



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: aprendizaje automático, aprendizaje profundo, visión por computador, estimación de calidad, nubes de punto, imágenes médicas.

Resumen

En el campo biomédico, la visualización y análisis de estructuras 3D, habitualmente nubes de puntos, desempeña un papel fundamental para el diagnóstico y investigación. Sin embargo, la calidad de estas representaciones pueden variar debido a diversos factores, como la adquisición, procesamiento y reconstrucción. Para mejorar cada uno de estos pasos es necesario una manera de cuantificar las distorsiones que puedan surgir. Ese problema se conoce como la estimación de calidad de nubes de puntos, un campo de especial y reciente importancia. No obstante, es un campo todavía sin explorar en el ámbito biomédico. Para estimar la calidad de representaciones 2D o 3D, generalmente se extraen características de la imagen original y la distorsionada para su posterior comparación. Estas características provienen del análisis de estructuras, colores y del conocimiento del sistema visual humano. No obstante, muchas veces no disponemos de la información de referencia, como es el caso en la medicina, sino solo la versión distorsionada.

Este TFG trata del desarrollo de un sistema capaz de estimar la calidad de una representación 3D biomédica sin referencia. La propuesta presenta la posibilidad de adaptación de modelos de estimación de calidad de nubes de puntos genéricas al ámbito médico haciendo uso de un meta-modelo de aprendizaje profundo para procesar proyecciones 2D multi-vista y un video del objeto 3D rotando. De esta forma extraemos características tanto estáticas como dinámicas. Para los experimentos se utilizaron diversos conjuntos de datos públicos de nubes de puntos genéricas, debido a que no existen en el caso biomédico, para la validación entre modelos y se propone un conjunto de datos médicos sintéticos generados a partir de datos privados. Aún siendo un conjunto de datos muy reducido, se logró una correlación entre los valores predecidos y la estimación real de calidad del 75 % mostrando el gran potencial que posee esta línea de investigación.

Quality Assessment of 3D medical images through machine learning

Brian Sena Simons

Keywords: machine learning, deep learning, computer vision, quality assessment, point cloud, medical images.

Abstract

In the biomedical field, the visualization and analysis of 3D structures, typically point clouds, play a fundamental role in diagnosis and research. However, the quality of these representations can vary due to various factors, such as acquisition, processing, and reconstruction. To improve each of these steps, a way to quantify the distortions that may arise is necessary. This problem is known as point cloud quality estimation, a field of special and recent importance. However, it is still an unexplored field in the biomedical domain. To estimate the quality of 2D or 3D representations, features are generally extracted from both the original and distorted images for subsequent comparison. These features come from the analysis of structures, colors, and knowledge of the human visual system. However, in many cases, we do not have the reference information, as is the case in medicine, but only the distorted version.

This Bachelor's thesis deals with the development of a system capable of estimating the quality of a biomedical 3D representation without a reference. The proposal presents the possibility of adapting generic point cloud quality estimation models to the medical field by using a meta-model of deep learning to process multi-view 2D projections and a video of the rotating 3D object. In this way, we extract both static and dynamic characteristics. For the experiments, various public datasets of generic point clouds were used, as they do not exist in the biomedical case, for model validation. Additionally, a set of synthetic medical data generated from private data is proposed. Despite being a very limited dataset, a correlation of 75% was achieved between the predicted values and the actual quality estimation, demonstrating the great potential of this line of research.

Yo, **Brian Sena Simons**, alumno de la titulación TITULACIÓN de la **Es-cuela Técnica Superior de Ingenierías Informática y de Telecomu-nicación**, 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 X de mes de 201 .

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 X de mes de 201 .

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

1. Introducción	7
1.1. Definición del Problema	7
1.2. Motivación	9
1.3. Objetivos	11
1.4. Planificación del proyecto	11
2. Fundamentos Teóricos	15
2.1. Image Quality Assessment IQA	15
2.2. Aprendizaje Automático y Profundo	18
2.2.1. Aprendizaje Automático	18
2.2.2. Aprendizaje Profundo	19
2.2.3. Ensemble o Conjunto <i>Deep Learning</i>	25
2.3. Imágenes médicas y distorsiones	26
3. Estado del Arte	29
3.1. Estado del arte de FR-IQA	30
3.2. Estado del arte de NR-IQA	31
3.3. Estado del arte de IQA en imágenes médicas	32
3.4. Estado del arte de FR-PCQA	33
3.5. Estado del arte de NR-PCQA	34
4. Materiales y Métodos	37
4.1. Materiales	37
4.1.1. Datos públicos	37
4.1.2. Conjunto de datos médicos	40
4.2. Métodos	40
4.2.1. Zhang et al[26]	41
4.2.2. VQA-PC	42
5. Implementación y Experimentos	45
5.1. Detalles Técnicos de Implementación	45
5.1.1. Obtención de los modelos 3D	46
5.1.2. Distorsiones	47
5.1.3. Métricas	48

5.1.4. Pseudo-MOS	50
5.1.5. Entrenamiento de los modelos ML y DL	51
5.2. Experimentos	51
5.2.1. Protocolo de validación experimental	51
5.2.2. Método con SVM	52
5.2.3. Experimentos Preliminares DL	54
5.2.4. Propuestas de mejoras	56
6. Conclusiones y Trabajos Futuros	59
7. Bibliografía	63

