2, págs. 104, 2017.
- [60] A. Wade, A. Nelson, G. Garvin y D. W. Holdsworth, “Preliminary radiological assessment of age-related change in the trabecular structure of the human os pubis,” *Journal of forensic sciences*, vol. 56, n.º 2, págs. 312-319, 2011.
- [61] C. Villa, M. N. Hansen, J. Buckberry, C. Cattaneo y N. Lynnerup, “Forensic age estimation based on the trabecular bone changes of the pelvic bone using post-mortem CT,” *Forensic science international*, vol. 233, n.º 1-3, págs. 393-402, 2013.
- [62] N. Lottering, M. S. Reynolds, D. M. MacGregor, M. Meredith y L. S. Gregory, “Morphometric modelling of ageing in the human pubic symphysis: Sexual dimorphism in an Australian population,” *Forensic science international*, vol. 236, 195.e1-195.e11, 2014.

- [63] M. López-Alcaraz, P. M. Garamendi, I. A. Aguilera y M. B. López, “Image analysis of pubic bone for age estimation in a computed tomography sample,” *International journal of legal medicine*, vol. 129, n.º 2, págs. 335-346, 2015.
- [64] H. Biwasaka et al., “Three dimensional surface analyses of pubic symphyseal faces of contemporary Japanese reconstructed with 3D digitized scanner,” *Legal Medicine*, vol. 15, n.º 5, págs. 264-268, 2013.
- [65] C. Villa, J. Buckberry, C. Cattaneo, B. Frohlich y N. Lynnerup, “Quantitative analysis of the morphological changes of the pubic symphyseal face and the auricular surface and implications for age at death estimation,” *Journal of forensic sciences*, vol. 60, n.º 3, págs. 556-565, 2015.
- [66] D. E. Slice y B. F. Algee-Hewitt, “Modeling bone surface morphology: a fully quantitative method for age-at-death estimation using the pubic symphysis,” *Journal of forensic sciences*, vol. 60, n.º 4, págs. 835-843, 2015.
- [67] D. Stoyanova, B. F. Algee-Hewitt y D. E. Slice, “An enhanced computational method for age-at-death estimation based on the pubic symphysis using 3 D laser scans and thin plate splines,” *American journal of physical anthropology*, vol. 158, n.º 3, págs. 431-440, 2015.
- [68] D. Stoyanova, B. F. Algee-Hewitt, J. Kim y D. E. Slice, “A computational framework for age-at-death estimation from the skeleton: Surface and outline analysis of 3D laser scans of the adult pubic symphysis,” *Journal of forensic sciences*, vol. 62, n.º 6, págs. 1434-1444, 2017.
- [69] P. Villar, I. Alemán, L. Castillo, S. Damas y O. Cordón, “A first approach to a fuzzy classification system for age estimation based on the pubic bone,” en *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, 2017, págs. 1-6.
- [70] J. C. Gámez-Granados et al., “Automating the Decision Making Process of Todd’s Age Estimation Method from the Pubic Symphysis with Explainable Machine Learning,” *Information Science*, Preimpresión, 2022.
- [71] J. C. Gámez, D. García, A. González y R. Pérez, “Ordinal classification based on the sequential covering strategy,” *International Journal of Approximate Reasoning*, vol. 76, págs. 96-110, 2016.
- [72] A. Kotěrová, D. Navega, M. Štepanovský, Z. Buk, J. Brůžek y E. Cunha, “Age estimation of adult human remains from hip bones using advanced methods,” *Forensic science international*, vol. 287, págs. 163-175, 2018.

- [73] A. Kotěrová, M. Štepanovský, Z. Buk, J. Brůžek, N. Techataweewan y J. Velemínská, “The computational age-at-death estimation from 3D surface models of the adult pubic symphysis using data mining methods,” *Scientific Reports*, vol. 12, n.º 1, pág. 10 324, 2022.
- [74] Y. Feng, Y. Feng, H. You, X. Zhao e Y. Gao, “Meshnet: Mesh neural network for 3d shape representation,” en *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, págs. 8279-8286.
- [75] Z. Wu et al., “3d shapenets: A deep representation for volumetric shapes,” en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, págs. 1912-1920.
- [76] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman y D. Cohen-Or, “MeshCNN: A Network with an Edge,” *ACM Transactions on Graphics*, vol. 38, n.º 4, 90:1-90:12, 2019.
- [77] Z. Lian et al., “Shape retrieval on non-rigid 3D watertight meshes,” en *Eurographics workshop on 3d object retrieval (3DOR)*, Citeseer, 2011.
- [78] L. J. Latecki y R. Lakamper, “Shape similarity measure based on correspondence of visual parts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, n.º 10, págs. 1185-1190, 2000.
- [79] L. Schneider, A. Niemann, O. Beuing, B. Preim y S. Saalfeld, “MedmeshCNN - Enabling meshcnn for medical surface models,” *Computer Methods and Programs in Biomedicine*, vol. 210, pág. 106 372, 2021.
- [80] L. M. Sangalli, P. Secchi y S. Vantini, “AneuRisk65: A dataset of three-dimensional cerebral vascular geometries,” *Electronic Journal of Statistics*, vol. 8, n.º 2, págs. 1879-1890, 2014.
- [81] A. Mandado. “Surface class segmentation with MeshCNN.” (2021), URL: <https://atomicsulfate.github.io/meshcnn-4-cadseg/>.
- [82] S. Koch et al., “ABC: A Big CAD Model Dataset For Geometric Deep Learning,” en *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [83] Blender Foundation, *Blender - a 3D modelling and rendering package*, 2018. URL: <http://www.blender.org>.
- [84] P. Cignoni, C. Rocchini y R. Scopigno, “Metro: measuring error on simplified surfaces,” en *Computer graphics forum*, Wiley Online Library, vol. 17, 1998, págs. 167-174.
- [85] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli y G. Ranzuglia, “MeshLab: an Open-Source Mesh Processing Tool,” en *Eurographics Italian Chapter Conference*, The Eurographics Association, 2008.

