



*ugr* | Universidad  
de Granada

TRABAJO FIN DE GRADO  
INGENIERÍA INFORMÁTICA

Estimación de la calidad de imágenes  
médicas 3D por medio de aprendizaje  
automático

---

**Autor**  
Brian Sena Simons

**Directores**  
Pablo Mesejo Santiago  
Enrique Bermejo Nievas



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

—  
Granada, Septiembre de 2023



# **Estimación de la calidad de imágenes médicas 3D por medio de aprendizaje automático**

Brian Sena Simons

**Palabras clave:** modelos 3D, nubes de puntos, imágenes biomédicas, estimación de la calidad, aprendizaje profundo, visión por computador.

## **Resumen**

En el campo biomédico, la visualización y análisis de estructuras 3D desempeña un papel fundamental. Sin embargo, la calidad de estas representaciones puede variar debido a diversos factores, como la adquisición, procesamiento y reconstrucción de las estructuras anatómicas a analizar. Para mejorar cada uno de estos pasos se hace necesaria una manera de cuantificar las distorsiones que puedan surgir. Este campo de investigación todavía no ha sido suficientemente explorado en el ámbito biomédico.

Este TFG presenta un sistema capaz de estimar la calidad de una representación 3D biomédica sin referencia, es decir, sin emplear un objeto no distorsionado de referencia. La propuesta adapta modelos de estimación de calidad de nubes de puntos genéricas al ámbito médico. Dicha representación es la más común y versátil para estructuras complejas, reconstrucciones y segmentaciones en la medicina [1].

De cara a la validación experimental, como no existen bases de datos públicas con imágenes biomédicas, se emplearon nubes de puntos generalistas, de personas, animales y objetos. De hecho, como parte de este TFG, se propone un conjunto de datos médicos sintéticos generados a partir de datos privados. Los datos consisten en un conjunto de tomografías computarizadas y modelos generados mediante el escaneo láser 3D de diferentes estructuras óseas. A este conjunto de datos se le aplican las distorsiones más comunes en las representaciones médicas 3D (compresión, ruido y otras). El valor real de calidad, de cada ejemplo generado, es estimado por medio de los mejores métodos, para cada tipo de distorsión, según la literatura.

A pesar de que los resultados obtenidos en este trabajo no dejan de ser preliminares, cabe mencionar que se alcanzó una correlación entre valores de calidad obtenidos y deseados del 88 %, sugiriendo que se trata de una línea de investigación tremadamente prometedora.



# **Quality Assessment of 3D medical images through machine learning**

Brian Sena Simons

**Keywords:** 3D models, point clouds, medical images, quality assessment, deep learning, computer vision.

## **Abstract**

In the biomedical field, the visualization and analysis of 3D structures play a fundamental role. However, the quality of these representations can vary due to various factors, such as the acquisition, processing, and reconstruction of the anatomical structures to be analyzed. To improve each of these steps, a way to quantify the distortions that may arise is necessary. This research area has not yet been sufficiently explored in the biomedical domain.

This Bachelor's thesis presents a system capable of estimating the quality of a 3D biomedical representation without reference, that is, without using an undistorted reference object. The proposal adapts quality estimation models from generic point clouds to the medical field. This representation is the most common and flexible for complex structures, reconstructions, and segmentations in medicine [1].

For experimental validation, as there are no public databases with biomedical images, generic point clouds of people, animals, and objects were used. In fact, as part of this Bachelor's thesis, a set of synthetic medical data generated from private data is proposed. The data consists of a collection of computerized tomographies and models generated through 3D laser scanning of different bone structures. This dataset is subjected to the most common distortions in 3D medical representations (compression, noise, and others). The real quality value of each generated example is estimated using the best methods for each type of distortion, according to the literature.

Although the results obtained in this work are still preliminary, it's worth mentioning that a correlation of 88% was achieved between obtained quality values and desired values, suggesting that this is an extremely promising line of research.



---

Yo, **Brian Sena Simons**, alumno de la titulación Grado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación**, con Pasaporte NX4L843F5, 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: Brian Sena Simons

Granada a 30 de agosto de 2023.



---

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

D. **Enrique Bermejo Nievas**, Profesor del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

**Informan:**

Que el presente trabajo, titulado *Estimación de la calidad de imágenes médicas 3D por medio de aprendizaje automático* ha sido realizado bajo su supervisión por **Brian Sena Simons**, 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 30 de agosto de 2023.

**Los directores:**

**Pablo Mesejo Santiago      Enrique Bermejo Nievas**



# Agradecimientos

Primeramente, me gustaría agradecer a mis tutores, Enrique Bermejo y Pablo Mesejo, por darme la oportunidad de desarrollar este proyecto con ellos. Agradezco la paciencia infinita y comprensión a la hora de resolver mis dudas. Segundo, agradezco a la propia Universidad de Granada por haberme dado la oportunidad de continuar mis estudios universitarios en tan distinguida casa de estudios.

Quiero agradecer también a mis padres, Joana Sena y Robert Netland, por la oportunidad de realizar mis estudios en España con todo sus apoyos. A mis compañeros de piso, en especial al recién graduado en Física Yllari Kay, que han estado ahí desde primero de carrera y me han ayudado en cada paso. En general, a todos mis amigos, incluido los de Brasil, por el cariño hacia mis obligadas ausencias.



# Índice general

<b>1. Introducción</b>	<b>7</b>
1.1. Definición del Problema . . . . .	7
1.2. Motivación . . . . .	11
1.3. Objetivos . . . . .	12
1.4. Planificación del proyecto . . . . .	13
<b>2. Fundamentos Teóricos</b>	<b>17</b>
2.1. Image Quality Assessment (IQA) . . . . .	17
2.2. Aprendizaje Automático y Profundo . . . . .	20
2.2.1. Aprendizaje Automático . . . . .	20
2.2.2. Aprendizaje Profundo . . . . .	21
2.2.3. Ensemble de modelos de Deep Learning . . . . .	27
2.3. Imágenes médicas y distorsiones . . . . .	28
<b>3. Estado del Arte</b>	<b>31</b>
3.1. Estado del arte de IQA . . . . .	32
3.2. Estado del arte de PCQA . . . . .	33
3.3. Estado del arte de IQA en imágenes médicas . . . . .	35
<b>4. Materiales y Métodos</b>	<b>37</b>
4.1. Materiales . . . . .	37
4.1.1. Conjunto de datos genéricos . . . . .	37
4.1.2. Conjunto de datos médicos . . . . .	40
4.2. Métodos . . . . .	42
4.2.1. Modelo NR 3D-QA . . . . .	42
4.2.2. Modelo VQA-PC . . . . .	43
4.3. Evaluación . . . . .	46
4.3.1. Etiquetado . . . . .	46
4.3.2. Métricas de similitud . . . . .	47
<b>5. Implementación y Experimentos</b>	<b>51</b>
5.1. Diseño Experimental . . . . .	51
5.1.1. Protocolo de validación experimental . . . . .	51
5.2. Resultados . . . . .	53

5.2.1. Experimentos NR3DQA . . . . .	53
5.2.2. Experimentos VQA-PC . . . . .	55
5.2.3. Experimentos finales . . . . .	57
5.3. Discusión de resultados . . . . .	61
<b>6. Conclusiones y Trabajos Futuros</b>	<b>63</b>
<b>7. Bibliografía</b>	<b>67</b>

# Índice de figuras

1.1.	Ejemplo de artefactos sobre imágenes DICOM. . . . .	9
1.2.	Ejemplo de visualización de un directorio DICOM. . . . .	12
1.3.	Objetivos de este proyecto. . . . .	13
2.1.	Visualización del problema de la métrica <i>Minkowski</i> . . . . .	18
2.2.	Visualización del hiperplano MSE de imágenes distorsionadas.	18
2.3.	Resumen de subproblemas de IQA. . . . .	19
2.4.	Ejemplo de aprendizaje supervisado. . . . .	21
2.5.	Ejemplo de aprendizaje no supervisado. . . . .	21
2.6.	Ejemplo gráfico de una red neuronal. . . . .	23
2.7.	Ejemplo de red convolucional para imágenes médicas . . . . .	24
2.8.	Representación visual de la operación de convolución . . . . .	25
2.9.	Ejemplo de operación de <i>max-pooling</i> con <i>stride</i> a 2x2. . . . .	25
2.10.	Red convolucional espaciotemporal <i>SlowFast</i> . . . . .	26
2.11.	Ejemplo gráfico sobre voxelización. . . . .	27
2.12.	Representación de métodos de <i>ensemble</i> . . . . .	28
2.13.	Ejemplo de tomografía computarizada y modelo 3D. . . . .	29
3.1.	Crecimiento de interés en el campo según <i>Scopus</i> . . . . .	31
4.1.	Ejemplo de conjuntos de datos SJTU. . . . .	37
4.2.	Ejemplo de conjunto de datos WPC. . . . .	38
4.3.	Ejemplo de conjunto de datos LS-PCQA. . . . .	39
4.4.	Ejemplo de nuestras imágenes médicas. . . . .	40
4.5.	Ejemplo de distorsiones generadas sobre clavículas. . . . .	41
4.6.	Ejemplo de distorsiones que se presentan según la perspectiva.	45
4.7.	Ejemplo de las rotaciones que utiliza el modelo VQA-PC. . .	45
4.8.	Ejemplo detallado de las etapas del método de VQA-PC. . .	46
5.1.	Diagrama de secuencias del proyecto. . . . .	52
5.2.	Ejemplo de uso de K-fold para la búsqueda de hiperparámetros.	53
5.3.	Curvas de aprendizaje del test preliminar. . . . .	57
5.4.	Ejemplo visual de los distintos métodos de fusión a comparar.	58

- 5.5. Ejemplo de correspondencia de pendiente entre valores inferiores (sin normalizar) y los valores reales de las etiquetas. . . . 61

# Índice de tablas

1.1.	Planificación temporal inicial del proyecto. . . . .	14
1.2.	Planificación resultante del proyecto. . . . .	14
1.3.	Total de horas y días trabajados. . . . .	15
1.4.	Estimación final de coste del proyecto. . . . .	15
3.1.	Tablas estado del arte FR-IQA. . . . .	33
3.2.	Tablas estado del arte NR-IQA. . . . .	33
3.3.	Estado del arte de modelos NR-PCQA . . . . .	35
4.1.	Ejemplo de distorsiones en SJTU. . . . .	38
4.2.	Tabla de métricas para generación de etiquetas. . . . .	47
4.3.	Correlación de métricas sintéticas. . . . .	47
4.4.	Comparativa entre funciones de normalización. . . . .	48
5.1.	Resultados de prueba preliminar con SVM. . . . .	54
5.2.	Resultados de prueba preliminar NR3DQA. . . . .	54
5.3.	Resultado de mejoras sobre el método SVM . . . . .	55
5.4.	Hiperparámetros empleados en la experimentación preliminar. . . . .	55
5.5.	Descripción de la arquitectura ResNet50. . . . .	56
5.6.	Resultados de experimento preliminar. . . . .	56
5.7.	Valor medio sobre imágenes médicas. . . . .	59
5.8.	Desviación típica de los resultados médicos . . . . .	60
5.9.	Mediana de los valores sobre imágenes médicas. . . . .	60
5.10.	Resultados del método original sin pre-entrenar sobre imágenes reescaladas. . . . .	60
5.11.	Resultados en imágenes médicas reescaladas entrenando en LS-PCQA. . . . .	60



# Capítulo 1

## Introducción

### 1.1. Definición del Problema

Con la demanda incremental de aplicaciones, tanto para el entretenimiento como para el estudio biomédico, la información visual cada vez tiene un rol más importante. Sin embargo, la calidad de dicha información puede verse mermada con las etapas de adquisición, procesado, compresión, transmisión y reproducción. Es por ello que poder evaluar dicha calidad se ha vuelto un tema cada vez más importante [2-4]. Por lo tanto, este Trabajo Fin de Grado (TFG) se centra en el estudio de la evaluación de la calidad de imágenes, en inglés *Image Quality assessment (IQA)* [5]. Se trata de un problema fundamental en el procesamiento de imágenes y la visión por computador [6-9], que hace referencia a la tarea de medir y cuantificar la calidad perceptual de una imagen, teniendo en cuenta factores como el contenido, la resolución, el contraste, las distorsiones visuales y la percepción humana. La mejora de estas técnicas suele estar altamente conectada con el avance en los estudios del sistema visual humano [10].

El problema de la evaluación de la calidad de la imagen se aborda mediante enfoques subjetivos y objetivos [10].

- Los enfoques subjetivos implican realizar experimentos perceptuales en los que se recopilan las opiniones y evaluaciones de los observadores humanos. Estos observadores pueden calificar las imágenes en términos de su calidad visual o realizar comparaciones entre diferentes versiones de una misma imagen. Con base a las respuestas recopiladas, se pueden establecer modelos y métricas que reflejen la calidad percibida por los humanos, también conocida como *mean opinion score, MOS*<sup>1</sup>.

---

<sup>1</sup> *Mean Opinion Score* o valor medio de opinión, consiste en la media de la opinión de diversas personas para establecer un valor de referencia.

- Los enfoques objetivos buscan desarrollar algoritmos y métricas que puedan estimar la calidad de la imagen sin intervención humana. Estos enfoques se basan en características y propiedades visuales extraídas automáticamente a partir de la imagen, que se utilizan para calcular una puntuación de calidad. Estas características pueden incluir medidas de nitidez, contraste, estructura, color, distribución de texturas y otros aspectos relevantes para la percepción visual.

La elección entre enfoques subjetivos u objetivos depende del contexto y los recursos disponibles. Los enfoques subjetivos son considerados como la referencia estándar para la evaluación de la calidad de la imagen, ya que capturan la apreciación humana. Sin embargo, estos enfoques pueden ser costosos y requieren de un número significativo de participantes. Por otro lado, los enfoques objetivos se pueden llegar a automatizar, haciendo que sean muy prácticos para grandes cantidades de datos y diversas aplicaciones.

No obstante, el objetivo del campo es desarrollar algoritmos y métricas que puedan proporcionar una estimación precisa y consistente de la calidad de la imagen, teniendo en cuenta tanto aspectos subjetivos como objetivos respecto a las distorsiones [11]. Y, de esta forma, poder evaluar y comparar diferentes métodos de adquisición, compresión, restauración o manipulación de imágenes teniendo en cuenta que el receptor final es el humano.

Para abordar el problema de la IQA de forma objetiva, se emplean diversas técnicas y enfoques [2, 5, 10]. Entre ellos se incluyen métodos basados en características, modelos de percepción visual, aprendizaje automático y técnicas de procesamiento de señales [12-14]. Uno de los enfoques más habituales consiste en utilizar características básicas de la imagen. Las características elementales de la imagen son por ejemplo el contraste, la nitidez, la exposición y la uniformidad del color [5, 10]. Estas características pueden ser cuantificadas mediante algoritmos de procesamiento de imágenes y proporcionar una estimación inicial de la calidad.

Por otro lado, los modelos de percepción visual intentan simular cómo el sistema visual humano percibe y evalúa la calidad de la imagen. Estos modelos se basan en el entendimiento de los mecanismos y procesos perceptuales del cerebro humano, y utilizan características visuales y estadísticas para calcular la calidad percibida [5, 14]. Buscan emular la forma en que los humanos responden a las imágenes en términos de su calidad visual [15, 16].

Finalmente, se suelen emplear algoritmos de aprendizaje automático para tratar de resolver el problema. Se intenta aproximar una función que a partir del conjunto de características extraídas pueda determinar la calidad de la imagen en una escala específica, generalmente en el rango de 0 a 10.

Entre las aplicaciones más comunes de los algoritmos de estimación de calidad se podrían citar las siguientes [17-19]: la comparativa entre algoritmos de

compresión (ya que permite elegir aquellos con menor pérdida de información), la generación de mapas de calidad<sup>2</sup> (permitiendo el estudio de métodos de reducción de ruido local) y la determinación de la calidad del servicio de transmisión o *quality-of-service (QoS)* (ya que permiten evaluar los errores de transmisión). Se podría incluso extender al pre-procesamiento de datos de entrenamiento o estimar la precisión de un modelo de predicción basado en la calidad de los datos [20].

El uso de algoritmos IQA se encuentra ampliamente difundido en el ámbito general de las imágenes 2D. Sin embargo, el número de métodos propuestos decrece al desplazarnos a tres dimensiones. Además, en el ámbito médico, la naturaleza de estas imágenes y las distorsiones que pueden presentar (véase Figura 1.1) implican una disminución en la precisión de los modelos cuando se aplican directamente sobre ellas [2].

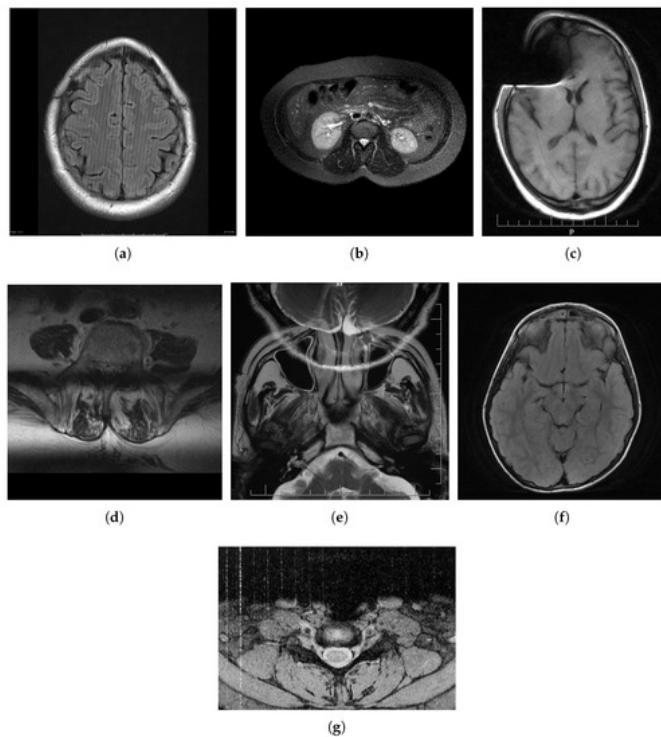


Figura 1.1: Ejemplo de artefactos sobre imágenes médicas [21]: (a) *herringbone*, (b) *ghosting*, (c) susceptibilidad magnética, (d) superposición de cortes, (e) *aliasing*, (f) efecto de Gibbs, y (g) *zipper*.

En la Figura 1.1 se aprecian las siguientes distorsiones:

(a) ***Herringbone***: causa variaciones en intensidad y superposición de bandas

---

<sup>2</sup>Nivel de calidad perceptual de diferentes regiones o píxeles de la imagen.

oscuras en imágenes debido a la transformada de Fourier.

- (b) ***Ghosting***: influenciado por reacciones físicas del paciente, factores ambientales y movimientos pulsátiles, como latidos cardíacos, puede causar distorsiones que se superponen con la imagen.
- (c) **Susceptibilidad magnética**: al ubicar tejidos en un campo magnético, su magnetización desigual debido a la susceptibilidad magnética causa distorsiones geométricas y variaciones de señal, acentuadas por implantes metálicos altamente susceptibles.
- (d) **Superposición de cortes**: pérdida de señal visible en la imagen debido a la adquisición desde múltiples ángulos. En esta distorsión, las secciones en los bordes tienen una intensidad de señal reducida y no crean un perfil de corte con bordes rectos.
- (e) ***Aliasing***: Los artefactos de aliasing surgen cuando el campo de visión es menor que la zona corporal capturada, generando superposición de estructuras regulares. En imágenes médicas, estas distorsiones pueden aparecer como un patrón de franjas o líneas que no corresponden correctamente a la anatomía real.
- (f) **Efecto de Gibbs**: también conocidos como artefactos de truncamiento o artefactos de anillos, son una serie de líneas en la imagen de resonancia magnética que aparecen paralelas al área donde ha ocurrido un cambio repentino e intenso en la intensidad de la señal.
- (g) ***Zipper***: área de píxeles alternantes claros y oscuros, presente en la dirección de codificación de frecuencia y que aparece en toda la serie de imágenes.

La mayoría de estas distorsiones, y combinaciones de ellas, no ocurren de forma natural en las imágenes habituales del problema IQA. Como consecuencia, es necesario diseñar adaptaciones de los modelos actuales y, por lo tanto, el número de métodos médicos existentes se reduce, con ninguno, al momento de escritura, aplicado directamente a la reconstrucción 3D. Dichas reconstrucciones suelen ser nubes de puntos [1] y las distorsiones anteriores afectan al resultado final.

Las nubes de puntos o conjunto de puntos arbitrarios extraídos de la superficie de un objeto de interés es una de las representaciones más comunes y flexibles: los objetos representados ya sea mediante volúmenes de vértices o mallas poligonales pueden muestrearse fácilmente en nubes de puntos. Además, la adquisición de datos a través de escaneo, segmentación u obtenidos de algoritmos de reconstrucción 3D generalmente proporcionan información geométrica en forma de un conjunto de puntos. Las superficies reconstruidas pueden ser aplicadas para: medición morfológica del grosor de la corteza o los

huesos, extracción de la línea central (esqueleto de curva) para traqueotomía o colonoscopía, particionamiento de superficies para clasificación de superficies corticales o anatómicas, así como registro y correspondencia de formas de tumores o huesos carpales [1].

Es por ello que se propone investigar específicamente el uso de métodos tridimensionales para el ámbito biomédico, aplicado a las reconstrucciones y visualizaciones volumétricas que se suelen emplear en medicina. El proyecto esta disponible en [https://github.com/CodeBoy-source/TFG\\_NRPCQA](https://github.com/CodeBoy-source/TFG_NRPCQA).

## 1.2. Motivación

En el caso del ámbito biomédico, dados los rápidos avances de las técnicas no invasivas y la gran cantidad de fabricantes de equipamientos, nació el estándar *DICOM* [22] en 1995 con el objeto de hacer que el intercambio de imágenes médicas se realizase de forma fácil, segura y con alta calidad. Este estándar pretendía permitir la integración con diversos sistemas, almacenar información extra en forma de metadatos y anotaciones, así como segmentaciones que facilitasen la reconstrucción 3D de diferentes regiones anatómicas.

Cada vez más frecuentemente se emplean volúmenes tridimensionales, como tomografías computarizadas o resonancias magnéticas en lugar de radiografías convencionales, porque proporcionan una visión más completa y detallada de la anatomía y las estructuras internas del cuerpo (véase Figura 1.2). Esta visualización tridimensional permite a los médicos y personal sanitario identificar con mayor precisión lesiones, enfermedades o anomalías, así como facilitar la planificación quirúrgica, entre otros [23-25].