- [86] V. Singh, S. V. Sheshappanavar y C. Kambhamettu, “MeshNet++: A Network with a Face,” en *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, págs. 4883-4891.

Apéndice A

Nodule-98-30K: Parámetros y Resultados Completos

Modelo	Optimizador	Inicialización	Convolución	Densas + <i>dropout</i>	Aumento de datos
0	AMSGrad	Xavier	32, 64, 256, 256	2042, 512, 128; 0.25, 0.25, 0.25	20, 0.2, No, 0.2
1	AMSGrad	Xavier	32, 64, 256, 256	2042, 512, 128; 0.25, 0.25, 0.25	40, 0.4, Sí, 0.4
2	AMSGrad	Xavier	32, 64, 256, 256	1024, 512, 128; 0.5, 0.25, 0.125	20, 0.2, No, 0.2
3	AMSGrad	Xavier	32, 64, 256, 256	1024, 512, 128; 0.5, 0.25, 0.125	40, 0.4, Sí, 0.4
4	AMSGrad	Xavier	256, 128, 64, 32	2042, 512, 128; 0.25, 0.25, 0.25	20, 0.2, No, 0.2
5	AMSGrad	Xavier	256, 128, 64, 32	2042, 512, 128; 0.25, 0.25, 0.25	40, 0.4, Sí, 0.4
6	AMSGrad	Xavier	256, 128, 64, 32	1024, 512, 128; 0.5, 0.25, 0.125	20, 0.2, No, 0.2
7	AMSGrad	Xavier	256, 128, 64, 32	1024, 512, 128; 0.5, 0.25, 0.125	40, 0.4, Sí, 0.4
8	AMSGrad	Kaiming	32, 64, 256, 256	2042, 512, 128; 0.25, 0.25, 0.25	20, 0.2, No, 0.2
9	AMSGrad	Kaiming	32, 64, 256, 256	2042, 512, 128; 0.25, 0.25, 0.25	40, 0.4, Sí, 0.4
10	AMSGrad	Kaiming	32, 64, 256, 256	1024, 512, 128; 0.5, 0.25, 0.125	20, 0.2, No, 0.2
11	AMSGrad	Kaiming	32, 64, 256, 256	1024, 512, 128; 0.5, 0.25, 0.125	40, 0.4, Sí, 0.4
12	AMSGrad	Kaiming	256, 128, 64, 32	2042, 512, 128; 0.25, 0.25, 0.25	20, 0.2, No, 0.2
13	AMSGrad	Kaiming	256, 128, 64, 32	2042, 512, 128; 0.25, 0.25, 0.25	40, 0.4, Sí, 0.4
14	AMSGrad	Kaiming	256, 128, 64, 32	1024, 512, 128; 0.5, 0.25, 0.125	20, 0.2, No, 0.2
15	AMSGrad	Kaiming	256, 128, 64, 32	1024, 512, 128; 0.5, 0.25, 0.125	40, 0.4, Sí, 0.4
16	RMSProp	Xavier	32, 64, 256, 256	2042, 512, 128; 0.25, 0.25, 0.25	20, 0.2, No, 0.2
17	RMSProp	Xavier	32, 64, 256, 256	2042, 512, 128; 0.25, 0.25, 0.25	40, 0.4, Sí, 0.4
18	RMSProp	Xavier	32, 64, 256, 256	1024, 512, 128; 0.5, 0.25, 0.125	20, 0.2, No, 0.2
19	RMSProp	Xavier	32, 64, 256, 256	1024, 512, 128; 0.5, 0.25, 0.125	40, 0.4, Sí, 0.4
20	RMSProp	Xavier	256, 128, 64, 32	2042, 512, 128; 0.25, 0.25, 0.25	20, 0.2, No, 0.2
21	RMSProp	Xavier	256, 128, 64, 32	2042, 512, 128; 0.25, 0.25, 0.25	40, 0.4, Sí, 0.4
22	RMSProp	Xavier	256, 128, 64, 32	1024, 512, 128; 0.5, 0.25, 0.125	20, 0.2, No, 0.2
23	RMSProp	Xavier	256, 128, 64, 32	1024, 512, 128; 0.5, 0.25, 0.125	40, 0.4, Sí, 0.4
24	RMSProp	Kaiming	32, 64, 256, 256	2042, 512, 128; 0.25, 0.25, 0.25	20, 0.2, No, 0.2
25	RMSProp	Kaiming	32, 64, 256, 256	2042, 512, 128; 0.25, 0.25, 0.25	40, 0.4, Sí, 0.4
26	RMSProp	Kaiming	32, 64, 256, 256	1024, 512, 128; 0.5, 0.25, 0.125	20, 0.2, No, 0.2
27	RMSProp	Kaiming	32, 64, 256, 256	1024, 512, 128; 0.5, 0.25, 0.125	40, 0.4, Sí, 0.4
28	RMSProp	Kaiming	256, 128, 64, 32	2042, 512, 128; 0.25, 0.25, 0.25	20, 0.2, No, 0.2
29	RMSProp	Kaiming	256, 128, 64, 32	2042, 512, 128; 0.25, 0.25, 0.25	40, 0.4, Sí, 0.4
30	RMSProp	Kaiming	256, 128, 64, 32	1024, 512, 128; 0.5, 0.25, 0.125	20, 0.2, No, 0.2
31	RMSProp	Kaiming	256, 128, 64, 32	1024, 512, 128; 0.5, 0.25, 0.125	40, 0.4, Sí, 0.4