No obstante, cabe mencionar que las distorsiones están muy presentes en las imágenes médicas [27]. Estas, a su vez, podrían afectar al volumen 3D que se puede generar a partir de las imágenes médicas, influyendo en el análisis y diagnóstico asociados. Por ejemplo, en [28] se estudiaron las razones por las que se suelen rechazar las radiografías y su relación con el diagnóstico final. Reveló que la mayoría de los rechazos se producen por errores de posicionamiento, valores inadecuados de exposición, artefactos y los problemas de cooperación del paciente. Además, no es difícil imaginar que una alta calidad de imagen médica tiene implicaciones significativas sobre el cuidado del paciente. Ya que la mala calidad de imagen puede provocar diagnósticos erróneos. Sin mencionar los elevados costes que supone realizar nuevas pruebas para conciliar las anteriores.

Por todo ello, las contribuciones relativas al IQA en el ámbito biomédico son claramente bienvenidas, resultando en una potencial reducción de costes (menos pruebas), de tiempo de consulta, y mejora en la calidad del diagnóstico médico.

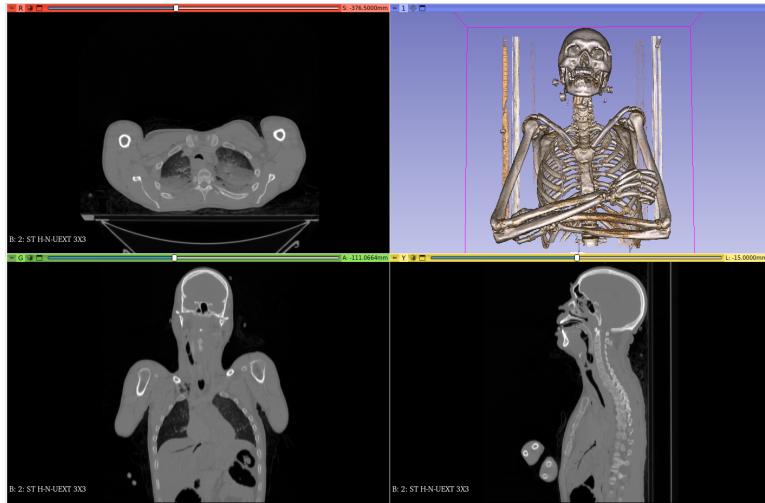


Figura 1.2: Ejemplo de visualización de un directorio *DICOM* empleando *Slicer3D* [26]. Se pueden observar las proyecciones axial (arriba izquierda), coronal (abajo izquierda), y sagital (abajo derecha). También se muestra una renderización volumétrica de los huesos (arriba derecha).

### 1.3. Objetivos

El objetivo principal de este Trabajo de Fin de Grado (TFG) consiste en desarrollar un **método adecuado para abordar al problema de la estimación de la calidad de imágenes médicas tridimensionales**. Este objetivo se puede descomponer en una serie de metas parciales:

1. Realizar una revisión exhaustiva del estado del arte para la estimación de calidad de imágenes 3D, así como de la calidad de imágenes médicas 3D en particular.
2. Estudiar las distorsiones de imagen más comunes, en general, y analizar los patrones de distorsión que afectan la calidad de las imágenes biomédicas.
3. Analizar pausadamente los enfoques de inteligencia artificial más prometedores que permitan abordar el problema planteado.
4. Generar un conjunto de datos sintético que permita validar los métodos empleados. Para ello, será necesario estudiar diferentes estrategias y métricas de evaluación objetivas.
5. Realizar un estudio experimental que permita validar los enfoques propuestos y extraer conclusiones sobre su aplicabilidad al problema.

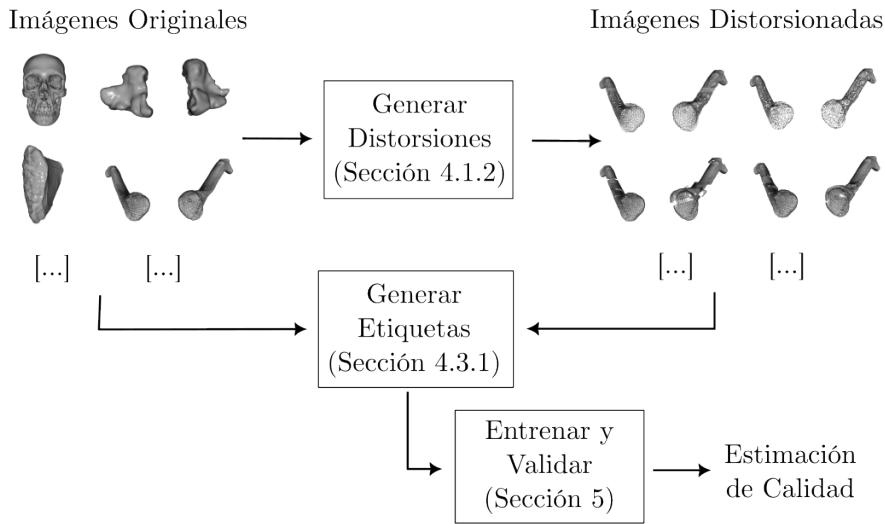


Figura 1.3: Objetivos de este proyecto.

## 1.4. Planificación del proyecto

Al planificar el proyecto, es fundamental tener en cuenta que el TFG tiene una carga de 12 créditos ECTS, donde cada crédito representa aproximadamente 25 horas de trabajo. En total, se estima que se necesitarán alrededor de 300 horas para llevar a cabo el proyecto. Considerando que el segundo cuatrimestre tiene aproximadamente 20 semanas, se requerirá dedicar al TFG unas 15 horas por semana, lo cual equivaldría a unas 3 horas diarias durante 5 días a la semana.

La naturaleza del proyecto no presenta una complejidad significativa en términos de su alcance y requisitos, y el equipo de trabajo tampoco incluye un numeroso grupo de personas cuya colaboración deba ser sincronizada, lo cual permite abordar su desarrollo a través de un enfoque de ciclo de vida en cascada [29]. No obstante, bajo este enfoque se evita retroceder en cualquiera de las fases del ciclo, y aunque se espera que el diseño y los requisitos del sistema sean estables, existe la posibilidad de realizar ajustes menores conforme se obtenga más información sobre el problema y los métodos. Es por ello que utilizamos una pequeña variante, la versión con retroalimentación.

Las fases del ciclo de vida son:

- Análisis de requisitos: Consiste en reuniones iniciales con los clientes, en este caso sería los directores del TFG. Se organiza el análisis bibliográfico del problema *IQA* y *PCQA*<sup>4</sup>, teniendo en cuenta un estudio previo de las distorsiones médicas.
- Diseño: Consiste en la investigación y selección de métodos conforme al análisis anterior, tanto para la resolución como la validación de la solución. Así como pruebas preliminares y diseño del software de experimentación.
- Implementación: Consiste en la adaptación de las técnicas encontradas, implementación de nuevas funcionalidades y generación de un conjunto de datos médicos.
- Pruebas: Realización de diversos experimentos de validación, tanto al la generación de las distorsiones como a los modelos y resultados.

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

Tabla 1.1: Planificación temporal inicial del proyecto.

La planificación inicial se muestra en la Tabla 1.1. Dicha planificación sufrió ciertos retrasos debido a que el alumno estaba realizando prácticas de empresa, tenía una asignatura y participaba de un curso de *Google* ofrecido por la universidad. Además, se esperaba que ocurrieran retrasos, sobre todo en la implementación, como se puede ver en la Tabla 1.2, dada la novedad de la propuesta y la dificultad del problema. En concreto, por ejemplo, el hecho de simular las distorsiones médicas fue un proceso iterativo y manual que llevó más tiempo de lo esperado.

Tarea	Semanas - Horas	Febrero				Marzo				Abril				Mayo				Junio				Julio			
		21	28	07	14	21	28	04	11	18	25	02	09	16	23	30	06	13	20	27	04	11	18	25	
Análisis de Requisitos	5 - 75																								
Diseño	4 - 60																								
Implementación	8 - 120																								
Pruebas	6 - 90																								

Tabla 1.2: Planificación resultante del proyecto.

Para realizar este proyecto se tuvo en cuenta los siguientes materiales: suscripción a *Google Colab Pro*, un portátil personal de gama media, *Google Drive 100GB* y otros gastos. Además, para el coste estimado, se asume un salario de 25€/hora, como para un investigador *senior* o responsable I+D de una empresa tecnológica en España.

<sup>4</sup>*Point cloud quality assessment* o estimación de calidad de nubes de puntos

Respecto al servidor GPU, con las especificaciones actuales de *Google*, se estima un coste aproximado de 10.000€. Se asume una amortización de 2 años, lo que implica un pago diario de 13.70€. El desglose total de los costes se puede ver en la siguiente Tabla 1.4.

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

Tabla 1.3: Total de horas y días trabajados.

Item	Costo
Salario	8 250.00€
Portátil de Gama Media	700.00€
Google Colab Pro	55.50€
Servidor GPU	2 109.8€
Google Drive 100GB	10.00€
Otros	300.00€
<b>Total</b>	11.425,3 €

Tabla 1.4: Estimación final de coste del proyecto.



## Capítulo 2

# Fundamentos Teóricos

### 2.1. Image Quality Assessment (IQA)

Existen tres subproblemas presentes en el ámbito de *IQA* [2-4]. Los primeros, son problemas donde tenemos acceso a la imagen original, que suponemos exenta de desperfectos, en la cual se pueden aplicar métodos basados en diferencia de características entre ambas, como puede ser al nivel del color de píxel posición a posición, y se denomina “*Full Reference*”(*FR*). La tarea, aparentemente sencilla, en realidad presenta una complejidad alta dada por la necesidad de codificar la percepción humana a la hora de calificar la calidad de una imagen [30], ya que métricas que miden distancias no suelen ser suficientes al no haber buena correlación entre la calidad percibida y el resultado de la métrica.

La mayoría de las veces no se menciona, pero al optar métodos de sensibilidad al error (distancias) se imponen un conjunto de suposiciones cuestionables. Primeramente, se asume la misma importancia para todas las señales de la imagen<sup>1</sup>, que la magnitud del error es lo único que determina la calidad, que el contenido de la imagen no afecta al resultado final tras aplicar una distorsión, y que si cambiamos el orden de las señales la medida de distorsión no es afectada. Lamentablemente, ninguna de estas suposiciones se cumplen [10] (véase Figuras 2.1 y 2.2, que utilizan distancias *Minkowski*, que puede considerarse como una generalización tanto de la distancia euclídea como de la distancia de Manhattan, y *MSE*, que se calcula como la media de la suma de las diferencias al cuadrado, respectivamente).

El siguiente subproblema es aquel donde tenemos algún tipo de información adicional, incompleta, respecto a la imagen original el momento de análisis de la calidad de la imagen final, denominados “*Reduced Reference*”(*RR*). La información extra puede incluir metadatos, parámetros de compresión,

---

<sup>1</sup>Con *señales* nos referimos a los diferentes canales de color RGB.

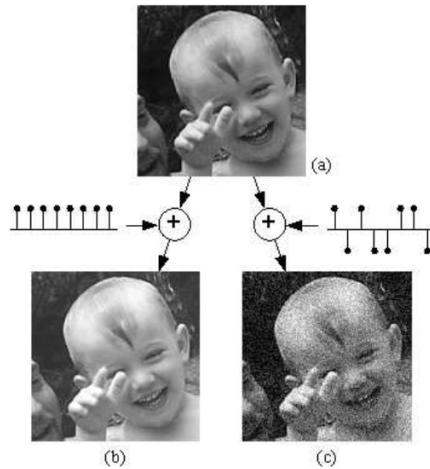


Figura 2.1: En este ejemplo, extraído de [5], vemos que sumar una constante positiva a una imagen de referencia (a) produce la imagen (b) que contiene la misma distancia *Minkowski* que (c), imagen fabricada por la misma constante pero permutando signo de forma aleatoria, resultando que la percepción final es que la imagen (c) es peor que la imagen (b).

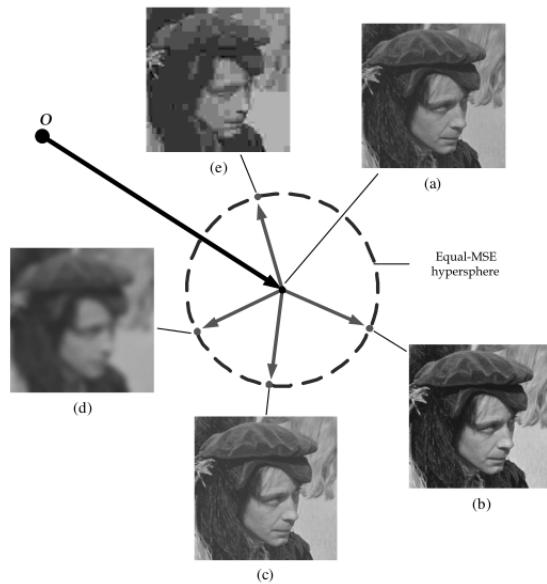


Figura 2.2: En este ejemplo, extraído de [10], la misma imagen distorsionada de distintas maneras resulta en la misma distancia, con valor  $MSE=181$ . No obstante, es evidente que algunas distorsiones producen efectos visuales más marcados que otras.

características estadísticas o extraídas de una región de interés específica.

Y por último, tenemos aquellos problemas donde desconocemos el origen y cualquier información respecto a la imagen inicial, denominados problemas “*No reference*” (*NR*). Estas métricas están exentas de cualquier información de referencia y se centran en capturar características generales de calidad.

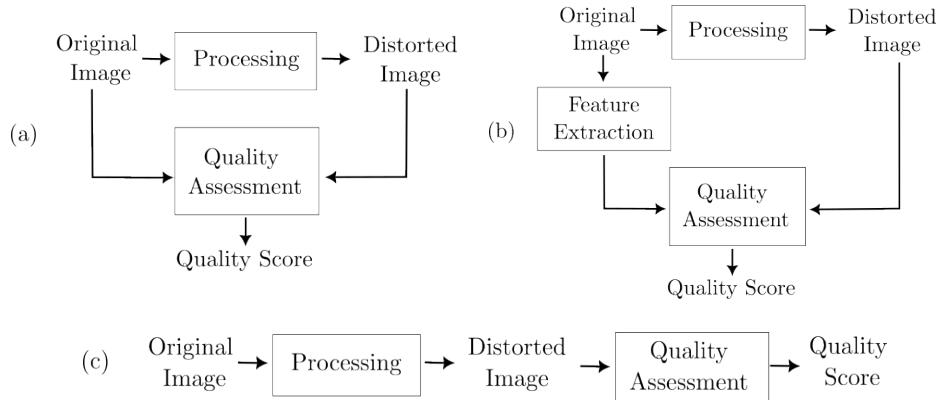


Figura 2.3: Resumen de subproblemas IQA. La imagen (a) representa el subproblema FR, la imagen (b) RR y la imagen (c) NR.

La evaluación de calidad de imagen sin referencia es, quizás, el problema más difícil en el análisis de imágenes. De cierto modo, el modelo debe ser capaz de evaluar la calidad de cualquier imagen sin saber nada de la imagen “real”, original. Superficialmente parece “misión imposible”. No obstante, esa es una tarea sorprendentemente sencilla para el ser humano [10].

Para resolver problemas NR, debemos disponer de conocimientos de la naturaleza de las imágenes de las que tratamos y los efectos de las distorsiones. Lo que se denomina estadísticas de escena naturales (NSS, por sus siglas en inglés). Un ejemplo sería JPEG, un algoritmo de compresión que se codifica por bloques 8x8. Los efectos negativos de la compresión se representan por el difuminado entre bloques y los artefactos que generan. Entender estos efectos permite diseñar métricas específicas [31].

As veces resulta difícil describir las características de la imagen y los efectos de la distorsión. Es por ello que los métodos de aprendizaje profundo son cada vez más frecuentes y dan mejores resultados. Permitimos que sea la máquina la que aprenda las propiedades de la distorsión, su relación con el contenido y efecto sobre la percepción visual [32-34].

La complejidad del problema crece conforme nos desplazamos a las tres dimensiones. El analizar la calidad de los modelos 3D implica mayor nivel de dificultad dado que nos enfrentamos a dos grandes retos: La complejidad computacional de las operaciones y la escasez de bases de datos etiquetadas sobre objetos tridimensionales para entrenar y evaluar modelos.

Para las nubes de puntos, que representan una colección de puntos en un espacio tridimensional ( $x, y, z$ ) cada uno con un color asociado  $RGB^2$ , se pueden emplear métricas y algoritmos basándose en criterios como la densidad de puntos, la uniformidad, la precisión geométrica y la detección de artefactos. También se pueden considerar aspectos relacionados con la coherencia de los colores o texturas asociadas a los puntos [35-37]. Un enfoque común es la evaluación de calidad de una nube de puntos tridimensional mediante proyecciones 2D desde diferentes perspectivas [38-40]. De esta forma podemos tratar el problema como uno de *IQA* 2D reduciendo la complejidad computacional, pudiendo implementar métodos y soluciones ya existentes.

Teniendo en cuenta todas estas consideraciones, el presente TFG aborda la estimación, sin referencia, de calidad de imágenes médicas en espacio tridimensional.

## 2.2. Aprendizaje Automático y Profundo

### 2.2.1. Aprendizaje Automático

El aprendizaje automático [41] o *Machine Learning (ML)* es una de las ramas que compone lo que definimos como la inteligencia artificial (*IA*, por sus siglas en inglés). Permite a las computadoras aprender a partir de datos sin programación explícita. A través de algoritmos y modelos, pueden reconocer patrones, hacer predicciones y tomar decisiones basadas en información proporcionada.

En este caso hablamos de dar soluciones a problemas complejos sin solución analítica (o que resulta muy costoso hallarla), es decir, necesitamos que la computadora sea la que identifique los patrones en los datos y realice predicciones sobre ellos [42]. 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 [43].

Dependiendo de factores como las necesidades del problema, la naturaleza de los datos a utilizar o el objetivo a alcanzar, podemos encontrar distintos tipos de algoritmos de aprendizaje. En este documento se recogerán dos grandes grupos: aprendizaje supervisado y aprendizaje no supervisado. En el primero disponemos de un conjunto de datos anotados, es decir, con las salidas deseadas para cada ejemplo y en el segundo se espera que sea la máquina la que determine los patrones (véanse Figuras 2.4 y 2.5). En general se suelen aplicar las técnicas de *ML* sobre grandes conjuntos de datos sobre

---

<sup>2</sup>RGB son las siglas en inglés para rojo, verde y azul. Los colores se representan por tripletas de valores en escala 0-255 ó 0-1 que significan la cantidad que aporta cada color.

los cuales deseamos detectar los patrones subyacentes [44].

Puede observarse que dadas estas descripciones, el problema presente puede ser abordados mediante técnicas de *ML*: Tenemos datos de entrada (características extraídas de nubes de puntos distorsionadas) y una salida (valor de calidad). Además, existen conjuntos de datos públicos etiquetadas para distintos tipos de distorsiones. Así, estamos ante un problema de aprendizaje supervisado.

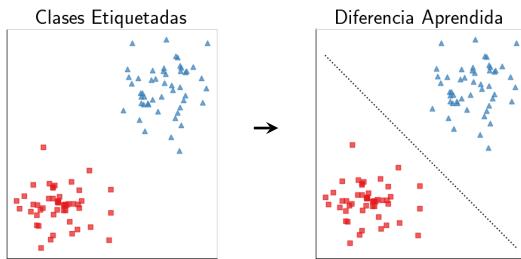


Figura 2.4: Ejemplo de aprendizaje supervisado. Vemos como a partir de un conjunto de clases etiquetadas aprendemos un hiperplano que las separa.

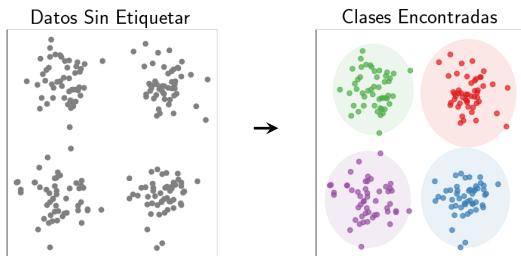


Figura 2.5: Ejemplo de aprendizaje no supervisado. Dado un conjunto de puntos aprendemos un conjunto de clases a partir de los patrones.

### 2.2.2. Aprendizaje Profundo

En el aprendizaje profundo o *Deep Learning (DL)*, a diferencia de los modelos anteriores donde tenemos un conjunto de variables extraídas por un humano experto, las características sobre la cual inferimos son obtenidas por el propio modelo automáticamente [45-47]. En términos generales, la extracción automática de características suele desempeñar mejores resultados en contra de las características manuales.

La mayoría de los modelos de *DL* son basados en múltiples capas jerárquicas de procesado de datos. Las más conocidas son las redes neuronales (ANN, por sus siglas en inglés), modelo bioinspirado que simula el funcionamiento de las neuronas del cerebro humano (abstracción simplificada) [48, 49].

A alto nivel, el funcionamiento de una red neuronal implica tres etapas principales: entrada, procesamiento y salida. En la etapa de entrada, se proporciona a la red neuronal un conjunto de datos o características que representan la información que se desea analizar o procesar. Estos datos de entrada se propagan a través de la red neuronal (*feedforward*). En la etapa de procesamiento, las neuronas reciben las entradas y realizan cálculos utilizando pesos y funciones de activación. Los pesos representan la importancia relativa de las diferentes entradas en el cálculo, y las funciones de activación determinan la salida de una neurona en función de su entrada. A medida que los datos se propagan a través de la red neuronal, las capas intermedias procesan y combinan las entradas, extrayendo características relevantes y creando representaciones internas cada vez más abstractas. Esto permite que la red neuronal aprenda y descubra patrones en los datos. Finalmente, en la etapa de salida, la red neuronal produce una respuesta o predicción basada en las características extraídas. Esto puede ser la clasificación de una imagen, la predicción de un valor numérico o cualquier otro resultado deseado. En esta última etapa se calcula el error de predicción respecto a la salida deseada con la función de pérdida y se ajusta los pesos respectivamente.

El aprendizaje de una red neuronal se logra mediante un proceso llamado entrenamiento. Donde de forma iterativa repetimos el proceso descrito anteriormente varias veces con distintos ejemplos. El conjunto de datos es muy relevante para el correcto aprendizaje. Debe de ser representativo, extenso y limpio de anormalidades ya que estaremos extrayendo características y relevancias a partir de ellos.

En definitiva, una red neuronal es en esencia una serie de ajustes de parámetros para lograr el resultado deseado. Estos incluyen ajuste de los pesos y sesgos iniciales, selección de las funciones de activación, como las más utilizadas sigmoide o ReLU, de una función de pérdida y un optimizador, encargado de determinar como ajustar los pesos según el error obtenido en cada fase del entrenamiento. No obstante, existe un fenómeno denominado sobreentrenamiento o *overfitting*. Ocurre cuando hay un sobreajuste de los parámetros hacia los datos de entrenamiento, disminuyendo la capacidad de generalización del modelo. Informalmente, es como decir que el modelo ha memorizado los resultados y, por ello, con datos nunca vistos posee errores substancialmente altos. Para lidiar con estos problemas se deben elegir también formas de regularización del modelo, es decir, restricciones sobre el entrenamiento para evitar el sobre ajuste.