Tabla A.1: Nodule-98-30K-GS-*, hiperparámetros para cada modelo

Modelo	Accuracy	Precision		Recall		F1		L_{CE}
		Presente	Ausente	Presente	Ausente	Presente	Ausente	
0	0.50	0.00	0.50	0.00	1.00	0.00	0.67	0.744
1	0.50	0.00	0.50	0.00	1.00	0.00	0.67	0.701
2	0.70	0.64	0.83	0.90	0.50	0.75	0.62	0.580
3	0.60	0.58	0.62	0.70	0.50	0.64	0.56	0.669
4	0.50	0.00	0.50	0.00	1.00	0.00	0.67	0.753
5	0.45	0.43	0.46	0.30	0.60	0.35	0.52	0.694
6	0.50	0.00	0.50	0.00	1.00	0.00	0.67	0.725
7	0.70	0.64	0.83	0.90	0.50	0.75	0.63	0.580
8	0.50	0.50	0.50	0.20	0.80	0.29	0.62	0.707
9	0.45	0.45	0.44	0.50	0.40	0.48	0.42	0.704
10	0.50	0.00	0.50	0.00	1.00	0.00	0.67	0.763
11	0.45	0.00	0.47	0.00	0.90	0.00	0.62	0.847
12	0.65	0.71	0.62	0.50	0.80	0.59	0.70	0.648
13	0.45	0.44	0.45	0.40	0.50	0.42	0.48	0.950
14	0.60	0.75	0.56	0.30	0.90	0.43	0.69	0.659
15	0.45	0.40	0.47	0.20	0.70	0.27	0.56	0.767
16	0.70	0.64	0.83	0.90	0.50	0.75	0.62	0.616
17	0.45	0.00	0.47	0.00	0.90	0.00	0.62	0.818
18	0.45	0.33	0.74	0.10	0.80	0.15	0.59	0.675
19	0.50	0.00	0.50	0.00	1.00	0.00	0.67	0.688
20	0.40	0.33	0.43	0.20	0.60	0.25	0.50	0.920
21	0.55	1.00	0.53	0.10	1.00	0.18	0.69	0.661
22	0.50	0.00	0.50	0.00	1.00	0.00	0.67	0.730
23	0.45	0.33	0.47	0.10	0.80	0.15	0.59	0.775
24	0.65	0.67	0.64	0.60	0.70	0.63	0.67	0.864
25	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.676
26	0.50	0.50	0.50	0.20	0.80	0.29	0.62	0.688
27	0.45	0.33	0.47	0.10	0.80	0.15	0.59	0.784
28	0.55	0.67	0.53	0.20	0.90	0.31	0.67	0.744
29	0.60	0.75	0.56	0.30	0.90	0.43	0.69	0.630
30	0.55	0.57	0.54	0.40	0.70	0.47	0.61	0.879
31	0.65	0.62	0.71	0.80	0.50	0.70	0.59	0.864

Tabla A.2: Nodule-98-30K-*, resultados en el conjunto de test.