## Redes Convolucionales

Las redes convolucionales o *convolutional neural network* (CNN) [53, 54] son un tipo de arquitectura de redes neuronales diseñadas específicamente para el

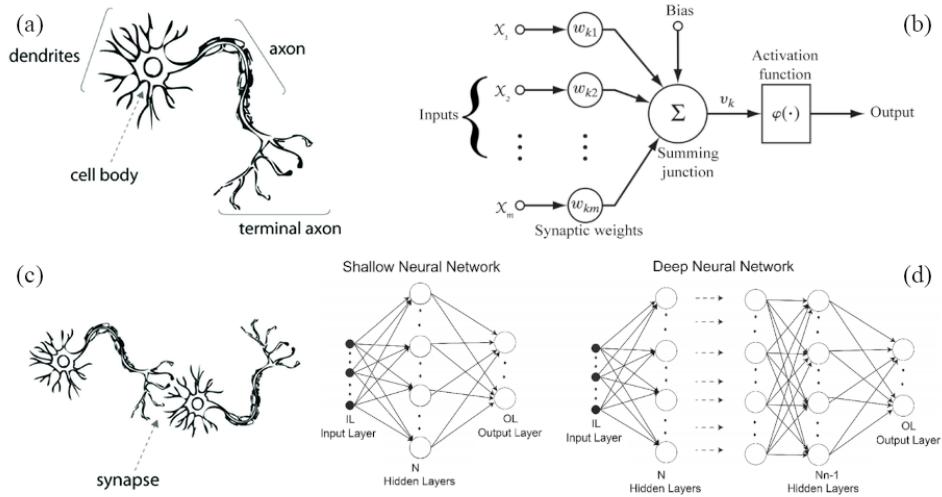


Figura 2.6: Ejemplo gráfico de una red neuronal [50-52]. (a) y (b) muestran una neurona biológica y una artificial, respectivamente. (c) visualiza la sinapsis (o proceso mediante el cual las neuronas se comunican entre sí para transmitir información). (d) muestra dos redes neuronales artificiales: a la izquierda, una red neuronal superficial (*shallow*), con una única capa oculta; a la derecha, una red neuronal profunda (*deep*), con múltiples capas ocultas.

procesamiento de datos estructurados en forma de matrices, como imágenes. Se ha descubierto que son aplicables para el procesado de texto, sonidos y, recientemente, a superficies tridimensionales. Utilizan capas convolucionales que aplican filtros a regiones locales de la entrada para extraer características relevantes. En la Figura 2.7 podemos ver un ejemplo de esquema jerárquico de extracción de características para el diagnóstico médico a partir de una radiografía. Se puede observar que, a diferencia de una ANN, existen dos capas adicionales: capas convolucionales y capas de *pooling*.

### Capas convolucionales

Para simplificar la explicación, la realizaremos sobre imágenes 2D. Una capa convolucional es encargada de realizar la operación de convolución sobre los datos de entrada. La convolución se refiere a una operación matemática que combina dos funciones para crear una tercera función. En este caso, se aplica una operación de convolución entre una matriz de entrada (como una imagen) y un filtro (*kernel*). La operación de convolución implica deslizar el filtro sobre la matriz de entrada, multiplicando los elementos coincidentes y sumándolos para obtener un único valor en la matriz de salida, conocida como mapa de características. Este proceso se repite en diferentes ubicaciones de la matriz

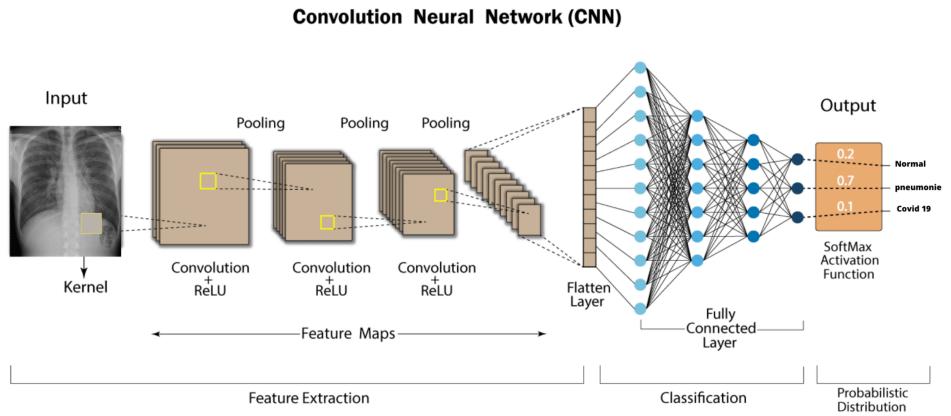


Figura 2.7: Red neuronal convolucional (CNN) aplicada a un problema de clasificación de imágenes biomédicas [55]. Se pueden identificar los principales bloques constitutivos de una CNN: capa convolucional, capa de pooling, función de activación, y capa totalmente conectada.

de entrada para generar el mapa de características completo. En la Figura 2.8 vemos una operación sobre la ubicación inicial de la imagen, esquina superior izquierda. La elección del siguiente trozo o *patch* de la imagen suele venir determinado por el paso o *stride*. Habitualmente se utiliza un *stride* de 1. Es decir, elegimos la matriz adyacente con distancia horizontal igual a 1 hasta llegar al final de esa fila y luego nos desplazamos 1 hacia abajo. Por medio de este proceso, la red es capaz de capturar dependencias temporales y espaciales en los datos con la aplicación de los filtros correspondientes.

Podemos observar en la Figura 2.8 que aplicar directamente el operador de convolución a una imagen resulta en una reducción del tamaño del mapa de activación debido a la naturaleza del operador. Sin embargo, esto no siempre es deseable. Para abordar este problema, se puede agregar relleno o *padding* a la imagen de entrada utilizando información existente en la misma. Esto garantiza que el mapa de activación tenga la misma dimensionalidad que la imagen original. Además, es posible reducir aún más la salida ajustando los saltos o *strides* del filtro de convolución mientras se recorre la imagen.

### Capa de pooling

El propósito principal de las capas de *pooling* es reducir la cantidad de parámetros y la complejidad computacional de la red, al tiempo que conservan las características más relevantes. Además, el *pooling* puede ayudar a hacer que la representación sea invarianta a pequeñas variaciones en la posición o el tamaño de los objetos en la imagen, lo que mejora la capacidad de generalización del modelo.

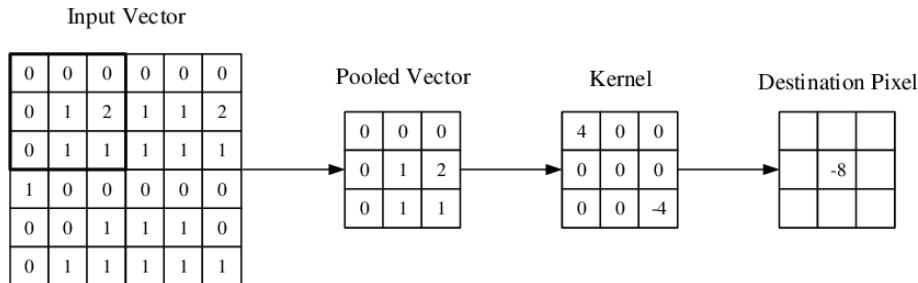


Figura 2.8: Representación visual de la operación de convolución sobre una imagen, extraída de [56].

En la Figura 2.9 vemos un operador de *pooling* común, el operador de valor máximo. También es habitual el uso del operador de valor medio y valor mínimo. 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.

Las capas convolucionales y de *pooling* trabajan en conjunto para procesar y extraer características. Dependiendo de la complejidad del problema, se puede ajustar el número de estas.

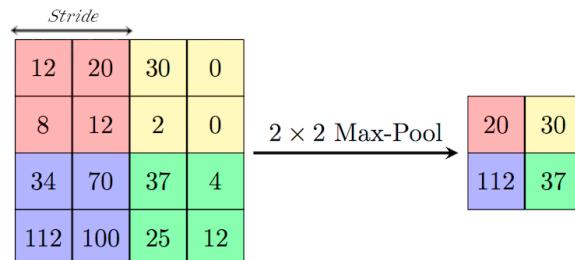


Figura 2.9: Ejemplo de operación de *max-pooling* con *stride* a 2x2.

### Capas totalmente conectadas

Las capas totalmente conectadas o *fully connected*, también llamadas capas densas o *dense*, son aquellas en las que todas sus neuronas están conectadas con todas las neuronas de la capa anterior y de la siguiente. Si bien existen modelos totalmente convolucionales, resulta común que las CNNs incluyan capas totalmente conectadas al final de la arquitectura. Estas capas forman una ANN clásica. 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.

### Aplicadas a Videos

Las redes convolucionales se pueden llegar a aplicar incluso a videos. Para ello, se puede utilizar una variante de las redes convolucionales llamada redes convolucionales 3D (3D CNNs) o redes convolucionales espaciotemporales. Estas redes están diseñadas específicamente para capturar tanto las características espaciales como las temporales presentes en los vídeos.

La principal diferencia entre una red convolucional tradicional y una 3D CNN es la adición de una dimensión temporal en las operaciones de convolución. En lugar de considerar solo imágenes individuales, se toman secuencias de imágenes (*frames*) para capturar la información temporal.

En este TFG se explora el uso de una 3D CNN capaz de analizar vídeos que pertenece a la familia que se conoce como *SlowFast networks* [57]. Están basadas en dos caminos de entrada de datos. Un conjunto de *frames* espaciados en el tiempo, *slow path*, para obtener información espacial y otro con todos ellos, *fast path*, para obtener información de movimiento (véase Figura 2.10).

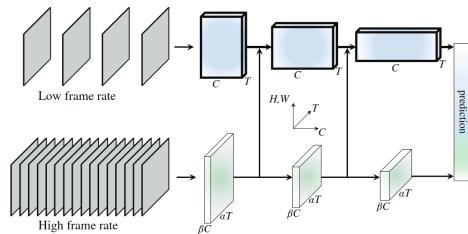


Figura 2.10: Ejemplo extraído de [57] para ilustrar la distinción entre la extracción espacial del camino “*slow path*” (camino superior) y de movimiento con el camino “*fast path*” (camino inferior).

### Aplicadas a nubes de puntos

De forma similar al caso de los videos, actualmente existen modelos de 3D CNNs capaces de procesar nubes de puntos al añadir una dimensión más que representa la profundidad de los píxeles. Sin embargo, la complejidad de diseño y tiempo de cómputo para estos modelos 3D crece enormemente. Esto es debido a que, habitualmente, las nubes de puntos están formadas por puntos dispersos en el espacio, en lo que denominamos datos sin estructura ni orden propio, y debemos mapear una operación de convolución que está basada en operaciones sobre datos ordenados y estructurados.

Aunque cada vez hay más métodos que se aplican directamente sobre la nube de puntos desde la publicación de *PointNet* [58], habitualmente se

intenta estructurar la información de las nubes de puntos mediante lo que denominamos véxeles [59]. La voxelización es el proceso de transformar una nube de puntos u otra representación tridimensional en una estructura discreta conocida como volumen voxelizado. Esto implica dividir el espacio tridimensional en una cuadrícula de véxeles y asignar valores a cada véxel según la información contenida en los datos originales. La voxelización proporciona una representación estructurada y discreta que permite el uso de técnicas específicas para volúmenes y facilita el procesamiento y análisis de datos 3D.

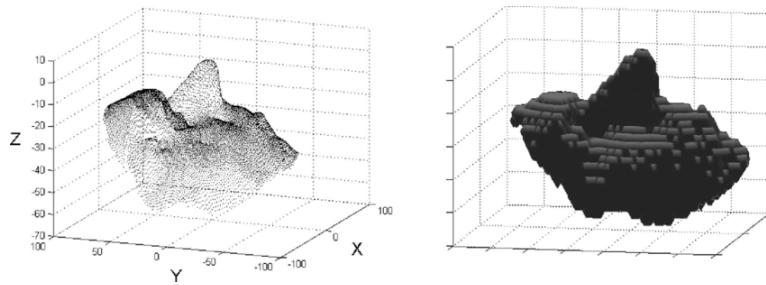


Figura 2.11: Ejemplo extraído de [59] que demuestra el resultado de transformar una nube de puntos sin estructura en una cuadrícula voxelizada.

### 2.2.3. Ensemble de modelos de Deep Learning

Un *ensemble*, en el contexto del aprendizaje automático, es una técnica que combina múltiples modelos de aprendizaje para mejorar la precisión y el rendimiento general de las predicciones. En lugar de depender de un único modelo, se crean múltiples modelos y se combinan sus predicciones para obtener un resultado final más robusto y preciso [41, 44, 60].

La idea fundamental detrás de los *ensembles* es que los diferentes modelos pueden tener fortalezas y debilidades diferentes, y al combinar sus predicciones, se puede obtener una mejor generalización y un mayor rendimiento en una variedad de situaciones. Para construir un *ensemble* se suele utilizar un conjunto de técnicas que se describirán a continuación.

El *bagging* consiste en generar múltiples conjuntos de datos de entrenamiento mediante muestreo con reemplazo, entrenando un modelo en cada conjunto y promediando o ponderando sus predicciones. En el *boosting*, los modelos se construyen secuencialmente, corrigiendo los errores del modelo anterior, y se combinan para formar un modelo más fuerte. La aumentación de datos consiste en ampliar el conjunto de entrenamiento para mejorar la capacidad de generalización del modelo y reducir el sobreajuste por medio de transformaciones sobre los datos como la rotación y ampliación, se podría usar como paso en el proceso de *bagging*. Los *random forests* combinan *bagging* y árboles

de decisión, generando múltiples árboles utilizando diferentes subconjuntos de datos y características. Las predicciones de los árboles individuales se combinan para obtener la predicción final. Por último, el *stacking* entrena múltiples modelos base y utiliza un meta-modelo para combinar sus predicciones. Cada estrategia tiene sus beneficios y consecuencias.

El presente TFG evalúa el uso de un meta-modelo para la estimación de calidad de las imágenes médicas 3D sin referencia.

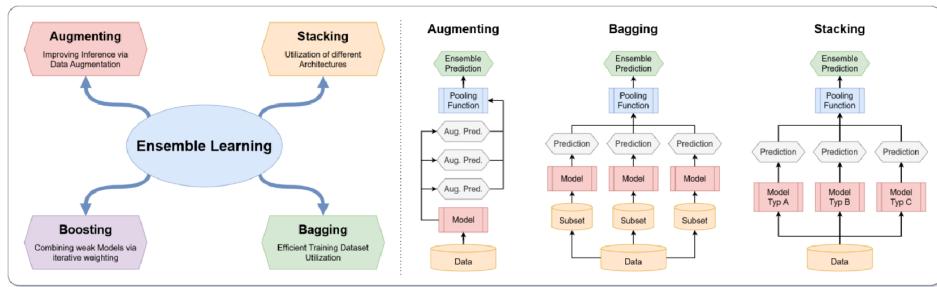


Figura 2.12: Representación de métodos de *ensemble* [60]. A la izquierda vemos una definición general de los distintos tipos de *ensemble*, a la derecha, el primer esquema es sobre aumentación de datos (un conjunto de datos y un modelo que se entrena con varias ejemplos alterados o no). El siguiente esquema representa *bagging* (el mismo modelo entrenado con distintos subconjuntos de datos). Por último, observamos el empleo de *stacking* (distintos modelos entrenados con el mismo conjunto de datos).

### 2.3. Imágenes médicas y distorsiones

Las tomografías computarizadas son un tipo de técnica de imagen médica que utiliza rayos X para obtener imágenes detalladas del interior del cuerpo. Durante una tomografía computarizada, el paciente se coloca en una mesa que se mueve a través de un anillo en forma de donut llamado *gantry*. Dentro del *gantry*, se encuentra un tubo de rayos X que gira alrededor del paciente, emitiendo haces de rayos X en forma de abanico. Los detectores ubicados en el lado opuesto del *gantry* registran la cantidad de rayos X que atraviesan el cuerpo del paciente. Estos datos se recopilan en múltiples ángulos y se utilizan para reconstruir imágenes transversales del cuerpo.

El número de imágenes en una tomografía computarizada se selecciona en función de varios factores, como el área del cuerpo que se está examinando, el propósito clínico de la exploración y las preferencias del radiólogo o médico que interpreta las imágenes. Ajustar el número de imágenes puede influir en el tiempo de adquisición, la cantidad de radiación utilizada y la cantidad de información detallada que se obtiene de la exploración. Afecta directamente

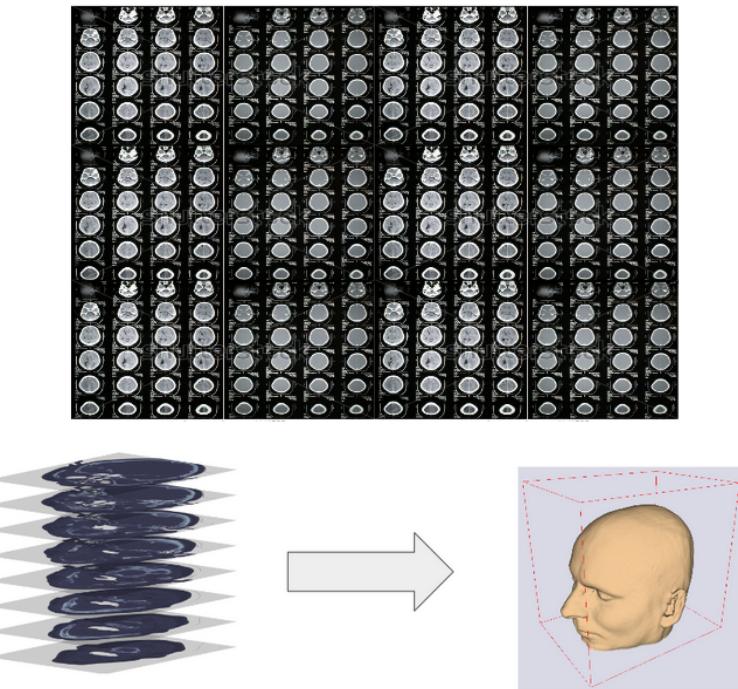


Figura 2.13: Ejemplo de tomografía computarizada y modelo 3D [61].

a la calidad del modelo 3D generado al final, ya que el número de cortes es la tercera dimensión que relaciona las imágenes (profundidad).

Sobre todo nos centraremos en las distorsiones geométricas que ocurren en la generación volumétrica de la imagen. La generación del volumen consiste en disponer de un conjunto segmentado<sup>3</sup> en todas las capas de las imágenes. A continuación usando el conjunto segmentado unificamos las coordenadas de los puntos por medio de intersecciones entre rayos proyectados sobre las imágenes (*ray casting*, véase Figura 2.13).

Dentro de las causas de las distorsiones geométricas sobre los volúmenes 3D están el difuminado por movimiento, errores de contraste (dificultad al segmentar), artefactos luminosos y problemas de interpolación o ruido al generar la proyección.

---

<sup>3</sup>La segmentación se refiere al proceso de dividir una imagen o conjunto de datos en regiones o componentes más pequeños. Ejemplo, en una foto del bosque, segmentamos los árboles para distinguirlos del suelo.



## Capítulo 3

# Estado del Arte

La estimación de calidad de imágenes y objetos 3D, al ser un componente sumamente ligado al avance tecnológico y necesidades de manejo de información digital, ha tomado mayor interés en el comienzo del siglo actual. Puede observarse en la Figura 3.1 que existe una tendencia creciente en el número de publicaciones en relación a la aplicación de IA en las nubes de puntos y en imágenes médicas, llegando ambas a sobrepasar 6000 documentos a partir de 2020. Vemos que ambos incluso siguen lado al lado en número de publicaciones cuando especificamos que sean documentos relacionados con la estimación de calidad, sobre pasando los 250 documentos. Por otro lado, y afirmando lo mencionado sobre el bajo número de publicaciones en el ámbito biomédico para la estimación de calidad en 3D, vemos que, aunque hay también una tendencia positiva, en 2022 tenemos solo 62 publicaciones. Esto se interpreta como indicador de lo novedoso y pionero de este proyecto.

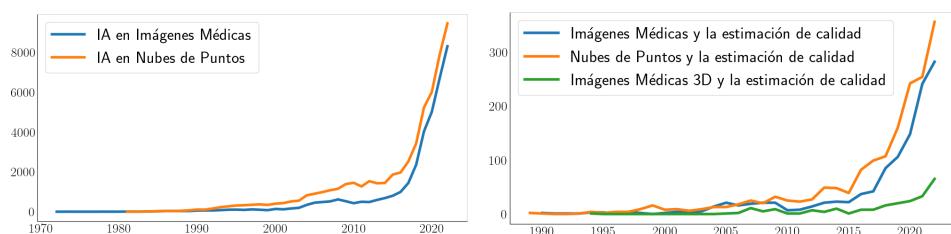


Figura 3.1: Crecimiento de interés en el campo según *Scopus*<sup>1</sup>. A la izquierda podemos ver un incremento de publicaciones desde 1970 sobre la IA aplicada a nubes de punto, en naranja, y aplicada de forma general a imágenes médicas, en azul. A la derecha podemos ver de forma similar el crecimiento de métodos aplicados a la estimación de calidad a nubes de puntos, naranja, a imágenes médicas, azul, y a imágenes médicas 3D, verde.

<sup>1</sup>Las búsquedas se pueden consultar en el Apéndice 7

### 3.1. Estado del arte de IQA

Para la resolución del problema, tanto FR como NR, ha sido crucial los avances de los conocimientos del sistema visual humano (HVS, por sus siglas en inglés). Se han propuesto métricas inspiradas en el HVS y conscientes del contenido de la imagen, que combinan las características del HVS con algoritmos matemáticos. En FR por ejemplo, Visual SNR [62] cuantifica la fidelidad visual de las imágenes con el ratio de relación entre señal-ruido y PSNR-HVS [63] combina el ese ratio con la función que determina nuestra sensibilidad al contraste. Wang y Bovik afirmaron que los ojos humanos obtienen información de imagen a través de tres canales: brillo, contraste y estructura [14] y desarrollaron un índice universal de calidad de imagen [64] y una similitud estructural (SSIM, por sus siglas en inglés) [14] para problemas FR. Surgieron incluso métodos que, basados en las respuestas del HVS, introducen la saliencia visual<sup>1</sup> en la evaluación de la calidad de imagen como VSI [15], ya que se ha observado que la saliencia de la imagen desempeña un papel importante.

Desde entonces se han propuesto diversas variantes de estos métodos, adaptadas al contexto e información disponible. Para la estimación de calidad de las imágenes sin referencia surgieron métricas para tipos de distorsiones específicas como imágenes borrosas [65], imágenes comprimidas en JPEG [66], imágenes con artefactos de bloque [67] e imágenes con cambios de contraste [68]. Luego, se buscaron maneras de estimar la calidad de imágenes de forma más genérica, sin depender del tipo de distorsión. Para ello, se han propuesto varias métricas basadas en NSS y el sistema HVS. Un ejemplo conocido es el evaluador sin referencia BRISQUE [69], que extrae características NSS de un modelo estadístico de coeficientes de luminancia normalizados localmente en el dominio espacial y demuestra que estas características se correlacionan bien con las evaluaciones humanas. También se han presentado métricas basadas en aprendizaje automático, como el índice basado en patrones locales de gradiente LGP [70] que extrae características estadísticas locales de la magnitud y fase del gradiente de la imagen y utiliza una SVM para mapear la calidad subjetiva de la imagen a características estadísticas locales que transmiten información estructural importante.

Recientemente, las redes neuronales convolucionales se han introducido con éxito en el campo de la evaluación de la fidelidad de imágenes sin referencia. Se propuso un trabajo pionero llamado IQA-CNN [71], y posteriormente se han realizado muchos esfuerzos para mejorar su rendimiento mediante el diseño de estructuras convolucionales más profundas. En concreto, DIQaM-NR [72], que mejora frente redes menos profundas.

En las Tablas 3.1 y 3.2 se recogen los métodos mencionados con las métricas

---

<sup>1</sup>Cualidad estética de la forma de un objeto o una configuración que destaca.

SROCC, PLCC y RMSE. Estas métricas se explicaran en detalle en la Sección 4.3.2. Las dos primeras son mejores cuanto más cercas al 1 y la última, RMSE, es mejor cuánto más pequeña sea. Se comparan sobre los conjuntos de datos públicos LIVE [14, 73, 74], CSIQ [75] y TID2008 [76]. El primero y el segundo poseen información de distorsiones por compresión JPEG, por difuminado gaussiano y ruido blanco. El último, posee más distorsiones, llegando hasta 17 diferentes.

<b>Métrica</b>	<b>LIVE</b>			<b>CSIQ</b>			<b>TID2008</b>		
	SRCC	PLCC	RMSE	SRCC	PLCC	RMSE	SRCC	PLCC	RMSE
VSNR [62]	0.927	0.923	10.506	0.811	0.800	0.158	0.705	0.682	0.982
PSNRHVS [63]	0.919	0.903	12.540	0.830	0.804	0.156	0.594	0.608	1.065
UQI [64]	0.894	0.899	11.982	0.810	0.831	0.146	0.585	0.664	1.003
SSIM [14]	0.948	0.845	8.946	0.876	0.861	0.133	0.775	0.773	0.851
MS-SSIM [77]	0.951	0.949	8.169	0.913	0.899	0.115	0.854	0.845	0.717
VSI [15]	0.952	0.948	8.682	0.942	0.928	0.098	0.898	0.876	0.647
DSS [13]	0.962	0.931	9.961	0.961	0.957	0.076	0.873	0.877	0.644
CD-MMF [12]	<b>0.981</b>	<b>0.980</b>	<b>5.413</b>	<b>0.967</b>	<b>0.9614</b>	<b>0.067</b>	<b>0.942</b>	<b>0.9414</b>	<b>0.429</b>
WaDIQaM [72]	0.970	<b>0.980</b>	-	-	-	-	-	-	-

Tabla 3.1: Tabla extraída de [78], donde vemos el progreso de las métricas FR conforme avanza los conocimientos del HVS, ML y DL.

<b>Métrica</b>	<b>LIVE</b>		
	SROCC	PLCC	RMSE
BRISQUE [69]	0.940	0.942	-
LGP [70]	0.957	0.954	7.901
IQA-CNN [71]	0.956	0.953	-
DIQaM-NR [72]	<b>0.960</b>	<b>0.972</b>	-
Hallucinated-IQA [33]	<b>0.982</b>	<b>0.982</b>	-

Tabla 3.2: Tabla extraída de [78], donde vemos el progreso de las métricas NR al utilizar métodos cada vez más complejos.

### 3.2. Estado del arte de PCQA

Los métodos de evaluación de fidelidad de imágenes 3D NR tienen más perspectivas de aplicación práctica que los métodos FR, ya que no utilizan ninguna información adicional del objeto de referencia. El enfoque común para este problema es el uso de métricas basadas en aprendizaje, donde se crea un modelo de predicción basado en propiedades de la nube de puntos que se creen relacionadas con la calidad de percepción, como por ejemplo la información del vecindario de los puntos, donde una gran cantidad de información geométrica puede ser extraída. Para el problema FR surgieron métodos basados en la extracción de esas características, donde la mayoría considera tanto información geométrica como los atributos lumínicos. Un ejemplo sería PointSSIM [79], una métrica que busca la similitud estructural entre nubes de puntos basándose en las estadísticas locales de curvatura de los puntos, junto a la información extraída de los colores, como adaptación del método SSIM [14], o PCQM [80] donde experimentaron con combinaciones de 3 medidas geométricas y 5 comparaciones de color para encontrar el mejor

vector características. Los métodos NR utilizan características similares, sin embargo no pueden hacer la diferencia entre las características de la imagen original y la distorsionada si no que tienen que inferir a partir de los valores de extraídos de la última.

Al igual que en las primeras aproximaciones de los métodos NR-IQA, existen algoritmos de estimación específicos, como el propuesto por Liu et al [81], método centrado en predecir la calidad de nubes de puntos codificadas mediante V-PCC, algoritmo de compresión, utilizando un modelo NR a nivel de *bitstream*. A continuación, los métodos siguen el camino explorado por los métodos FR-PCQA extrayendo características de las nubes de puntos para entrenar modelos. Chetouni et al. [82] utilizó distancias geométricas, nivel de curvatura media y niveles de color, en escala gris. Zhang et al. [36] siguiendo la misma línea extrajo características de los vectores y valores singulares de cada punto, además de utilizar características lumínicas. En el siguiente paso, al igual que en FR, se adaptaron ideas de otros ámbitos, por ejemplo, utilizando proyecciones que trasladan el mundo tridimensional a un espacio 2D para utilizar los métodos más conocidos de estimación de calidad en imágenes. En PQA-Net [83] se realiza un mapeado utilizando una estrategia de proyección multi-vista para extraer un vector de características de 384 dimensiones, que alimenta a dos módulos de aprendizaje que calculan conjuntamente la calidad de la nube de puntos degradada. En IT-PCQA [40] deciden utilizar métodos IQA aplicadas a las multi-proyecciones. Extendiendo los trabajos anteriores, tenemos VQA-PC [38] que trata las multi-proyecciones como vídeo, pudiendo así utilizar información espacial, imágenes en posiciones específicas, e información de consistencia temporal de la nube de puntos rotando. Se realizó incluso un estudio sobre el impacto del número de proyecciones 2D de distintas perspectivas en el rendimiento de las métricas de calidad [40, 84].

Los últimos pasos deciden utilizar información de la nube de puntos, para entender remediar cierta pérdida de información que puede ocurrir al proyectar-la. En ResSCNN [85] modifican el esqueleto de PointNet [58] para utilizar convoluciones dispersas, extraer características de forma jerárquica y predecir la calidad de la nube. También, con intención de ayudar al desarrollo de métodos NR-PCQA de aprendizaje profundo, construyeron el mayor conjunto de datos sintéticos de nubes de puntos distorsionadas, en el momento de escritura, con sus correspondientes estimaciones de calidad. Otro método que trabaja directamente sobre la nube de puntos es SGR [35], que extrae regiones locales de la nube de puntos y analiza la calidad de los parches. Recientemente, ha salido el modelo MM-PCQA [39] que utiliza tanto información directa de la nube de puntos como información de las proyecciones. Incluso, hay métodos que han optado por utilizar un *ensemble* [86] de modelos pre-entrenado en 2D y obtuvieron muy buenos resultados.

MODELO	STJU-PCQA		WPC	
	PLCC	SRCC	PLCC	SRCC
IT-PCQA [40]	0.58	0.63	0.55	0.54
Zhang et al. [36]	0.7382	0.7144	0.6514	0.6479
GPA-Net [37]	0.806	0.78	-	-
ResSCNN [85]	0.86	0.81	0.72	0.75
VQA-PC [38]	0.8635	0.8509	0.7976	0.7968
SGR [35]	0.89	0.84	-	-
MM-PCQA [39]	<b>0.92</b>	<b>0.91</b>	<b>0.83</b>	<b>0.83</b>

Tabla 3.3: Resumen del estado del arte de modelos NR-PCQA en dos datasets muy conocidos: SJTU [11] y WPC [87, 88].

### 3.3. Estado del arte de IQA en imágenes médicas

En el caso de la evaluación de calidad FR y RR, se requiere disponer de una imagen de referencia sin distorsión o una parte de una imagen con la cual se pueda comparar la imagen evaluada. Sin embargo, en el caso de las imágenes médicas, no existe una imagen sin distorsión [89]. Por lo tanto, el desarrollo de métodos de evaluación de calidad de imágenes sin referencia es de particular importancia en este campo [33, 69-72]. Como fue mencionado, la salida de estos algoritmos pueden ser utilizados para filtrar imágenes de baja calidad en un gran conjunto de imágenes médicas o para ayudar a mejorar su calidad, siendo esta crucial para el diagnóstico [89]. Predominantemente, el conocimiento del ámbito de la imagen es crítico para estimar su calidad. Es por ello que la mayoría de los métodos actuales son para un tipo específico de examen médico o distorsión.

Por ejemplo, basándose en el sistema visual humano, Bhateja et al. [90] utilizaron métricas de fusión de imágenes de resonancia magnética (MRI) de dos etapas para IQA. Con el objetivo de desarrollar métodos automáticos de aprendizaje profundo, Xu et al. [91] introdujeron una técnica semi-supervisada dedicada a la evaluación de calidad de imágenes de MRI cerebral fetal utilizando un método de *Mean Teacher* [92] y la consistencia de las regiones de interés. Además, Liu et al. [93] utilizaron el aprendizaje semi-supervisado para resolver el problema de crear anotaciones ruidosas en la tarea de segmentación de imágenes. Esta técnica de evaluación de calidad de tres etapas utiliza un modelo residual jerárquico y proporciona una evaluación a nivel de corte, volumen y sujeto.

Otro método de estimación utiliza una red generativa adversaria no emparejada y un clasificador entrenado débilmente supervisado para evaluar imágenes MRI [94]. Para abordar el problema de desperdiciar información espacial 3D potencialmente importante, se creó el enfoque HyS-net [95], basado en una hiper-red que es capaz de auto adaptación. Así como fue expuesto anteriormente, no es posible simplemente implementar métodos IQA, es por ello que Chow y Rajagopal [96] propusieron un enfoque más reciente que adapta el evaluador de calidad de imagen más famoso BRISQUE [69].

Estos métodos consisten de aproximaciones 2D, específicas a un tipo de examen, que resuelven el problema de la estimación de calidad de imágenes médicas por medio de los múltiples planos anatómicos bidimensionales. Sin embargo, no se ha encontrado nada específico en la literatura para las reconstrucciones 3D generadas a partir de dichas imágenes médicas, por lo que es un campo sin explorar. No obstante, cada vez más frecuentemente se emplean volúmenes tridimensionales en la medicina y, consecuentemente, es necesario explorar ese campo a pesar de la complejidad existente.

## Capítulo 4

# Materiales y Métodos

### 4.1. Materiales

#### 4.1.1. Conjunto de datos genéricos

Para este TFG se han utilizado diversos conjuntos de datos públicos para la evaluación y elección de los modelos anteriormente descritos. De entre ellos están SJTU [11], WPC [87, 88] y LS-PCQA [85] que tratan de conjuntos de nubes de puntos generalistas, de personas, animales y objetos.

El primero de ellos parte de 10 nubes de puntos de referencia (véase Figura 4.1), a las cuales se aplican 7 tipos de distorsiones. Estas son: compresión, ruido al color, ruido geométrico, ruido gaussiano y combinación entre ellas (véase Tabla 4.1). Todas se aplican en una escala creciente de intensidad del 1 al 6. Luego, se obtiene un MOS de 10 individuos para las 420 nubes de puntos que sirve como medida de calidad de las mismas y para evaluar las predicciones del modelo.

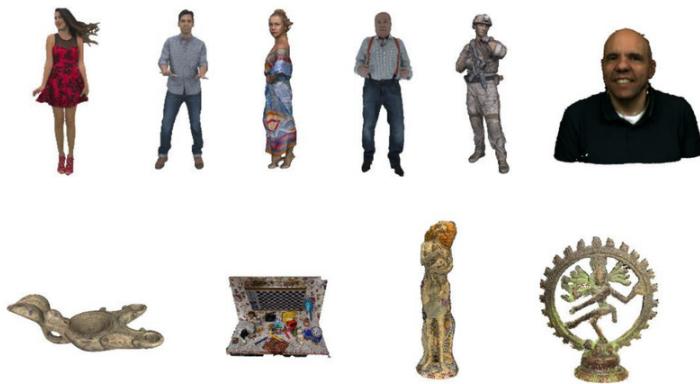


Figura 4.1: Ejemplo de conjuntos de datos SJTU.

Número	Tipo de Distorsión
0	OT: Compresión octree [97]
1	CN: Ruido fotométrico
2	DS: Submuestreo uniforme
3	DS + CN
4	DS + GGN
5	GGN: Ruido geométrico gaussiano
6	CN + GGN

Tabla 4.1: Ejemplo de distorsiones en SJTU.

El segundo dataset, WPC [87, 88], también posee distorsiones como submuestreo uniforme y ruido gaussiano (aplicados de manera distinta), pero a su vez posee nuevos tipos de distorsiones. Estos son basados en distintos tipos de compresión: V-PCC, G-PCC y *trisoup*. Además, posee distintos tipos de nubes de puntos (véase Figura 4.2) que pueden influir en el rendimiento del modelo si el conjunto no es suficientemente amplio y representativo de lo que puede encontrarse una vez entrenado.

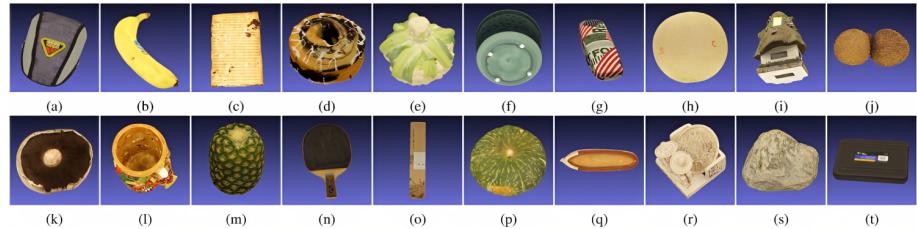


Figura 4.2: Ejemplo de conjunto de datos WPC.

Los dos anteriores han sido utilizados sobre todo para la evaluación y elección del modelo de regresión a utilizar. Son los conjuntos de datos más conocidos y que habitualmente están presentes en las publicaciones más recientes. Además, se realizaron pruebas de ejecuciones de algunos métodos de código abierto para verificar los resultados. Sin embargo, es el último el que finalmente se utiliza para entrenar un modelo para estimar la calidad de las imágenes médicas. Esto es porque LS-PCQA [85] es el mayor conjunto de datos, en el momento de escritura, y posee tipos de distorsiones que pueden simular lo que sería ciertos errores y ruidos presentes en imágenes médicas. Por ejemplo, el ruido gaussiano (simular errores de transmisión y almacenado de datos), rotación y movimiento local (simular el movimiento del paciente) y compresión octree y por submuestreo uniforme (algoritmos de compresión comúnmente usados). Aparte, es el con mayor amplitud de modelos base, con distintos tipos y categorías de objetos (véase Figura 4.3).

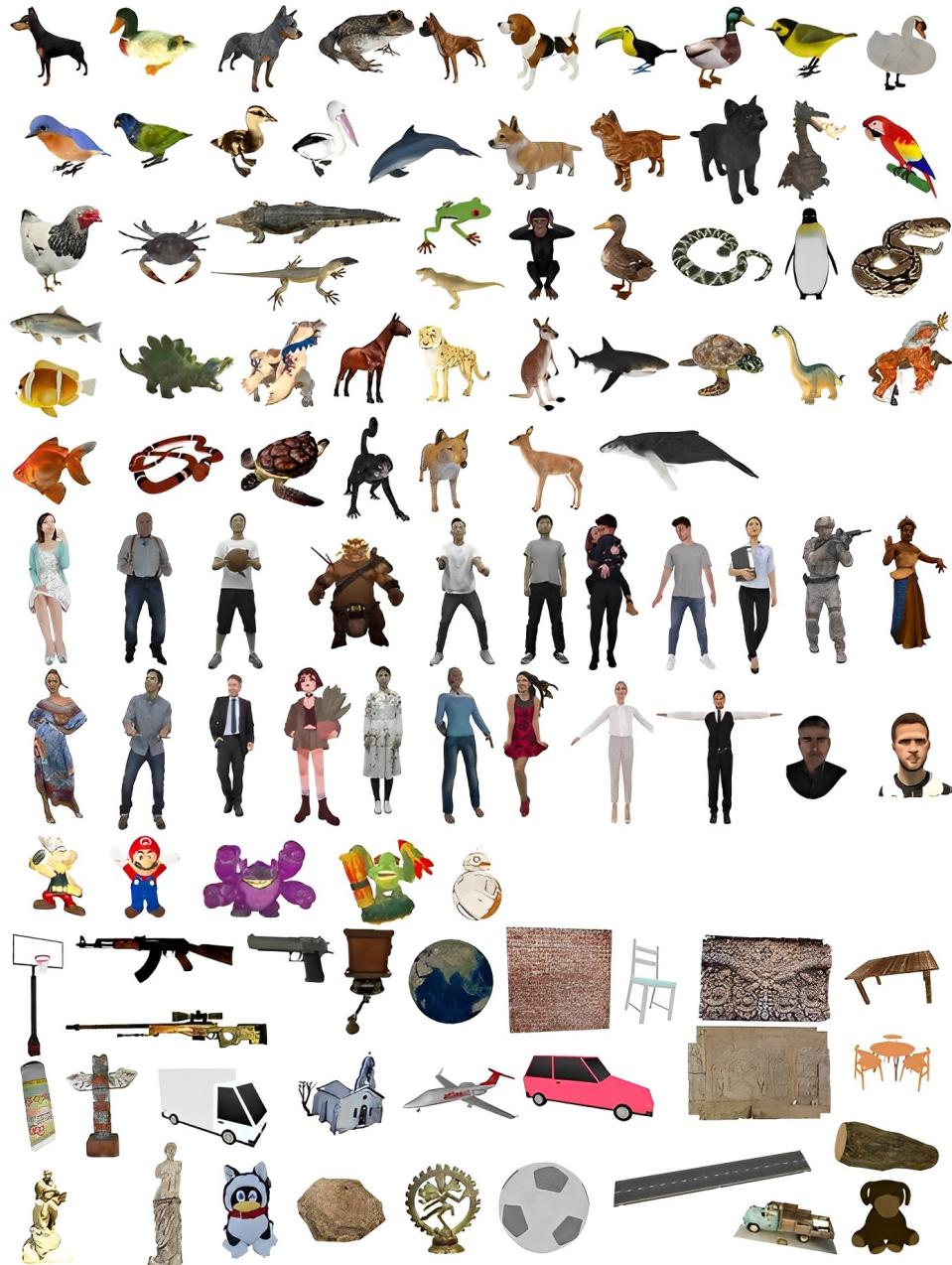


Figura 4.3: Ejemplo de conjunto de datos LS-PCQA. Vemos que en este conjunto de datos tenemos una gran variedad de nubes de puntos. En las primeras filas tenemos un conjunto de modelos de animales, seguidos de representaciones de seres humanos y, por último, varios objetos abstractos.

### 4.1.2. Conjunto de datos médicos

Para este TFG se tiene disponible una conjunto de tomografías computarizadas, de diferentes partes del cuerpo, de 2 individuos distintos del conjunto de datos públicos NMID [98]. De los cuales han sido segmentados las clavículas, el seno frontal y los senos maxilares. A parte, disponemos de modelos generados mediante escáner láser 3D, en concreto con el dispositivo Artec Spider, que permite generar reconstrucciones 3D de objetos reales. Estos modelos 3D fueron adquiridos en el Departamento de Medicina Legal, Toxicología y Antropología Física de la Universidad de Granada y constan del cráneo de otros 3 individuos, una pubis izquierda y una pubis derecha (véase Figura 4.4). Es decir, en total disponemos de 11 nubes de puntos de alta calidad, que representan distintos volúmenes de exámenes médicos. A estos datos no se hicieron ningún tipo de pre-procesado, apenas se centraron las nubes de puntos a los ejes (operación necesaria para hacer la rotación para las distintas perspectivas, más detalles en el Apéndice 7) y se eliminaron aquellos puntos aislados de todos, frutos de errores en el algoritmo de reconstrucción 3D de las nubes de puntos a partir de segmentaciones DICOM.

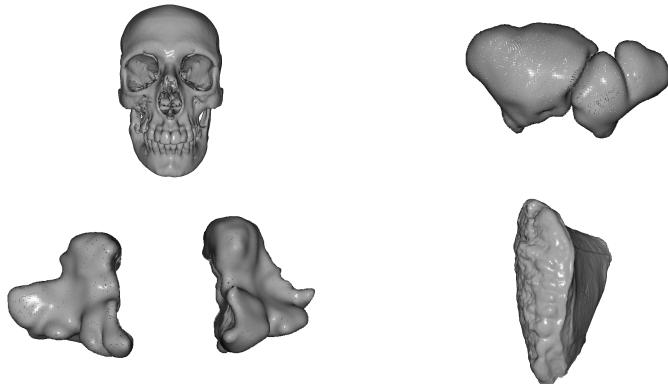


Figura 4.4: Ejemplo de nuestras imágenes médicas. Arriba a la izquierda tenemos un cráneo y a su derecha un seno frontal. Abajo a la izquierda tenemos un maxilar y a su derecha el pubis derecho.

Cada nube de punto es centrada sobre los ejes como paso previo. A continuación, reducimos el número de puntos anormales analizando estadísticamente el vecindario de cada punto. A cada uno de estos ejemplos disponibles, se les aplica las 5 distorsiones médicas discutidas en la Sección 2.3. Para simular dichas distorsiones, partiremos de todos los ejemplos de imágenes médicas mencionados anteriormente, considerados exentos de desperfectos. Dichos ejemplos están segmentados por profesionales.

Dos de las distorsiones serán para simular los resultados de varios algoritmos de compresión, como puede ser *octree compression* [97] y la reducción de

número de puntos por medio de submuestreo aleatorio (en inglés *random downsampling*). Un *octree* es una representación más eficiente que los véxeles, se trata de la descomposición de forma recursiva de la escena en 8 partes hasta la profundidad máxima, donde cada nodo representa un cubo tridimensional llamado octante. En la compresión se analiza los octantes y se elimina los que aportan menos información. En el segundo, establecemos un porcentaje de puntos que eliminar y los eliminamos de forma aleatoria hasta alcanzar ese porcentaje de reducción. Las demás representarán efectos que podrían ocurrir por desplazamiento de los puntos, como el movimiento del paciente y aquellos provocados por ruido en la transmisión de datos.

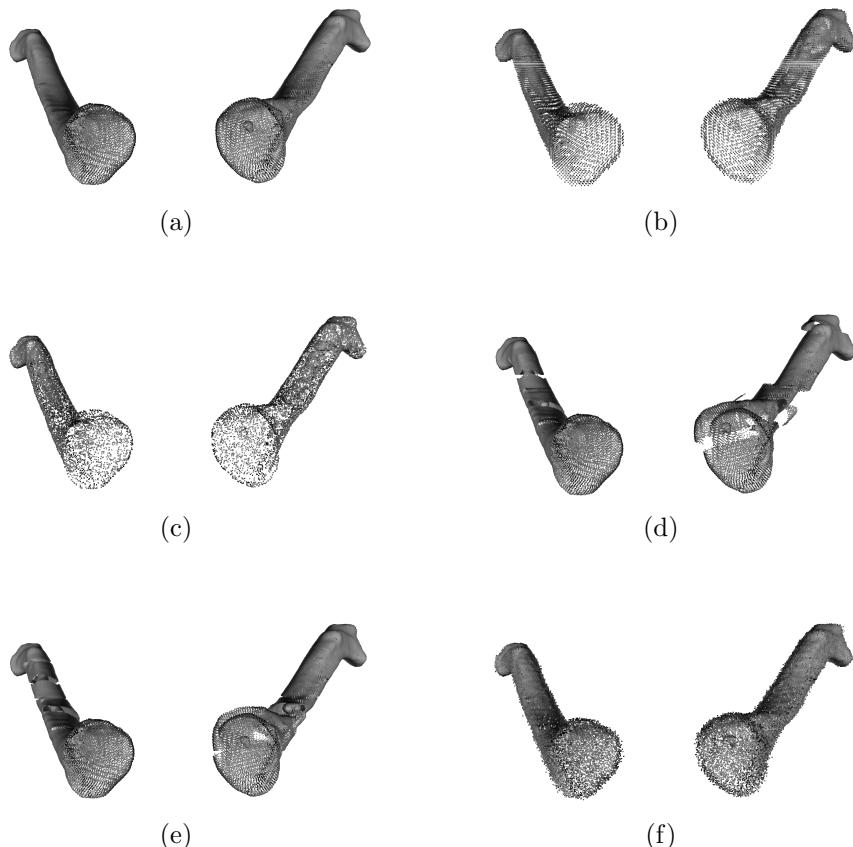


Figura 4.5: Ejemplo de distorsiones generadas sobre clavículas. Tenemos la imagen original (a) y luego su versión tras compresión *octree* (b), tras reducción de puntos por *random downsampling* (c), tras simulación de movimiento local (d) y rotación local (e), y, por último, ruido gaussiano (f).

En total tendremos: compresión *octree*, submuestreo uniforme, movimiento local, rotación local y ruido gaussiano. Cada uno de estas distorsiones son

aplicadas en 7 niveles crecientes de intensidad. Las resoluciones de compresión *octree* van de 0.4 a 1.0 en intervalos de 0.1. El submuestreo aleatorio realiza una reducción del 10 % al 70 % de los puntos. Tanto el movimiento como la rotación local se aplican de 1 a 7 veces. Por último, el ruido gaussiano va de 0.15 %, 0.20 %, 0.25 %, 0.30 %, 0.35 %, 0.4 % y 0.5 % del *bounding box*. Como resultado, tenemos un total de 385 ejemplos (véase Apéndice 7).

## 4.2. Métodos

Como se pudo observar en la Sección 3, actualmente hay una tendencia, justificada, a los métodos de DL frente a ML. Sin embargo, se experimentará con ambos. No obstante, se tuvo que descartar o adaptar todos los métodos que tuvieran en cuenta información de textura, cosa que no existe en los volúmenes médicos habituales. También se descartaron modelos que utilizan información perceptual de regiones locales, ya que necesitamos dicha información respecto de la imagen en su totalidad. Ambas características son dificultades añadidas a la hora de resolver el problema. La primera restringe el problema a la estimación de calidad de las estructuras en la imagen, eliminando la percepción de calidad por contraste y saturación. La segunda incrementa la complejidad computacional.

### 4.2.1. Modelo NR 3D-QA

Antes de probar directamente con modelos de DL, se experimentó con un método de ML basado en la extracción de características de escena y entrenamiento de un modelo de vectores soporte para la regresión. Para ello necesitamos definir qué tipo de características queremos extraer.

Zhang et al [36] proponen utilizar características geométricas y de color. Para la primera, extraen la curvatura (4.1), anisotropía (4.2), linealidad (4.3), planaridad (4.4) y esfericidad (4.5) de los puntos. Estas características se pueden extraer del vecindario de un punto por medio de la matriz de covarianza y los valores singulares. Las fórmulas que las definen son:

$$Cur(p_i) = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \quad (4.1)$$

$$Ani(p_i) = \frac{\lambda_1 - \lambda_3}{\lambda_1} \quad (4.2)$$

$$Lin(p_i) = \frac{\lambda_1 - \lambda_2}{\lambda_1} \quad (4.3)$$

$$Pla(p_i) = \frac{\lambda_2 - \lambda_3}{\lambda_1} \quad (4.4)$$

$$Sph(p_i) = \frac{\lambda_3}{\lambda_1} \quad (4.5)$$

Donde  $\lambda_1$ ,  $\lambda_2$  y  $\lambda_3$  se refieren a los correspondientes valores singulares. Para la extracción de las características de color, primeramente convierten el espacio de color RGB en el espacio LAB mediante los siguientes pasos de transformación:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2.7688 & 1.7517 & 1.1301 \\ 1.0000 & 4.5906 & 0.0601 \\ 0 & 0.0565 & 5.5942 \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4.6)$$

$$\begin{cases} L = 116f\left(\frac{Y}{Y_n}\right) - 16 \\ A = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \\ B = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \end{cases} \quad (4.7)$$

Donde R, G y B son los correspondientes canales RGB de color. La función que determina la transformación final viene descrita por (4.8):

$$f(t) = \begin{cases} \sqrt[3]{t} & \text{sí } t > \sigma^3 \\ \frac{t}{3\sigma^2} + \frac{4}{29}, & \text{en cualquier otro caso.} \end{cases} \quad (4.8)$$

Donde  $\sigma = \frac{6}{29}$ . Sin embargo, en el caso de las imágenes médicas estas características deben ser descartadas, dado que el color existente al visualizar las nubes de puntos médicas no son más que un valor sintético añadido previamente que permite una visualización más agradable de las mismas. Finalmente, estiman la entropía de cada una de las características ya que argumentan que existe una alta correlación entre la entropía y la distorsión por cuantización. Además, a las características geométricas se les calcula la distancia a las distribuciones gaussiana y gamma tras observarse que la distribución de estas se veía afectada por la intensidad de las distorsiones.

Para cada tipo de característica, se utiliza el valor medio y la desviación típica obtenida para cada punto de la nube de punto. Por último, sobre estas medidas, para el conjunto de datos de entrenamiento se normaliza con (4.9). Donde F es la característica extraída y C una pequeña constante para la estabilidad numérica. El conjunto de test se normaliza utilizando las medias y desviaciones de los datos de entrenamiento.

$$\hat{F} = \frac{F - \text{mean}(F)}{\text{std}(F) + C} \quad (4.9)$$

#### 4.2.2. Modelo VQA-PC

Zhang et al [38] propusieron un modelo de estimación de calidad de nubes de puntos utilizando proyecciones 2D de diferentes perspectivas. Observaron

que los métodos que trabajan directamente sobre la nube de puntos tienen una elevada dificultad computacional, sin suponer una mejora excesiva, y que deben todavía madurar en el campo dado la alta complejidad de las nubes de puntos. Por ello proponen utilizar proyección multi-vista. No obstante, argumentaron que los métodos anteriores de proyección se basan en la hipótesis de que los humanos percibimos la calidad de modelos 3D desde una perspectiva estática, cosa que no es cierta en la práctica dado que los objetos 3D permiten operaciones geométricas de rotación y escalado. Y por ello, proponen unificar la percepción estática con la dinámica tratando a las proyecciones como vídeos.

De esta forma, se puede extraer características espaciales y temporales, como discutido en la Sección 2.2.2, utilizando redes convolucionales adaptadas a vídeos, de la familia *SlowFast* [57]. Siguiendo la motivación de que las deformaciones geométricas no deseadas se presentan de forma abrupta según la perspectiva (véase Figura 4.6), y que incluso se pueden observar incoherencias entre perspectivas adyacentes utilizaron 4 ejes de rotación: vertical, horizontal, diagonal derecha y diagonal izquierda. Para cada eje se genera un total de 30 *frames*, en total habrá 120 (véase Figura 4.7). El ángulo de rotación es de 12 grados para todos los casos. Terminando la rotación de un eje en la misma posición inicial. A continuación se extraen características temporales del vídeo, que es posible generar a partir de cada *frame* de los distintos ejes de rotación encadenados secuencialmente de forma ordenada, y se elige 1 *frame* de cada eje de rotación para representar la información espacial. Por último, tenemos que aprender una función de interacción entre los dos vectores característicos extraídos. Proponen concatenar los vectores y aprender una función por medio de una capa totalmente conectada utilizando el MSE.

Para realizar la secuencia de vídeo, necesitamos realizar correctamente el conjunto de rotaciones descritos por las siguientes ecuaciones:

$$\theta_A = \begin{cases} X_\alpha^2 + Y_\alpha^2 = R^2 \\ Z_\alpha = 0 \end{cases} \quad (4.10)$$

$$\theta_B = \begin{cases} Y_\alpha^2 + Z_\alpha^2 = R^2 \\ X_\alpha = 0 \end{cases} \quad (4.11)$$

$$\theta_C = \begin{cases} X_\alpha^2 + Y_\alpha^2 + Z_\alpha^2 = R^2 \\ X_\alpha + Z_\alpha = 0 \end{cases} \quad (4.12)$$

$$\theta_D = \begin{cases} X_\alpha^2 + Y_\alpha^2 + Z_\alpha^2 = R^2 \\ X_\alpha - Z_\alpha = 0 \end{cases} \quad (4.13)$$

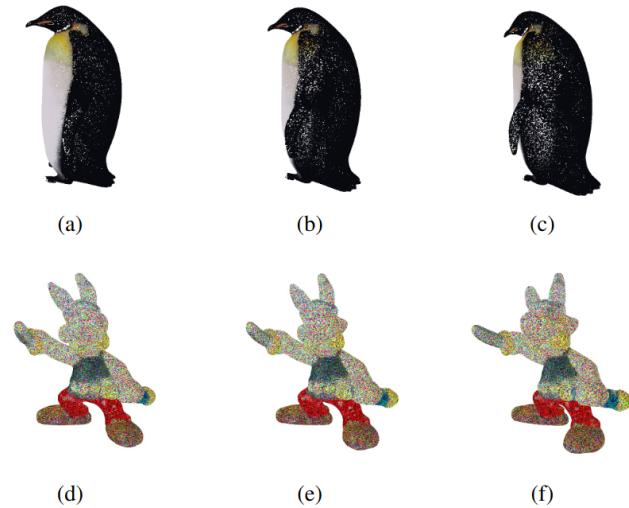


Figura 4.6: Ejemplo de distorsiones que se presentan según la perspectiva. Vemos que al girar el pingüino se empieza a observar un bajo número de puntos en su lateral izquierdo, permitiendo verse a través de él. De forma similar, en la imagen de abajo se ve cierta deformación de la cabeza.

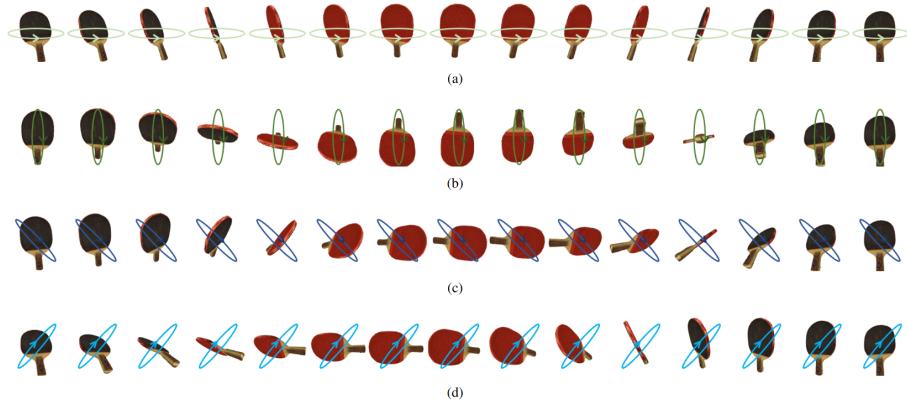


Figura 4.7: Ejemplo de las rotaciones que utiliza el modelo VQA-PC. Se observa que el final de cualquier eje de rotación es la posición inicial, permitiendo así unir suavemente una secuencia de imágenes encadenada de los ejes que genere un vídeo de rotación utilizado luego para la estimación.

Y para llevar a cabo la rotación debemos calcular el punto medio de la nube de puntos por medio de la siguiente Ecuación (4.14):

$$O_\sigma = \frac{1}{N} \sum_{n=1}^N \sigma_n \quad (4.14)$$

Donde el  $O_\sigma$  representa la coordenada (X,Y,Z) del centro medio de la nube de punto, y  $\sigma_n$  representa la coordenada del punto  $n$ -ésimo punto. Utilizando ese centro, aplicamos las ecuaciones (4.10) a (4.13).

Para extraer las características espaciales empleamos un modelo pre-entrenado, en concreto se investigaron variaciones de arquitecturas ResNet [99]. Una familia de redes residuales, que en su momento resolvieron el problema del estancamiento en el entrenamiento de redes neuronales profundas debido a la degradación del gradiente. El único modelo al que optimizaremos sus pesos es ResNet, el modelo de extracción temporal solo es un paso previo.

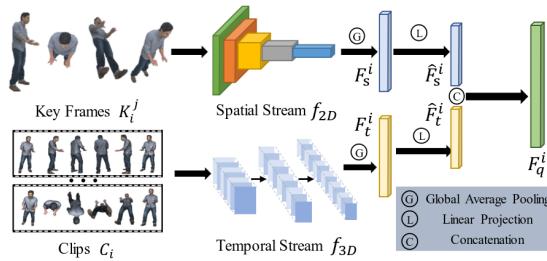


Figura 4.8: Ejemplo detallado de las etapas del método de VQA-PC.

## 4.3. Evaluación

### 4.3.1. Etiquetado

Al tener que generar un conjunto de datos médicos por medio de la simulación de distorsiones a distintos niveles de intensidad, es también necesario una manera de etiquetar cada ejemplo generado con el valor de calidad esperado. Para la generación de dicha etiqueta, se optó seguir el camino propuesto por [85]. En opción a generar un entorno controlado con los estándares ITU-R [100, 101], organizar al menos 16 personas, y que cada uno evalúe durante 30 segundos el nivel de calidad de cada nube de punto en una escala 1-5, hemos hecho uso del gran avance de las métricas con referencia que poseen un alto nivel de correlación con la percepción de calidad del observador final. Para ello, se hace un desglose de rendimiento de cada métrica para cada tipo de distorsión, como se observa en la Tabla 4.2. El rendimiento es medido con

el coeficiente de correlación de rangos de Spearman. Las métricas comparadas son p2p0 [102] y p2pl [103], con distancia manhattan (M) y hausdorff (H), PCQM [80], GraphSIM [104] y MPED [105].

Distortion	M-p2po	M-p2pl	H-p2po	H-p2pl	PCQM	GraphSIM	MPED
DownSample	<b>0.881</b>	0.626	0.841	0.811	0.524	0.842	0.857
GaussianShifting	0.741	0.718	0.829	<b>0.834</b>	0.816	0.742	0.598
LocalOffset	<b>0.937</b>	0.934	0.770	0.770	0.851	0.906	0.897
LocalRotation	0.819	0.712	<b>0.831</b>	0.734	0.657	0.723	0.742
Octree	0.779	0.788	<b>0.819</b>	0.752	0.676	0.757	0.710

Tabla 4.2: Tabla de métricas para generación de etiquetas extraídas de [85].

En [85] validaron sus muestras generadas con las métricas del estado del arte para el subproblema FR por medio de un análisis subjetivo con el estándar ITU-R, en un entorno controlado, definido anteriormente, y obtuvieron una correlación del 90 % entre las muestras etiquetadas subjetivamente y las que fueron etiquetadas objetivamente (véase Tabla 4.3). Con ello, se logra justificar la generación de las etiquetas como una sustitución de los métodos de evaluación subjetivos.

	Parte I	Parte II
SROCC	0.902697	0.878517
PLCC	0.910713	0.871917

Tabla 4.3: Correlación de métricas sintéticas extraídas de [85]. Donde “Parte I” y “Parte II” se refieren a dos conjuntos de datos etiquetados manualmente. El primero se utiliza para la elección de las métricas y el segundo para test.

#### 4.3.2. Métricas de similitud

Las métricas más utilizadas en la resolución del problema de estimación de calidad de imágenes suelen ser: el coeficiente de correlación de rangos de Spearman (SROCC, por sus siglas en inglés), el coeficiente de correlación lineal de Pearson (PLCC), coeficiente de correlación de orden de rango de Kendall (KROCC) y raíz del error cuadrático medio (RMSE) [2].

Las tres primeras al ser coeficientes de correlación toman valores en el intervalo [-1, 1]. Siendo el valor -1 una correlación negativa entre los datos, es decir, ambos decrecen en el tiempo. Al contrario, cuando esta es +1, tenemos una relación positiva que implica un crecimiento en el tiempo. Sin embargo, cada una de ellas miden la correlación de forma distinta.

PLCC es una métrica que mide la correlación lineal entre dos conjuntos de datos. Evalúa si existe una relación lineal entre los valores de ambos conjuntos. Si definimos  $x$  e  $y$  como los vectores que contienen las puntuaciones de calidad objetiva y subjetiva de  $m$  imágenes, siendo  $x_i$  e  $y_i$  los elementos contenidos en la posición  $i$ , entonces podemos formular PLCC como la Ecuación (4.15).

$$PLCC(x, y) = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^m (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^m (y_i - \bar{y})^2}} \quad (4.15)$$

Como mencionado en [85] o [38], se sugiere una transformación no lineal para las puntuaciones objetivas antes de calcular el PLCC y el RMSE. Para ello utilizaremos la función de regresión logística-5, con 5 parámetros a aprender, como en la Ecuación (4.16).

$$Q = \beta_1 \left( \frac{1}{2} - \frac{1}{1 + e^{\beta_2(Q_s - \beta_3)}} \right) + \beta_4 Q_s + \beta_5 \quad (4.16)$$

Donde  $Q$  es el valor final normalizado,  $Q_s$  es el valor predicho y  $\beta_i$  se refiere a los parámetros a aprender. La elección se debe al análisis comparativo entre esta, logística-4, con 4 parámetros, y la función de regresión cúbica-4 desarrollado en [85]. Los resultados se pueden ver en la Tabla 4.4, y las fórmulas adicionales en el Apéndice 7.

	Logística-4	Logística-5	Cúbica-4
SROCC	0.8572	<b>0.9026</b>	0.8957
PLCC	0.8626	<b>0.9107</b>	0.9044

Tabla 4.4: Comparación de la correlación entre dos conjuntos de datos, la etiqueta y la predicción, tras utilizar las diferentes técnicas de normalización no lineal. Vemos que hay una mayor correlación entre los datos si se normalizan con la función logística-5.

También podemos no depender de la escala de los datos, para ello tendríamos que utilizar SROCC. Esta es una métrica que mide la correlación de clasificaciones o *rankings* entre dos conjuntos de datos. Evalúa si el orden relativo de los elementos es similar en ambos conjuntos. Por ello, es también invariante a transformaciones monótonas en los datos. Se puede formular como la Ecuación 4.17.

$$SROCC(x, y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}} \quad (4.17)$$

De hecho, la correlación de rangos de Spearman es equivalente a calcular la correlación de Pearson sobre los rangos de los valores de entrada.

$$SROCC(x, y) = PLCC(rank(x), rank(y)) \quad (4.18)$$

KROCC es una métrica similar a SROCC, pero utiliza el coeficiente de correlación de rangos de Kendall. También evalúa la correlación entre clasificaciones o *rankings*, pero se basa en la concordancia o discordancia de los pares de elementos en los conjuntos.

$$KROCC(x, y) = \frac{C - D}{\frac{1}{2}m(m - 1)} \quad (4.19)$$

Donde  $C$  alude a cuantos pares de datos,  $x$  e  $y$ , están bien correlacionadas, y  $D$  es el número de pares discordantes.

Por último, RMSE es una métrica que mide la diferencia entre los valores predichos y los valores reales en un conjunto de datos. Calcula la raíz cuadrada del promedio de los errores al cuadrado. Un valor de RMSE más bajo indica un mejor ajuste o precisión del modelo.

$$RMSE(x, y) = \sqrt{\frac{1}{m} \sum_{i=1}^m (x_i - y_i)^2} \quad (4.20)$$



# Capítulo 5

## Implementación y Experimentos

### 5.1. Diseño Experimental

Todo el código desarrollado se encuentra en el GitHub [https://github.com/CodeBoy-source/TFG\\_NRPCQA](https://github.com/CodeBoy-source/TFG_NRPCQA). Más detalles sobre la simulación de las distorsiones y el entorno de ejecución se recogen en el Apéndice 7. En la Figura 5.1 se observa el diagrama de secuencias que determina el flujo de tareas implementadas para el desarrollo de este proyecto. Como vimos en la sección anterior, el primer paso se divide en la simulación de distorsiones y en el etiquetado de un valor de calidad objetivo. A continuación, en esta sección se expone el proceso de obtención de los resultados de experimentación y el correspondiente análisis de cada uno de ellos.

#### 5.1.1. Protocolo de validación experimental

Para la validación del modelo y la estimación de su rendimiento para las distintas mejoras propuestas se ha utilizado la técnica de *cross-validation* ó validación cruzada, también conocida por *K-fold*. Esta técnica se distingue por realizar la división del conjunto de datos en K partes (pliegues). A continuación el modelo se entrena y evalúa K veces, utilizando cada una de los pliegues como conjunto de prueba y el resto de los pliegues como conjunto de entrenamiento en cada iteración (véase Figura 5.2). Al finalizar las K iteraciones, se promedia los resultados de evaluación obtenidos para obtener una medida general de rendimiento. Es decir, un K-fold con K=1 equivale a la técnica de *hold-out* donde se divide el conjunto de datos en el conjunto de entrenamiento y el conjunto de test.

En el caso de pre-entrenar usando LS-PCQA [85], con 3640 ejemplos en total

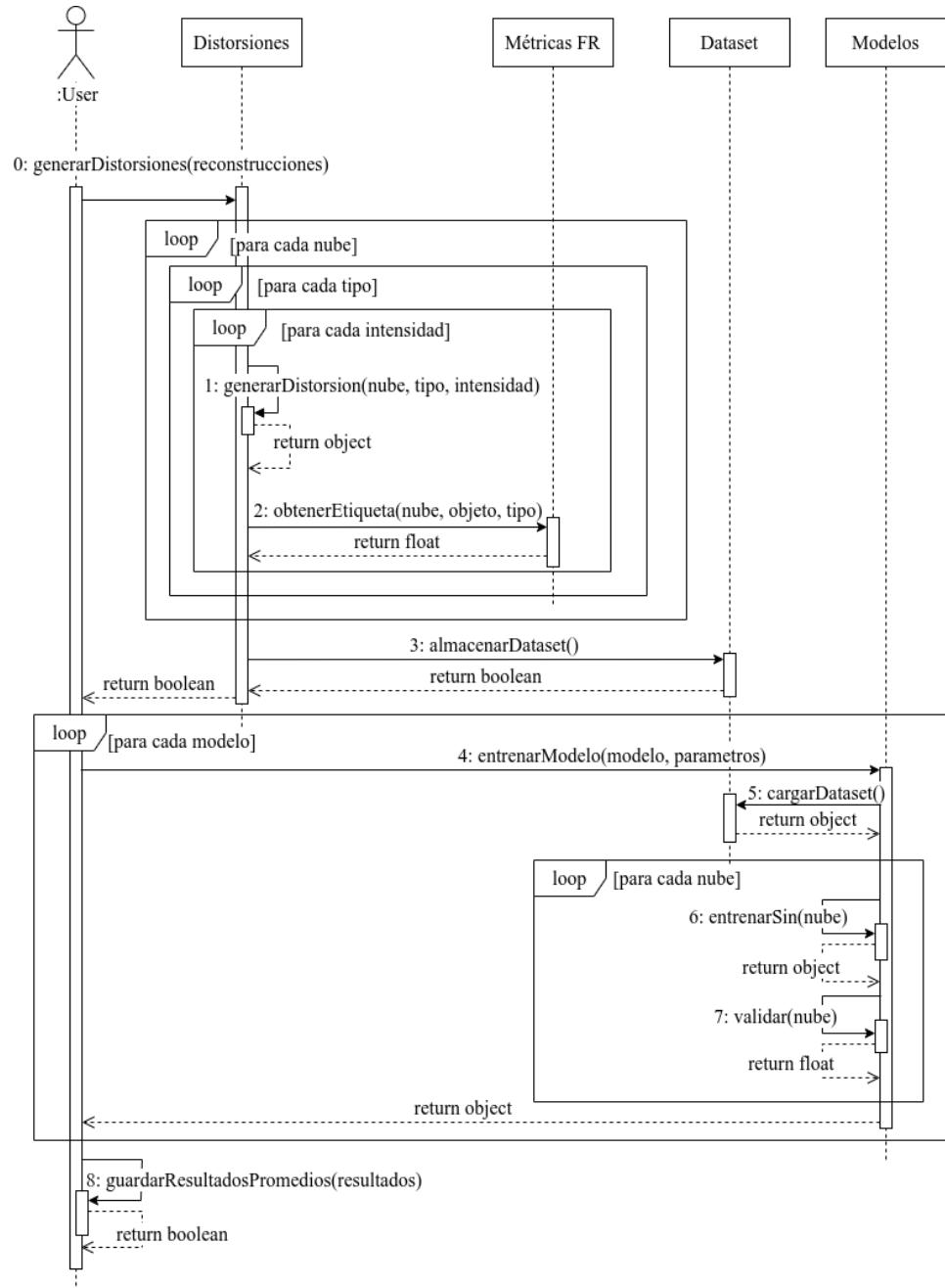


Figura 5.1: Diagrama de secuencias del proyecto. Se observan dos grandes bloques: generación de datos sintéticos (0 a 3) y experimentación (4 a 8).

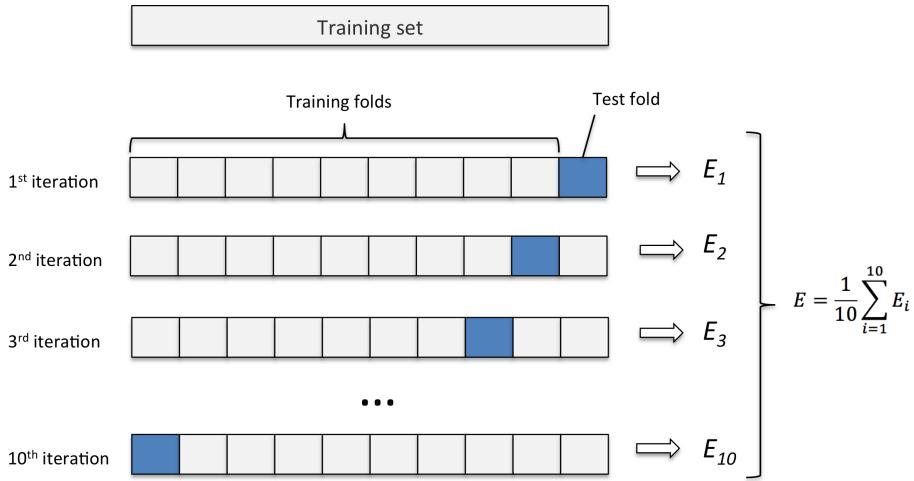


Figura 5.2: Ejemplo de uso de K-fold para la búsqueda de hiperparámetros.

para nuestras distorsiones, se optó por utilizar 80 % para entrenamiento y 20 % para test. Para las pruebas preliminares se utilizó el conjunto SJTU con las condiciones indicadas por [38], es decir, con nueve pliegues, uno por modelo para un total de 336 ejemplos para entrenamiento y 42 para test. En el caso de nuestro dataset de imágenes médicas, al ser pocos ejemplos se ha realizado un pliegue por modelo 3D, así que por cada pliegue hay un total de 350 ejemplos de entrenamiento y 35 ejemplos en test.

Primeramente, se replican los resultados obtenidos en las publicaciones originales de los métodos NR3DQA [36] y VQA-PC [38]. Para ello se utilizan los conjuntos de datos públicos SJTU [11] y WPC [87, 88]. A continuación, se analizan los resultados obtenidos sobre nuestro conjunto de imágenes médicas y, con ello, se proponen mejoras y adaptaciones.

## 5.2. Resultados

### 5.2.1. Experimentos NR3DQA

Para replicar el método de Zhang et al [36] podemos utilizar los archivos del directorio `NR3DQA/`. Para ello disponemos de unos cuántos scripts para la extracción de las características y visualización de las distribuciones como indica la publicación (véase Apéndice 7). Utilizaremos la librería de Pyntcloud [106] para obtener la matriz de covarianza y calcular las características en base a lo definido en la Sección 4.2.1. Para los conjuntos de datos SJTU y WPC, los resultados son similares a los obtenidos en la

publicación original (véase Tabla 5.1). Hay que tener en cuenta que para ambos se realiza un K-fold, donde el número K es igual al número de nubes de puntos.

Dataset	PLCC	SROCC	KROCC
SJTU	<b>0.810325</b>	<b>0.777403</b>	<b>0.608302</b>
WPC	0.637953	0.634853	0.463578

Tabla 5.1: Resultados de prueba preliminar con SVM. En SJTU tenemos una mejora de 7% respecto a la publicación original que podemos asociar al ruido de la inicialización aleatoria.

No obstante, utilizando las mismas funciones para el caso de los imágenes médicas y experimentando con múltiples modelos de regresión, vemos, en la Tabla 5.2, que el método no es capaz de determinar con precisión la calidad de imágenes, ya que la correlación entre la predicción y el valor real es muy cercano a 0. Incluso utilizando el conjunto SJTU para ayudar al entrenamiento del SVM, no obtenemos mejoras significativas, sino que obtenemos un 0.225 de SROCC.

Etiqueta Sintética	Modelo	Escalado	PLCC	SROCC
Valor de la métrica	SVM	RobustScaler	0.2017	0.1776
Valor normalizado	KNNRegressor	RobustScaler	0.2671	0.1882
Valor en escala 0-5	DecisionTree	StandardScaler	<b>0.309176</b>	<b>0.196713</b>

Tabla 5.2: Resultados de prueba preliminar NR3DQA. Vemos los mejores modelos y normalización para las diferentes escalas de las etiquetas sintéticas. Se observa que con el conjunto de imágenes médicas no hemos logrado buena correlación entre los valores predichos y el valor real.

Algo similar ocurre cuando intentamos utilizar más características geométricas como se indica en el trabajo de Weinmann et al [107]. Se argumenta que, en el proceso de segmentación, detección y clasificación de estructuras en nubes de puntos, las mejores métricas suelen ser: omnivarianza, entropía de valores singulares, la verticalidad del vecindario y otras. Los resultados de utilizar estas métricas adicionales, que se pueden observar en la Tabla 5.3, no son significativos.

En el estado del arte vimos que hay cierta inclinación al uso de modelos DL para intentar superar los resultados actuales y obtener una métrica más genérica. Se observa la dificultad del análisis de NSS a la hora de elegir que métricas deben extraerse para generar un buen vector características para un modelo ML que intenta resolver este problema. Dado el marco temporal y visto los resultados preliminares de la Sección 5.2.2, se determina conveniente hacer uso de modelos más complejos y permitir que la extracción de características sea automática.

Dataset	PLCC	SROCC	KROCC
SJTU	<b>0.853709</b>	<b>0.820057</b>	<b>0.649406</b>
WPC	0.642356	0.62917	0.455562
Nuestro	0.344601	0.170793	–

Tabla 5.3: Resultado de mejoras sobre el método SVM. Se observa mejoras, no sustanciales, sobre los conjuntos SJTU, WPC e imágenes médicas. No obstante, todavía sigue por detrás de los métodos DL, como el método VQA-PC que se discutirá más adelante.

### 5.2.2. Experimentos VQA-PC

Previo a tratar con los datos médicos, se han realizado pruebas de ejecución para verificar el funcionamiento del modelo, validar los resultados, familiarizarse con el código e identificar zonas de posibles mejoras.

#### Replicando los resultados sobre SJTU

Para validar el correcto funcionamiento del código y los resultados obtenidos en [38], realizamos el experimento en las mismas condiciones descrita por ellos en el conjunto de datos SJTU. Como ese posee 10 modelos, (véase Sección 4.1.1) se realiza un 9-fold. Se han utilizado los mismos hiperparámetros, estructura de red convolucional y transformaciones de datos (véanse Tablas 5.5 y 5.4).

Hiperparámetro	Valor
Tasa de aprendizaje	0.0004
Tamaño de batches	32
Tasa de decadencia	0.9
Frecuencia de decadencia	10
Épocas	30
K-fold	9

Tabla 5.4: Hiperparámetros empleados en la experimentación preliminar [38].

Para el conjunto de entrenamiento se recorta una zona aleatoria de la imagen con tamaño 224x224, a continuación se normaliza los colores conforme al siguiente vector de medias  $\mu = [0.485, 0.456, 0.406]$  y de desviación típica  $\sigma = [0.229, 0.224, 0.225]$ . Valores con los cuales se normalizaron las imágenes con las que entrenó ResNet [99]. Para el conjunto de test, se utiliza la misma normalización de colores, pero en vez de recortar una zona aleatoria de la imagen se recorta la parte central.

Tras pasada las 9 iteraciones, el resultado promedio es similar al estimado por el artículo original y, como se puede observar en la Figura 5.3, el modelo parece estar aprendiendo. Los resultados se reflejan en la Tabla 5.6. Como

Capa	Salida	Estructura	
Bloque inicial	112 × 112	$7 \times 7$ , 64, stride 2	
Bloque convolucional 1	56 × 56	$3 \times 3$ max pool, stride 2	
		$1 \times 1$ , 64	
		$3 \times 3$ , 64	× 3
Bloque convolucional 2	28 × 28	$1 \times 1$ , 256	
		$1 \times 1$ , 128	
		$3 \times 3$ , 128	× 3
Bloque convolucional 3	14 × 14	$1 \times 1$ , 512	
		$1 \times 1$ , 256	
		$3 \times 3$ , 256	× 3
Bloque convolucional 4	7 × 7	$1 \times 1$ , 1024	
		$1 \times 1$ , 512	
		$3 \times 3$ , 512	× 3
Bloque convolucional 5	1 × 1	$1 \times 1$ , 2048	
		average pool, 1000-d fc, softmax	
<b>Total de parámetros</b>		<b>23.803.969</b>	

Tabla 5.5: Descripción de la arquitectura ResNet50.

vemos el error de validación tiende a bajar, aunque con cierta variabilidad entre épocas en contra del error de entrenamiento que es más estable. .

Se debe tener en cuenta que, aunque el error cuadrático medio (MSE, por sus siglas en inglés) pueda parecer sustancialmente grande, nuestro criterio de elección sería la correlación entre las métricas. Es decir, no necesitamos estimar los valores de distorsión en la misma escala que las etiquetas. Apenas necesitamos ser capaces de comparar de forma ordenada las imágenes de menor a mayor calidad. En otras palabras, si la función a determinar es  $f(x) = x$ , tener  $f(x) = 100x$  sería equivalente. Por ello, de aquí en adelante utilizaremos la métrica SROCC. Cabe observar que los valores se acercan a los resultados obtenidos en la publicación original, utilizando el criterio del promedio del mejor resultado de cada pliegue de validación.

Kfold	MSE	SROCC
0	13.9222	0.8995
1	418120.5625	0.8547
2	10.9271	0.9081
3	19.8226	0.9295
4	443.6077	0.8700
5	28.3165	0.9544
6	292.239	0.7675
7	329.0685	0.8833
8	357.0455	0.8647
<b>Promedio</b>	<b>46623.94</b>	<b>0.8813</b>

Tabla 5.6: Resultados de experimento preliminar. Se enseña el MSE del modelo sin utilizar la regresión logística para normalizar las distancias.

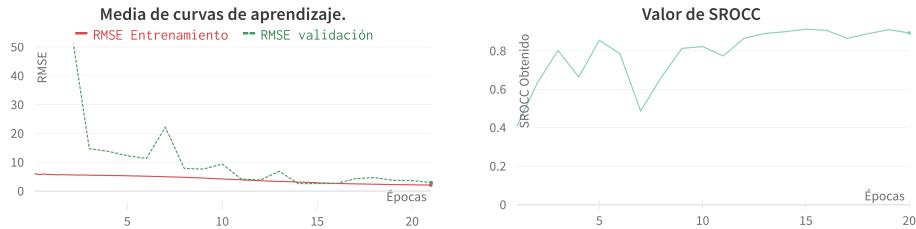


Figura 5.3: Curvas de aprendizaje del test preliminar. Cada curva describe el comportamiento medio en cada pliegue. A la derecha podemos ver el comportamiento de la métrica SROCC.

### 5.2.3. Experimentos finales

En nuestro modelo, una parte importante de la inferencia viene de la capacidad del mismo de unificar las características estáticas con las dinámicas. En la publicación [86] se proponen distintos métodos para la fusión de vectores características de modelos *ensemble*. En ella, se realiza una comparativa entre cuatro métodos: la concatenación, multiplicación, convolución 1x1 y fusión en el dominio de fourier. Argumenta que cada uno de los métodos de fusión permite la interacción de los vectores características de distinta manera. El método de concatenación, aunque es el método más habitual, genera vectores de mayor dimensionalidad, hecho que puede influir sustancialmente al tiempo de entrenamiento e inferencia. Además, no permite una interacción directa entre los vectores. Es por ello que se experimenta con cada uno de estos métodos.

La fusión por multiplicación (F1) puede provocar pérdida de información (valores cercanos a 0). No obstante, permiten una interacción directa de los vectores y genera un vector salida de menor dimensionalidad. Otro inconveniente es que tenemos que normalizar, si no son ya iguales, a la misma dimensión los vectores. De forma similar, la fusión por convolución (F2) permite realizar una proyección lineal de ambos vectores a uno de menor dimensionalidad. Por último, la fusión por *bi-linear pooling* (F3) permite realizar la multiplicación de los vectores en el dominio de fourier, de tal forma que todos los elementos afectan al resultado final de forma multiplicativa, y luego proyectar el vector resultante a una menor dimensión utilizando algún algoritmo de proyección, como *Count Sketch*.

Por último, se observa que el modelo realiza un recorte aleatorio de las imágenes de entrada, originalmente a escala 1920x1080, durante el entrenamiento para ajustarlas a una escala de 224x224, lo cual reduce el tiempo de cómputo y evita el sobre-entrenamiento. Durante la ejecución del test, se recorta el centro de la imagen. Este proceso de reescalamado, conocido como aumentación de datos, incrementa la cantidad de ejemplos y previene el sobreajuste. Sin

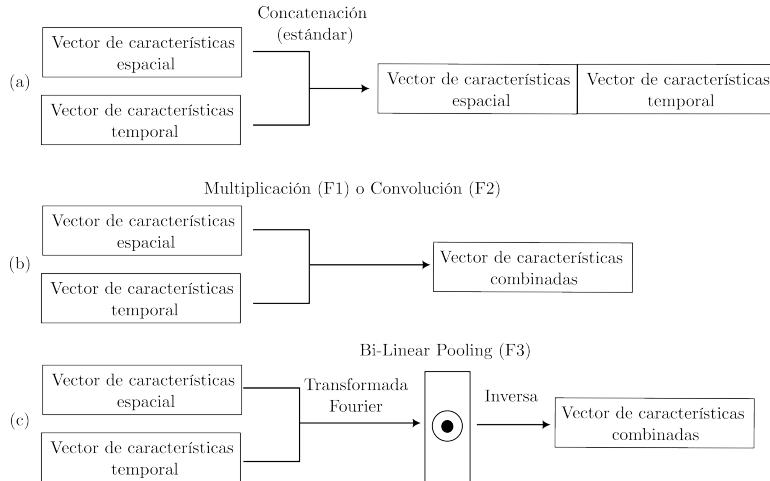


Figura 5.4: Ejemplo visual de los distintos métodos de fusión a comparar. El modelo estándar utiliza el método de concatenación (a) que genera un vector característica de alta dimensionalidad. Por otro lado, las propuestas que figuran en (b) y (c) generan un vector de menor dimensionalidad, la última operando sobre el dominio de fourier.

embargo, debido a que las nubes de puntos ocupan principalmente la zona central de la imagen y en poca proporción, dichos recortes pueden provocar ejemplos sin o con muy poca información. Esto puede ser perjudicial al modelo. Es por ello que se propone un reescalado uniforme a una escala de 398x224 para obtener siempre recortes válidos.

Para realizar la comparativa de estas mejoras, se utiliza el conjunto de datos médicos definidos anteriormente. Se hace la comparativa con las métricas sin normalizar y normalizadas a la escala 0-5, la misma con la que se entrenó VQA-PC [38] en SJTU. Se entrena por 30 épocas, con el uso de *early-stopping*, método utilizado para frenar el entrenamiento si el error de validación crece para evitar sobre-entrenamiento, con una paciencia de 6 épocas (en este contexto, paciencia alude a la espera de los resultados de las N siguientes épocas, en la que el error de validación crece, antes de terminar la ejecución por sobre-entrenamiento). Se han utilizado los mismos hiperparámetros que definido en la Tabla 5.4. Se omitirá el MSE del modelo debido a que la información principal se encuentra en la métrica SROCC como mencionado anteriormente al principio de la Sección 5.2.2.

En la Tabla 5.7 hacemos 11-fold sobre nuestro conjunto de datos teniendo en cuenta las métricas originales, las métricas normalizadas a escala 0-5 y las imágenes reescaladas. También se investiga rescalar y normalizar (última columna). Se compara estos resultados con los demás métodos de fusión de características propuesto. Vemos que el modelo con información previa, al

utilizar el reescalado, obtiene los mejores resultados. No obstante, F1 y F2 sin información previa consiguen acercarse a un margen de 4 % de ese valor. Por otro lado, la fusión F3 no logra mejorar de forma significativa.

Modelo	Valor medio SROCC			
	Estándar	Normalizado	Reescalado	Ambos
VQA-PC (SJTU)	0.7094	<b>0.6235</b>	<b>0.8425</b>	0.7126
<b>VQA-PC F1</b>	<b>0.7305</b>	0.6140	0.8164	0.7291
<b>VQA-PC F2</b>	0.6816	0.5770	0.8057	<b>0.7324</b>
<b>VQA-PC F3</b>	0.7080	0.5671	0.7482	0.7006

Tabla 5.7: Tabla de resultados iniciales sobre imágenes médicas. Partimos del modelo pre-entrenado de la publicación original [38] sobre SJTU y a continuación desde cero con los métodos de fusión de característica.

Los resultados son prometedores. El modelo con información adicional sobre otros tipos de distorsiones, conocimiento del conjunto SJTU, es el que obtiene el mejor resultado tras reescalar las imágenes, seguido por el modelo entrenado desde cero con la fusión por multiplicación (F1). No obstante, se ha observado cierta variabilidad en los resultados de cada pliegue, como se observa en la Tabla 5.8. Esto puede ser debido a diversos factores, desde la dificultad del modelo de aprender las características relevantes en tan pocas épocas, por la falta de ejemplos en este pequeño conjunto de imágenes médicas, por la variabilidad entre nubes de puntos (pocos ejemplos similares y muchas partes del cuerpo) o por dificultades en la generación de etiquetas sintéticas de calidad. Además, observándose detenidamente los valores obtenidos en cada pliegue, se observa una alta variabilidad para un ejemplo en concreto, el último pliegue, con SROCC a más de 3 desviaciones típicas para los casos F1-F3. Es por ello que se puede observar la mediana de cada modelo en la Tabla 5.9.

Para validar el rápido aprendizaje de los métodos F1-F3 y la posible mejora sobre el método de concatenación, se experimenta utilizar el modelo VQA-PC sin modificaciones desde cero sobre las imágenes reescaladas (véase Tabla 5.10). Se obtiene en media resultados algo peor que el modelo pre-entrenado, remarcando la importancia de información adicional y etapas de entrenamiento más largas a la hora de lidiar con algunas nubes de puntos, aunque su mediana es la mejor de los 4 modelos sin pre-entrenar en distorsiones. Se determina que, para un conjunto de datos pequeños, las mejoras F1-F3 no son significativas sin información adicional. Dado la importancia de la información adicional sobre las distorsiones a la hora de estimar la calidad de las imágenes de nuestro pequeño conjunto médico, se propone pre-entrenar sobre el conjunto LS-PCQA. En este caso, se utilizará las imágenes reescaladas y las métricas sin normalizar, dado que se ha observado anteriormente que es la mejor combinación. Sobre los datos de LS-PCQA, se invierte el MOS para que pase a representar el nivel de distorsión en lugar del nivel de calidad en escala 0-1, para que se asemeje a los valores sintéticos sin normalizar, y

también utilizamos imágenes reescaladas.

Se observa una mejora significativa en los métodos F2-F3, y se logra pasar la barrera del 90 % en la mediana de los métodos F0-F2 (véase Tabla 5.11). Gracias a la información adicional hemos logrado valores mucho mejores, llegando a obtener una correlación del 88 % de media, con mediana al 94 %, utilizando el modelo F2. En general, poseer información adicional ha resultado ser crucial para el aprendizaje del modelo. No obstante, la distribución de esa información es de suma importancia a la hora de determinar la capacidad de generalización del modelo. Vemos que con el nuevo conjunto de datos, el modelo F0 recibe un incremento en desviación típica al volverse incapaz de predecir correctamente el último pliegue al igual que los demás métodos.

Modelo	Desviación típica SROCC			
	Estándar	Normalizado	Reescalado	Ambos
VQA-PC (SJTU)	0.1448	0.2357	<b>0.0668</b>	0.1335
VQA-PC F1	<b>0.1222</b>	0.1402	0.1752	0.2250
VQA-PC F2	0.1462	0.1905	0.1741	<b>0.1187</b>
VQA-PC F3	0.1507	<b>0.1304</b>	0.1326	0.1462

Tabla 5.8: Desviación típica de los resultados obtenidos. Se observa que la mejora del reescalado de las imágenes incrementa la estabilidad del modelo inicial. El método de fusión F3 es el más estable en todos los casos.

Modelo	Mediana SROCC			
	Estándar	Normalizado	Reescalado	Ambos
VQA-PC (SJTU)	<b>0.7400</b>	<b>0.7510</b>	0.8417	0.7434
VQA-PC F1	0.7022	0.6331	<b>0.8636</b>	<b>0.7849</b>
VQA-PC F2	0.6350	0.5955	0.8538	0.7165
VQA-PC F3	0.7118	0.5179	0.7518	0.7334

Tabla 5.9: Mediana de los valores obtenidos. Se observa una mejora significativa para los métodos F1 y F2. También es evidente la estabilidad del modelo pre-entrenado sobre SJTU.

Modelo	SROCC		
	Media	Desviación	Mediana
VQA-PC F0	<b>0.8261</b>	0.1589	<b>0.8657</b>

Tabla 5.10: Resultados del método original con modelo original sin pre-entrenar sobre imágenes médicas reescaladas.

Modelo	SROCC		
	Media	Desviación	Mediana
VQA-PC F0	0.8325	0.2017	0.9140
VQA-PC F1	0.8242	0.2025	0.9095
VQA-PC F2	<b>0.8757</b>	<b>0.1468</b>	<b>0.9347</b>
VQA-PC F3	0.8071	0.1811	0.8692

Tabla 5.11: Resultados en imágenes médicas reescaladas con modelos pre-entrenados sobre el conjunto de datos LS-PCQA.

### 5.3. Discusión de resultados

El modelo adaptado de ML demuestra no ser capaz de determinar con calidad el nivel de distorsiones de las imágenes médicas. Incluso tras la implementación de la extracción de nuevas características según publicaciones recientes de segmentación de nubes de puntos. Esta mejora logra un incremento del 5 % de correlación en un conjunto de datos públicos llegando al 85 %, pero apenas consigue un 20 % sobre los datos médicos. Por otro lado, al transferir el modelo DL elegido directamente a imágenes médicas ya se consiguen buenos resultados, con una correlación media del 71 %. Con las mejoras propuestas de fusión de características se obtiene una mayor media con el método F1 ( fusión por multiplicación) a 73 %. No obstante, se justifica el reescalado de las imágenes para mejorar el ratio de información por recorte y se obtiene un 84 % de correlación media con el modelo estándar. Por último, tras entrenar con datos sintéticos de distorsiones similares y diversas nubes de puntos, se logra una media de 88 %, con mediana al 94 %, utilizando el método F2 ( fusión por convolución). En la Figura 5.5 se observa un ejemplo de la correlación entre los valores de calidad inferidos, para distintas intensidades de submuestreo, y los valores de las etiquetas sintéticas correspondientes.

De los distintos métodos de fusión de características, se concluye que para pocos conjuntos de datos los métodos F1, F2 y F3 ( fusión por *pooling* bilineal) quedan por detrás, aunque muy cerca, del método habitual de concatenar los vectores (F0). Siendo F3 el método más estable para las distintas transformaciones de los datos (reescalado y normalización). No obstante, es a la vez el método que obtiene los peores resultados. Cuando pre-entrenamos en un conjunto de datos más largo, el método F3 sigue siendo el con peores resultados, aunque más cerca de F1 que, a su vez, se acerca a F0. Por otro lado, con más datos F2 tiene un gran salto en sus resultados, logrando ser el mejor método de fusión de características.

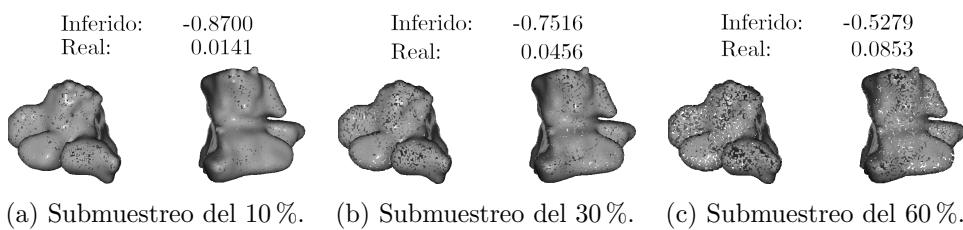


Figura 5.5: Ejemplo de correspondencia de pendiente entre valores inferidos (sin normalizar) y los valores reales de las etiquetas.



## Capítulo 6

# Conclusiones y Trabajos Futuros

La estimación de la calidad de imágenes (IQA) es un problema esencial a la hora de optimizar el formato y visualización de la información, además es de suma importancia para el ámbito biomédico. Este TFG aborda la obtención de una métrica de estimación de calidad capaz de evaluar representaciones 3D sin referencia, en concreto nubes de puntos, del ámbito biomédico para poder asistir en la mejora de los algoritmos de reconstrucción y visualización de dichos objetos.

En primer lugar, se realizó un estudio de la literatura relativa a la estimación de calidad de imágenes 2D, desde los métodos basados en extracción de características de escenas naturales y modelos de ML, hasta la extracción automática con DL. A continuación, se estudió el uso de estos y otros métodos sobre imágenes médicas 2D. Posteriormente se analizó el estado del arte de métodos dedicados a representaciones tridimensionales. Se observa un salto en complejidad teórica y computacional al tratarse de problemas en tres dimensiones. Por último, se concluye que no existe hasta el momento otra investigación que haya tomado el enfoque novedoso de estimar la calidad de reconstrucciones biomédicas 3D.

Ante la falta de propuestas específicas, el trabajo parte de la implementación de métodos relevantes del estado del arte de estimación de calidad de objetos 3D, tanto desde la perspectiva de métodos tradicionales (ML, en donde la extracción de características y la clasificación son etapas independientes) como de métodos *end-to-end* DL. El primero hace uso de características extraídas manualmente utilizando conocimiento humano sobre el sistema visual humano (HVS), como fenómenos de planaridad, esfericidad, anisotropía, curvatura, linealidad y consistencia de colores de las nubes de puntos, que luego se utilizan para estimar una regresión por SVM. En cuanto a modelos

basados en DL, se utilizó un modelo capaz de extraer información estática y dinámica de nubes de puntos haciendo uso de múltiples proyecciones 2D y de un vídeo del objeto 3D rotando. De esta forma, podemos simular el HVS. En ambos casos se proponen ajustes y pequeñas mejoras basadas en recientes publicaciones y se comparan los resultados con la propuesta original.

Para la validación sobre un conjunto de datos médicos fue necesaria la creación de un conjunto de datos sintético debido a la no existencia de un conjunto de datos públicos para este análisis. Para ello se estudiaron y se fabricaron las distorsiones más comunes del ámbito biomédico con respecto a las representaciones 3D. Para evitar la problemática logística del etiquetado a través de la evaluación humana sobre el dataset sintético, fue necesario estudiar el problema IQA con referencia y hacer uso de las métricas más empleadas. Dichas métricas demostraron una alta correlación con el HVS, justificando así su uso para generar etiquetas artificiales. Se generaron un total de 385 representaciones médicas 3D distorsionadas, 11 nubes de puntos base, 5 distorsiones a 7 niveles cada una. En las distorsiones se simula tanto errores de transmisión, compresión como el movimiento del paciente.

Como primera conclusión de nuestra experimentación base, siguiendo la tendencia del estado del arte, el modelo DL sale exitoso en la comparativa sobre objetos 3D genéricos. Dicha conclusión es consecuencia de lograr replicar satisfactoriamente los resultados de los métodos estudiados sobre los conjuntos de datos públicos. A continuación, se observa que el modelo adaptado de ML (NR3DQA) demuestra no ser capaz de determinar con calidad el nivel de distorsiones de las imágenes médicas. Sin embargo, el modelo basado en DL (VQA-PC) consigue resultados aceptables con una correlación media del 71 %. Finalmente, se aplican mejoras a los métodos y se concluye que, tras entrenar con datos sintéticos de distorsiones similares y diversas nubes de puntos, se obtiene una mejora considerable en el modelo basado en DL. En concreto, se alcanza una alta correlación (88 %) utilizando la aproximación F2 ( fusión por convolución).

Por lo tanto, se concluye que se han completado satisfactoriamente los objetivos planteados, determinando la posibilidad de resolución del problema adaptado al ámbito biomédico y abriendo puertas a futuras investigaciones. Todo el código se encuentra disponible en el siguiente repositorio de GitHub [https://github.com/CodeBoy-source/TFG\\_NRPCQA](https://github.com/CodeBoy-source/TFG_NRPCQA), a excepción de las imágenes médicas que son datos confidenciales.

Siendo un proyecto en una nueva línea de investigación, existen varias ampliaciones lógicas que se pueden realizar a este proyecto. Por un lado, se podría obtener una etiqueta manual con un experimento de evaluación, según los estándares, para obtener una opinión media de calidad (MOS) y volver a validar los resultados obtenidos entre los distintos modelos. Así como utilizar ese conjunto de MOS manual sobre imágenes médicas para normalizar las eti-

quetas sintéticas como lo hacen en la publicación original [85], donde utilizan un conjunto pequeño extraído manualmente para normalizar uno sintético varias veces más grande. También, para mejorar el método propuesto, se podría permitir que los pesos del modelo utilizado para la extracción de características del vídeo fueran alterados en vez de ser solamente un paso previo, de extracción. De esta forma se podría guiar el modelo a buscar nuevas características temporales. Además, se podría buscar simular las distorsiones sobre el conjunto de imágenes 2D generadas tras el examen en vez de hacerlo sobre la representación 3D final, teniendo así datos más realistas.

Por otro lado, se pueden explorar otros métodos que procesen modelos 3D directamente, o que hagan uso de proyecciones y de la nube de puntos simultáneamente, como en MM-PCQA [39]. Actualmente, ha crecido el número de publicaciones de adaptaciones de PointNet [58] y PointNet++ [108] para resolver distintos problemas de nubes de puntos, por lo que quizás se podría adaptar para la resolución de este problema, como el método de ResSCNN [85] y evitar así la pérdida de información al proyectar en 2D.



## Capítulo 7

# Bibliografía

- [1] F. Fol Leymarie, M.-C. Chang, C. Imielinska y B. Kimia, «A General Approach to Model Biomedical Data from 3D Unorganised Point Clouds with Medial Scaffolds.,» 2010, págs. 65-74.
- [2] Y. Ding, *Visual Quality Assessment for Natural and Medical Image*, 1.<sup>a</sup> ed. 2018.
- [3] G. Zhai y X. Min, «Perceptual image quality assessment: a survey,» *Science China Information Sciences*, vol. 63, págs. 1-52, nov. de 2020.
- [4] C. Z. Ke Gu Hongyan Liu, *Quality Assessment of Visual Content*, 1.<sup>a</sup> ed. Springer, 2022.
- [5] K. Seshadrinathan et al., «Image Quality Assessment,» *The Essential Guide to Image Processing*, págs. 553-595, 2009.
- [6] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2.<sup>a</sup> ed. Springer Nature, 2022.
- [7] R. Hartley y A. Zisserman, *Multiple View Geometry in Computer Vision*, 2.<sup>a</sup> ed. Cambridge University Press, 2011.
- [8] D. A. Forsyth y J. Ponce, *Computer Vision: A Modern Approach*, 2.<sup>a</sup> ed. Pearson, 2012.
- [9] M. Sonka, V. Hlavac y R. Boyle, *Image Processing, Analysis, and Machine Vision*, 4.<sup>a</sup> ed. Cengage Learning, 2015.
- [10] Z. Wang y A. C. Bovik, «Modern Image Quality Assessment,» en *Modern Image Quality Assessment*, vol. 1, 2006, págs. 1-146.
- [11] Q. Yang, H. Chen, Z. Ma, Y. Xu, R. Tang y J. Sun, «Predicting the Perceptual Quality of Point Cloud: A 3D-to-2D Projection-Based Exploration,» *IEEE Transactions on Multimedia*, 2020.
- [12] T.-J. Liu, W. Lin y C.-C. J. Kuo, «Image Quality Assessment Using Multi-Method Fusion,» *IEEE Transactions on Image Processing*, vol. 22, n.<sup>o</sup> 5, págs. 1793-1807, 2013.

- [13] A. Balanov, A. Schwartz, Y. Moshe y N. Peleg, «Image quality assessment based on DCT subband similarity,» en *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, págs. 2105-2109.
- [14] I. Bakurov, M. Buzzelli, R. Schettini, M. Castelli y L. Vanneschi, «Structural similarity index (SSIM) revisited: A data-driven approach,» *Expert Systems with Applications*, vol. 189, pág. 116 087, 2022.
- [15] L. Zhang, Y. Shen y H. Li, «VSI: A Visual Saliency-Induced Index for Perceptual Image Quality Assessment,» *IEEE Transactions on Image Processing*, vol. 23, n.º 10, págs. 4270-4281, 2014.
- [16] J. Wu, J. Ma, F. Liang, W. Dong, G. Shi y W. Lin, «End-to-End Blind Image Quality Prediction with Cascaded Deep Neural Network,» *IEEE Transactions on Image Processing*, vol. 29, págs. 7414-7426, 2020.
- [17] «Video Multi-Method Assessment Fusion.» (2016), URL: <https://github.com/Netflix/vmaf> (visitado 01-07-2023).
- [18] R. Rassool, «VMAF reproducibility: Validating a perceptual practical video quality metric,» en *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2017, págs. 1-2.
- [19] K. Ding, K. Ma, S. Wang y E. P. Simoncelli, «Comparison of Full-Reference Image Quality Models for Optimization of Image Processing Systems,» *International Journal of Computer Vision*, vol. 129, n.º 4, págs. 1258-1281, 2021.
- [20] Z. Wang, «Applications of Objective Image Quality Assessment Methods [Applications Corner],» *IEEE Signal Processing Magazine (SPM)*, vol. 28, n.º 6, págs. 137-142, 2011.
- [21] I. Stepien y M. Oszust, «A Brief Survey on No-Reference Image Quality Assessment Methods for Magnetic Resonance Images,» *Journal of Imaging*, vol. 8, n.º 6, 2022.
- [22] C. Parisot, «The DICOM standard,» *The International Journal of Cardiac Imaging*, vol. 11, n.º 3, págs. 171-177, 1995.
- [23] K. H. Höhne, H. Fuchs y S. M. Pizer, *3D imaging in medicine: algorithms, systems, applications*, 1.<sup>a</sup> ed. 1990.
- [24] O. H. Karatas y E. Toy, «Three-dimensional imaging techniques: A literature review,» *European Journal of Dentistry*, vol. 8, págs. 132-140, 2014.

- [25] L. H. G. A. Hopman et al., «Quantification of left atrial fibrosis by 3D late gadolinium-enhanced cardiac magnetic resonance imaging in patients with atrial fibrillation: impact of different analysis methods,» *European Heart Journal - Cardiovascular Imaging*, vol. 23, n.º 9, págs. 1182-1190, 2021.
- [26] A. Fedorov et al., «3D Slicer as an image computing platform for the Quantitative Imaging Network,» *Magnetic Resonance Imaging*, vol. 30, n.º 9, págs. 1323-1341, 2012.
- [27] Y. Sun y G. Mogos, «Impact of Visual Distortion on Medical Images,» *IAENG International Journal of Computer Science*, vol. 49, págs. 36-45, 2022.
- [28] E. Kjelle y C. Chilanga, «The assessment of image quality and diagnostic value in X-ray images: a survey on radiographers' reasons for rejecting images,» *Insights into Imaging*, vol. 13, n.º 1, pág. 36, 2022.
- [29] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 8.<sup>a</sup> ed. Palgrave Macmillan, 2005.
- [30] Z. Wang, A. C. Bovik y L. Lu, «Why is image quality assessment so difficult?» En *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, 2002, págs. IV-3313-IV-3316.
- [31] Z. Wang, H. Sheikh y A. Bovik, «No-reference perceptual quality assessment of JPEG compressed images,» en *Proceedings. International Conference on Image Processing*, vol. 1, 2002, págs. I-I.
- [32] W. Zhang, K. Ma, J. Yan, D. Deng y Z. Wang, «Blind Image Quality Assessment Using a Deep Bilinear Convolutional Neural Network,» *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, págs. 36-47, 1 2020.
- [33] K.-Y. Lin y G. Wang, «Hallucinated-IQA: No-Reference Image Quality Assessment via Adversarial Learning,» en *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, págs. 732-741.
- [34] K. Ma, W. Liu, T. Liu, Z. Wang y D. Tao, «DipIQ: Blind Image Quality Assessment by Learning-to-Rank Discriminable Image Pairs,» *IEEE Transactions on Image Processing*, vol. 26, págs. 3951-3964, 8 2017.
- [35] W. Zhou, Q. Yang, Q. Jiang, G. Zhai y W. Lin, «Blind Quality Assessment of 3D Dense Point Clouds with Structure Guided Resampling,» 2022. arXiv: 2208.14603.
- [36] Z. Zhang, W. Sun, X. Min, T. Wang, W. Lu y G. Zhai, «No-Reference Quality Assessment for 3D Colored Point Cloud and Mesh Models,» *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, n.º 11, págs. 7618-7631, 2022.

- [37] Z. Shan et al., «GPA-Net: No-Reference Point Cloud Quality Assessment with Multi-task Graph Convolutional Network,» 2023. arXiv: 2210.16478.
- [38] Z. Zhang et al., «Treating Point Cloud as Moving Camera Videos: A No-Reference Quality Assessment Metric,» 2022. arXiv: 2208.14085.
- [39] Z. Zhang et al., «MM-PCQA: Multi-Modal Learning for No-reference Point Cloud Quality Assessment,» 2023. arXiv: 2209.00244.
- [40] Q. Yang, Y. Liu, S. Chen, Y. Xu y J. Sun, «No-Reference Point Cloud Quality Assessment via Domain Adaptation,» 2022. arXiv: 2112.02851.
- [41] S. Russell y P. Norvig, *Artificial Intelligence: A Modern Approach*, 4.<sup>a</sup> ed. Prentice Hall, 2010.
- [42] Y. S. Abu-Mostafa, M. Magdon-Ismail y H.-T. Lin, *Learning From Data*. AMLBook, 2012.
- [43] T. M. Mitchell, «Machine Learning,» en *Machine Learning*, McGraw-Hill, 1997.
- [44] O. Maimon y L. Rokach, eds., *Data mining and knowledge discovery handbook*, 2.<sup>a</sup> ed. Springer Science+Business Media, LLC, 2010.
- [45] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*. MIT Press, 2016.
- [46] Y. LeCun, Y. Bengio y G. Hinton, «Deep learning,» *Nature*, vol. 521, n.<sup>o</sup> 7553, págs. 436-444, 2015.
- [47] J. Schmidhuber, «Deep learning in neural networks: An overview,» *Neural Networks*, vol. 61, págs. 85-117, 2015.
- [48] C. Bishop, *Neural networks for pattern recognition*. Oxford University Press, 1995.
- [49] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.
- [50] Z. Meng, Y. Hu y C. Ancey, «Using a Data Driven Approach to Predict Waves Generated by Gravity Driven Mass Flows,» *Water*, vol. 12, n.<sup>o</sup> 2, pág. 600, 2020.
- [51] E. Akgün y M. Demir, «Modeling Course Achievements of Elementary Education Teacher Candidates with Artificial Neural Networks,» *International Journal of Assessment Tools in Education*, vol. 5, pág. 19, 2018.
- [52] 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.<sup>o</sup> 15, págs. 11 051-11 068, 2020.

- [53] Y. LeCun et al., «Backpropagation Applied to Handwritten Zip Code Recognition,» *Neural Computation*, vol. 1, págs. 541-551, 1989.
- [54] R. Yamashita, M. Nishio, R. K. G. Do y K. Togashi, «Convolutional neural networks: an overview and application in radiology,» *Insights into Imaging*, vol. 9, n.º 4, págs. 611-629, 2018.
- [55] Z. Rguibi, A. Hajami, D. Zitouni, A. Elqaraoui y A. Bedraoui, «CXAI: Explaining Convolutional Neural Networks for Medical Imaging Diagnostic,» *Electronics*, vol. 11, n.º 11, 2022.
- [56] K. O'Shea y R. Nash, «An Introduction to Convolutional Neural Networks,» 2015. arXiv: 1511.08458.
- [57] C. Feichtenhofer, H. Fan, J. Malik y K. He, «SlowFast Networks for Video Recognition,» 2019. arXiv: 1812.03982.
- [58] R. Q. Charles, H. Su, M. Kaichun y L. J. Guibas, «PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,» en *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, págs. 77-85.
- [59] B. Gokberk, H. Dutagaci, A. Ulaş, L. Akarun y B. Sankur, «Representation Plurality and Fusion for 3-D Face Recognition,» *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, págs. 155-173, 2008.
- [60] D. Müller, I. Soto-Rey y F. Kramer, «An Analysis on Ensemble Learning optimized Medical Image Classification with Deep Convolutional Neural Networks,» 2022.
- [61] «Medical Visualization and Volume Rendering.» (2019), URL: <https://sbme-tutorials.github.io/2019/CG/notes/7-week7.html> (visitado 01-06-2023).
- [62] D. M. Chandler y S. S. Hemami, «VSNR: A Wavelet-Based Visual Signal-to-Noise Ratio for Natural Images,» *IEEE Transactions on Image Processing*, vol. 16, n.º 9, págs. 2284-2298, 2007.
- [63] K. Egiazarian, J. Astola, V. Lukin, F. Battisti y M. Carli, «A new full-reference quality metrics based on hvs,» 2006.
- [64] Z. Wang y A. Bovik, «A universal image quality index,» *IEEE Signal Processing Letters*, vol. 9, n.º 3, págs. 81-84, 2002.
- [65] S. Wang, C. Deng, B. Zhao, G.-B. Huang y B. Wang, «Gradient-based no-reference image blur assessment using extreme learning machine,» *Neurocomputing*, vol. 174, págs. 310-321, 2016.
- [66] Y. Zhan y R. Zhang, «No-Reference JPEG Image Quality Assessment Based on Blockiness and Luminance Change,» *IEEE Signal Processing Letters*, vol. 24, n.º 6, págs. 760-764, 2017.

- [67] C. Yim y A. C. Bovik, «Quality Assessment of Deblocked Images,» *IEEE Transactions on Image Processing*, vol. 20, n.<sup>o</sup> 1, págs. 88-98, 2011.
- [68] Y. Fang, K. Ma, Z. Wang, W. Lin, Z. Fang y G. Zhai, «No-Reference Quality Assessment of Contrast-Distorted Images Based on Natural Scene Statistics,» *IEEE Signal Processing Letters*, vol. 22, n.<sup>o</sup> 7, págs. 838-842, 2015.
- [69] A. Mittal, A. K. Moorthy y A. C. Bovik, «No-reference image quality assessment in the spatial domain,» *IEEE Transactions on Image Processing*, vol. 21, n.<sup>o</sup> 12, págs. 4695-4708, 2012.
- [70] W. Zhou, L. Yu, W. Qiu, Y. Zhou y M. Wu, «Local Gradient Patterns (LGP): An Effective Local-Statistical-Feature Extraction Scheme for No-Reference Image Quality Assessment,» *Information Sciences*, vol. 397, 2017.
- [71] L. Kang, P. Ye, Y. Li y D. Doermann, «Convolutional Neural Networks for No-Reference Image Quality Assessment,» en *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, págs. 1733-1740.
- [72] S. Bosse, D. Maniry, K.-R. Müller, T. Wiegand y W. Samek, «Deep Neural Networks for No-Reference and Full-Reference Image Quality Assessment,» *IEEE Transactions on Image Processing*, vol. 27, n.<sup>o</sup> 1, págs. 206-219, 2018.
- [73] H. R. Sheikh, M. F. Sabir y A. C. Bovik, «A statistical evaluation of recent full reference image quality assessment algorithms,» *IEEE Transactions on Image Processing*, vol. 15, n.<sup>o</sup> 11, págs. 3440-3451, 2006.
- [74] H. Sheikh, Z. Wang, L. Cormack y A. Bovik. «LIVE image quality assessment database.» (2004), URL: <http://live.ece.utexas.edu/research/quality> (visitado 01-06-2023).
- [75] E. Chandler y D.M. «Categorical image quality (CSIQ) database.» (2009), URL: <http://vision.okstate.edu/csiq> (visitado 01-06-2023).
- [76] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli y F. Battisti, «TID2008 - A Database for Evaluation of Full-Reference Visual Quality Assessment Metrics,» *Advances of Modern Radioelectronics*, vol. 10, págs. 30-45, 2009.
- [77] Z. Wang, E. Simoncelli y A. Bovik, «Multiscale structural similarity for image quality assessment,» en *The Thirtly-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2, 2003, 1398-1402 Vol.2.

- [78] Y. Niu, Y. Zhong, W. Guo, Y. Shi y P. Chen, «2D and 3D Image Quality Assessment: A Survey of Metrics and Challenges,» *IEEE Access*, vol. 7, págs. 782-801, 2019.
- [79] E. Alexiou y T. Ebrahimi, «Towards a Point Cloud Structural Similarity Metric,» en *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, jun. de 2020.
- [80] G. Meynet, Y. Nehmé, J. Digne y G. Lavoué, «PCQM: A Full-Reference Quality Metric for Colored 3D Point Clouds,» en *IEEE International Conference on Quality of Multimedia Experience (QoMEX)*, 2020.
- [81] Q. Liu, H. Su, T. Chen, H. Yuan y R. Hamzaoui, «No-reference Bitstream-layer Model for Perceptual Quality Assessment of V-PCC Encoded Point Clouds,» *IEEE Transactions on Multimedia*, págs. 1-1, 2022.
- [82] A. Chetouani, M. Quach, G. Valenzise y F. Dufaux, «Deep Learning-Based Quality Assessment Of 3d Point Clouds Without Reference,» en *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 2021, págs. 1-6.
- [83] Q. Liu et al., «PQA-Net: Deep No Reference Point Cloud Quality Assessment via Multi-View Projection,» *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, n.º 12, págs. 4645-4660, 2021.
- [84] E. Alexiou y T. Ebrahimi, «Exploiting user interactivity in quality assessment of point cloud imaging,» en *IEEE International Conference on Quality of Multimedia Experience (QoMEX)*, jul. de 2019.
- [85] Y. Liu, Q. Yang, Y. Xu y L. Yang, «Point Cloud Quality Assessment: Dataset Construction and Learning-based No-Reference Metric,» 2022. arXiv: 2012.11895.
- [86] I. Abouelaziz, A. Chetouani, M. El Hassouni, L. J. Latecki y H. Cherifi, «No-reference mesh visual quality assessment via ensemble of convolutional neural networks and compact multi-linear pooling,» *Pattern Recognition*, vol. 100, pág. 107174, 2020.
- [87] Q. Liu, H. Su, Z. Duanmu, W. Liu y Z. Wang, «Perceptual Quality Assessment of Colored 3D Point Clouds,» *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, págs. 1-1, 2022.
- [88] H. Su, Z. Duanmu, W. Liu, Q. Liu y Z. Wang, «Perceptual quality assessment of 3D point clouds,» en *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, págs. 3182-3186.
- [89] L. S. Chow y R. Paramesran, «Review of medical image quality assessment,» *Biomedical Signal Processing and Control*, vol. 27, págs. 145-154, 2016.

- 
- [90] V. Bhateja, M. Nigam, A. S. Bhaduria y A. Arya, «Two-stage multi-modal MR images fusion method based on Parametric Logarithmic Image Processing (PLIP) Model,» *Pattern Recognition Letters*, vol. 136, págs. 25-30, 2020.
  - [91] J. Xu et al., «Semi-Supervised Learning for Fetal Brain MRI Quality Assessment with ROI consistency,» 2020. arXiv: 2006.12704.
  - [92] A. Tarvainen y H. Valpola, «Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,» 2018. arXiv: 1703.01780.
  - [93] S. Liu, K.-H. Thung, W. Lin, D. Shen y P.-T. Yap, «Hierarchical Nonlocal Residual Networks for Image Quality Assessment of Pediatric Diffusion MRI With Limited and Noisy Annotations,» *IEEE Transactions on Medical Imaging*, vol. 39, n.º 11, págs. 3691-3702, 2020.
  - [94] T. Iqbal y H. Ali, «Generative Adversarial Network for Medical Images (MI-GAN),» *Journal of Medical Systems*, vol. 42, n.º 11, 2018.
  - [95] K. Qi et al., «Blind Image Quality Assessment for MRI with A Deep Three-dimensional content-adaptive Hyper-Network,» 2021. arXiv: 2107.06888.
  - [96] L. S. Chow y H. Rajagopal, «Modified-BRISQUE as no reference image quality assessment for structural MR images,» *Magnetic Resonance Imaging*, vol. 43, 2017.
  - [97] R. Schnabel y R. Klein, «Octree-based Point-Cloud Compression.,» *Eurographics Symposium on Point-Based Graphics*, págs. 111-120, 2006.
  - [98] H. Edgar, S. Daneshvari Berry, E. Moes, N. Adolphi, P. Bridges y K. Nolte, *New Mexico Decedent Image Database*, Office of the Medical Investigator, University of New Mexico, 2020.
  - [99] K. He, X. Zhang, S. Ren y J. Sun, «Deep Residual Learning for Image Recognition,» 2015. arXiv: 1512.03385.
  - [100] ITU-R, «Methodology for the Subjective Assessment of the Quality of Television Pictures,» International Telecommunication Union - Radiocommunication Sector (ITU-R), inf. téc. BT.500-13, 2012.
  - [101] J. Zhou et al., «Subjective quality analyses of stereoscopic images in 3DTV system,» 2011.
  - [102] R. Mekuria, Z. Li, C. Tulvan y P. Chou, «Evaluation criteria for PCC (Point Cloud Compression),» en *ISO/IEC MPEG Doc. N16332*, jul. de 2016.

- [103] D. Tian, H. Ochimizu, C. Feng, R. Cohen y A. Vetro, «Geometric distortion metrics for point cloud compression,» en *IEEE International Conference on Image Processing (ICIP)*, sep. de 2017.
- [104] Q. Yang, Z. Ma, Y. Xu, Z. Li y J. Sun, «Inferring Point Cloud Quality via Graph Similarity,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, n.º 6, págs. 3015-3029, 2022.
- [105] Q. Yang, Y. Zhang, S. Chen, Y. Xu, J. Sun y Z. Ma, «MPED: Quantifying Point Cloud Distortion based on Multiscale Potential Energy Discrepancy,» 2022. arXiv: 2103.02850.
- [106] «Pyntcloud.» (2019), URL: <https://pyntcloud.readthedocs.io/en/latest/> (visitado 01-06-2023).
- [107] M. Weinmann, B. Jutzi, C. Mallet y M. Weinmann, «Geometric Features and Their Relevance for 3D Point Cloud Classification,» *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-1/W1,
- [108] C. R. Qi, L. Yi, H. Su y L. J. Guibas, «PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space,» 2017. arXiv: 1706.02413.
- [109] R. B. Rusu y S. Cousins, «3D is here: Point Cloud Library (PCL),» en *2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2011, págs. 1-4.
- [110] Q.-Y. Zhou, J. Park y V. Koltun, «Open3D: A Modern Library for 3D Data Processing,» 2018. arXiv: 1801.09847.
- [111] «fastai.» (2016), URL: <https://docs.fast.ai/> (visitado 01-07-2023).



## Búsquedas en Scopus

### 1. Inteligencia artificial en imágenes médicas

TITLE-ABS-KEY ((deep AND learning) OR (machine AND learning) OR (artificial AND intelligence) OR (computer AND vision) OR (soft AND computing) AND ((biomedical AND image AND analysis) OR (medical AND imaging) OR (medical AND image AND analysis))) AND (LIMIT-TO (SUBJAREA , “COMP”) OR LIMIT-TO (SUBJAREA , “MEDI”) OR LIMIT-TO (SUBJAREA , “ENGI”))

### 2. Inteligencia artificial en nubes de puntos

TITLE-ABS-KEY ((deep AND learning) OR (machine AND learning) OR (artificial AND intelligence) OR (computer AND vision) OR (soft AND computing) AND ((point AND cloud) OR (3d OR tridimensional))) AND (LIMIT-TO (SUBJAREA , “COMP”) OR LIMIT-TO (SUBJAREA , “MEDI”) OR LIMIT-TO (SUBJAREA , “ENGI”))

### 3. Estimación de calidad en imágenes médicas

TITLE-ABS-KEY ((deep AND learning) OR (machine AND learning) OR (artificial AND intelligence) OR (computer AND vision) OR (soft AND computing) AND ((biomedical AND image AND analysis) OR (medical AND imaging) OR (medical AND image AND analysis)) AND ((quality AND assessment) OR (quality AND estimation) OR (mos))) AND (LIMIT-TO (SUBJAREA , “COMP”) OR LIMIT-TO (SUBJAREA , “MEDI”) OR LIMIT-TO (SUBJAREA , “ENGI”))

### 4. Estimación de calidad en nubes de puntos

TITLE-ABS-KEY ((deep AND learning) OR (machine AND learning) OR (artificial AND intelligence) OR (computer AND vision) OR (soft AND computing) AND ((point AND cloud) OR (3d OR tridimensional)) AND ((quality AND assessment) OR (quality AND estimation) OR (mos))) AND (LIMIT-TO (SUBJAREA , “COMP”) OR LIMIT-TO (SUBJAREA , “MEDI”) OR LIMIT-TO (SUBJAREA , “ENGI”))

### 5. Estimación de calidad de imágenes médicas 3D

TITLE-ABS-KEY ((deep AND learning) OR (machine AND learning) OR (artificial AND intelligence) OR (computer AND vision) OR (soft AND computing) AND ((biomedical OR medical OR medicine)) AND ((point AND cloud) OR (3d OR tridimensional)) AND ((quality AND assessment) OR (quality AND estimation) OR (mos))) AND (LIMIT-TO (SUBJAREA , “COMP”) OR LIMIT-TO (SUBJAREA , “MEDI”) OR LIMIT-TO (SUBJAREA , “ENGI”))

## Detalles Técnicos de Implementación

Este proyecto ha sido realizado mayoritariamente con el lenguaje de programación Python, debido a que casi todos los modelos analizados estaban descritos en el mismo. No obstante, para la distorsión por compresión *octree* [97], se hizo uso de la librería PCL [109] en el lenguaje C++.

Para el desarrollo y ejecución de los modelos fue necesario el uso de la librería de DL Pytorch junto con las librerías CUDA de para poder ejecutar los modelos en las tarjetas gráficas de NVIDIA. Para los cálculos numéricos y el manejo de datos se utilizaron Numpy y Polars, librería similar a Pandas pero basada en Rust, más eficiente y fácilmente paralelizable. Además, para el cálculo de las métricas se utilizó la librería scikit-learn. Para la visualización y fácil manipulación de las nubes de puntos se hizo uso de la librería de Open3D [110] y Pyntcloud [106]. Se ha gestionado el uso de estas librerías y todas sus dependencias tanto en entornos virtuales de python como en entornos creados por cuadernos jupyter de Colab.

Para el control de versiones del proyecto se utilizó de forma conjunta Git, GitHub y la gestión de versiones de Google Drive. El repositorio de este proyecto se puede acceder por la siguiente dirección: [https://github.com/CodeBoy-source/TFG\\_NRPCQA](https://github.com/CodeBoy-source/TFG_NRPCQA). Este mismo, se encuentra dividido en un conjunto de carpetas:

- **Distort**, donde se encuentra todo lo necesario para la generación de las distorsiones médicas dado un directorio de archivos .ply. A su vez, posee lo necesario para la generación de las etiquetas sintéticas de calidad, ver Sección 7.
- **Document**, donde se encuentra la documentación del proyecto, incluyendo a este documento.
- **NR3DQA**, implementación y experimentos del método propuesto por Zhang et al[36].
- **Utils**, conjunto de scripts de python para la realización de distintas tareas. Como por ejemplo la lectura de un directorio DICOM, la visualización de una o un conjunto de nubes de puntos y división del conjunto de datos LS-SJTU-PCQA[85].
- **VQA\_PC**, implementación de la variante VQA-PC[38] para la estimación de calidad de nubes de puntos y las modificaciones pertinentes sobre los métodos de fusión de características mencionados en [86].

### Generación de un conjunto de datos de imágenes médicas.

Los datos se encuentran en una carpeta del servicio UGRDrive, que provee almacenamiento en la nube para investigadores. Los modelos mencionados en la Sección 4.1.2 se encuentran dentro de una carpeta numerada por cada individuo con los ficheros necesarios para el desarrollo del proyecto. Se incluyen incluso algunos directorios DICOM enteros por si fuera necesario generar más datos a partir de la segmentación manual.

Se desarrolló un fichero `gen_distortions.py` que automáticamente genera un conjunto de distorsiones dado un directorio de entrada con archivos `.ply` y los guarda en un directorio de salida especificado por argumento. Para ello se hace uso de las distorsiones realizadas con Open3D [110] con el archivo del directorio `utils/distortions.py` y un ejecutable hecho con C++ y Makefile para la distorsión `octree`. A continuación, podemos generar las etiquetas sintéticas con `get_metrics.py`, que dado un directorio de entrada con las nubes de referencia y uno con las distorsiones, genera un `.csv` con las etiquetas sintéticas generadas con las métricas del estado del arte de los métodos FR-PCQA. Para ello se hace uso de un software desarrollado con PCL [109] y el archivo del directorio `utils/metrics.py`.

### Preprocesado de datos

El único preprocesado que sufren los datos iniciales es el centrado de la nube de puntos sobre los ejes, paso previo a la rotación. Y la reducción de puntos anormales por medio de un análisis de consistencia estadística del vecindario, eliminando así puntos aislados y ruido.

El proceso es muy sencillo, dado el vecindario de un punto definido por sus K vecinos más cercanos, calculamos la desviación típica y la media de sus atributos geométricos y eliminamos aquellos que sobrepasen un umbral determinado. En nuestro caso utilizamos K = 32 y el umbral a 5 desviaciones típicas. Para ello se puede utilizar `Utils/std_remove.py`.

### Distorsiones

#### Ruido Gaussiano

Para la generación del ruido gaussiano, que en este caso simula posibles errores de transmisión y generación, se hizo uso de la función que se denomina `gaussian_geometric_shift`. Esta función toma como entrada una nube de punto y un nivel de intensidad. La salida es una nube de puntos que, a cada punto, se le ha aplicado un desplazamiento geométrico, cuyo valor viene sacado de una distribución gaussiana de media 0 y desviación típica basada

en el nivel de intensidad. Ese nivel de intensidad es un porcentaje de la caja que recubre la nube de puntos, en inglés *bounding box*. Los valores utilizados son: 0.15 %, 0.2 %, 0.25 %, 0.30 %, 0.35 %, 0.4 % y 0.5 % del *bounding box*.

### **Compresión *Octree***

En la carpeta *Distort/octree/* tenemos la implementación en C++ de esta distorsión, en concreto en *point\_cloud\_compression.cpp*. Se facilita un *CMakeLists.txt* para la generación del ejecutable con el comando *cmake*. Recibe de entrada la ruta a la nube de puntos de referencia, la resolución de compresión *octree* y el directorio de salida. La resolución se refiere al tamaño de los véxeles más pequeños en el nivel más bajo del *octree*. Cuanto más pequeña sea la resolución, mayor será la precisión en la representación de los detalles espaciales. La profundidad del *octree* depende tanto de la resolución como de la dimensión espacial de la nube de puntos, ya que determina cuántos niveles de subdivisión serán necesarios para cubrir toda el área de la nube de puntos con la resolución especificada (para más detalles repasar Sección 2.3). Se facilita también la entrada de dos parámetros adicionales para obtener las estadísticas de compresión y otro para visualizar el resultado final del decodificador. Las resoluciones son: 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 y 1.0.

### **Submuestreo aleatorio**

Esta distorsión también simula pérdida de datos en momentos de transmisión o generación de la nube de puntos. Incluso se podría considerar una forma de compresión. El método es trivial, dado un nivel de intensidad en el intervalo 0–1, que representa el porcentaje de reducción, se procede a elegir de forma aleatoria puntos a ser eliminados hasta alcanzar ese porcentaje. Los valores de reducción son: 10 %, 20 %, 30 %, 40 %, 50 %, 60 % y 70 %.

### **Rotación y Movimiento Local**

Esta distorsión simula el movimiento del paciente durante el examen médico. Para ello hemos elegido de forma aleatoria una región local de la nube de punto, cuyo tamaño corresponde al 20 % del lado más grande del *bounding box*, y le hemos aplicado un desplazamiento geométrico que equivale al 1 % del lado más largo. Los niveles de intensidad en este caso se refieren a cuántas veces se repiten el proceso de seleccionado y desplazamiento local. La rotación es simplemente una extensión de la anterior, reflejando otro tipo de movimiento, donde la selección se rota 15 grados sobre el eje X. Se aplican niveles del 1 al 7 en intervalos de 1.

## Detalles técnicos de la experimentación

La estimación de calidad del modelo DL se puede realizar invocando al script `train.py` de la carpeta `VQA_PC`, el cual recibe múltiples parámetros de entrada: se define el modelo a utilizar, se define el método de fusión de características, el ratio de aprendizaje base, la frecuencia en la que decrece, dónde están los datos, etc. Ese script nos provee las métricas resultantes de haber realizado validación cruzada, ver Sección 5.1.1, para un conjunto de datos con los parámetros establecidos. Para realizar una prueba desde un modelo pre-entrenado disponemos del script `test.py`, al igual que el anterior recibe parámetros similares de entrada. Si se prefiere, se pone a disposición un cuaderno de jupyter con la implementación bajo la librería `fastai` [111]. Mientras que para el modelo ML, disponemos de los scripts de extracción de características y los scripts necesario para la evaluación del modelo sobre SJTU[11] o WPC[87, 88] en la carpeta `NR3DQA`.

Para la ejecución se utilizaron dos sistemas distintos. En las pruebas de alta carga de CPU, como la generación de las distorsiones y las proyecciones, se utilizó un ordenador portátil ASUS FX505DY con una CPU AMD Ryzen 5 3550H, 16 GB de RAM DDR4 y una AMD Radeon RX560X que posee 4GB de VRAM. Ya que, en contra del Intel Xeon CPU E5-2699 2.2GHZ de 2 vCPUs (hebras virtuales), nos permite utilizar hasta 4 núcleos, para un total de 8 hebras en paralelo. Sin embargo, dado la necesidad de una gran cantidad VRAM, aproximadamente 13GB, se utilizó los servicios de Colab para la ejecución del entrenamiento del modelo. En este, disponemos de una NVIDIA Tesla P100 con 16 GB de VRAM.

## Fórmulas Adicionales

La función logística-4 se puede definir como la Ecuación (1):

$$Q = \frac{\beta_1 - \beta_2}{1 + e^{-\frac{Q_s - \beta_3}{|\beta_4|}}} + \beta_2 \quad (1)$$

Mientras que la función cúbica se define como la Ecuación (2):

$$Q = aQ_s^3 + bQ_s^2 + cQ_s + d \quad (2)$$

Donde  $\beta_i$ ,  $a$ ,  $b$ ,  $c$  y  $d$  son parámetros a aprender,  $Q$  es el valor normalizado y  $Q_s$  es el valor predicho.