Índice de figuras

1.1.	Ejemplo de visualización de un directorio DICOM.	10
2.1.	Visualización del problema de la métrica <i>Minkowski</i>	17
2.2.	Visualización del hiperplano MSE de imágenes distorsionadas.	17
2.3.	Ejemplo de aprendizaje supervisado.	18
2.4.	Ejemplo de aprendizaje no supervisado.	19
2.5.	Ejemplo gráfico de una red neuronal[40-42].	20
2.6.	Ejemplo de red convolucional para imágenes médicas	21
2.7.	Representación visual de la operación de convolución	22
2.8.	Ejemplo de operación de <i>max-pooling</i> con <i>stride</i> a 2.	23
2.9.	Red convolucional espaciotemporal <i>SlowFast</i>	24
2.10.	Ejemplo extraído de[49] sobre voxelización.	24
2.11.	Representación de métodos de <i>ensemble</i> extraído de[50]. . . .	25
2.12.	Ejemplo de tomografía computarizada y modelo 3D[51]. . . .	26
2.13.	Ejemplo de artefactos sobre imágenes DICOM.	27
2.14.	Ejemplo de distorsiones generadas sobre clavículas.	28
3.1.	Crecimiento de interés en el campo según <i>Scopus</i>	29
4.1.	Ejemplo de conjuntos de datos SJTU[88].	37
4.2.	Ejemplo de conjunto de datos WPC[89, 90].	38
4.3.	Ejemplo de conjunto de datos LS-SJTU-PCQA[19].	39
4.4.	Ejemplo de nuestras imágenes médicas	40
4.5.	Ejemplo de distorsiones que se presentan según la perspectiva	43
4.6.	Ejemplo de las rotaciones que utiliza el modelo VQA-PC[28].	43
4.7.	Ejemplo detallado de las etapas del método de VQA-PC . .	44
5.1.	Ejemplo de uso de K-fold para la búsqueda de hiperparámetros.	52
5.2.	Curvas de aprendizaje del test preliminar.	56

Índice de tablas

1.1.	Planificación temporal inicial del proyecto.	12
1.2.	Planificación resultante del proyecto.	12
1.3.	Total de horas y días trabajados.	13
1.4.	Estimación final de coste del proyecto.	13
3.1.	Tablas estado del arte FR-IQA	31
3.2.	Tablas estado del arte NR-IQA	31
3.3.	Estado del arte de modelos NR-PCQA	35
4.1.	Ejemplo de distorsiones en SJTU[88]	38
5.1.	Comparativa entre funciones de normalización.	49
5.2.	Tabla de métricas para generación de etiquetas de [19].	50
5.3.	Correlación de métricas sintéticas de [19].	50
5.4.	Resultados de prueba preliminar con SVM.	52
5.5.	Resultados de prueba preliminar con SVM.	53
5.6.	Resultado de mejores sobre el método SVM	53
5.7.	Hiperparámetros de prueba preliminar.	54
5.8.	Descripción de la arquitectura ResNet50[91].	55
5.9.	Resultados de experimento preliminar.	55

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 la información puede sufrir drásticamente 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[1].

En concreto, este TFG se centra en el estudio de la evaluación de la calidad de la imagen, en inglés *Image Quality assessment (IQA)*[2]. Es un problema fundamental en el procesamiento de imágenes y de visión por computador. Se refiere 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 las técnicas suele estar altamente conectado con el avance de los estudio del sistema de visión humano[3].

El problema de la evaluación de la calidad de la imagen se aborda mediante enfoques subjetivos y objetivos. 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*¹.

Alternativamente, los enfoques objetivos buscan desarrollar algoritmos y

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

métricas automáticas que puedan estimar la calidad de la imagen sin intervención humana. Estos enfoques se basan en características y propiedades visuales extraídas 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. Mientras que 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. 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, se emplean diversas técnicas y enfoques [1-3]. Entre ellos se incluyen métodos basados en características, modelos de percepción visual, aprendizaje automático y técnicas de procesamiento de señales [4-6]. Uno de los enfoques comunes es 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[2, 3]. 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 [2, 6]. Buscan emular la forma en que los humanos responden a las imágenes en términos de su calidad visual[7, 8].

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.

Las aplicaciones comunes de los algoritmos de estimación de calidad suele ser la comparativa entre algoritmos de compresión, ya que permite elegir aquellos con menor pérdida de información, la generación de mapas de calidad²,

²Calidad perceptual en diferentes regiones de la imagen. Resaltan áreas que pueden

permitiendo el estudio de métodos de reducción de ruido, y la determinación de la calidad del servicio de transmisión o *quality-of-service (QoS)*, ya que permiten evaluar los errores de transmisión[9, 10]. 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[11].

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 debido al aumento de complejidad. Además, en el ámbito médico, la naturaleza de estas imágenes y las distorsiones que pueden presentar, ver Figura 2.13, implican una disminución en la precisión de los modelos cuando se aplican directamente sobre ellas[1]. Por lo tanto, el número de métodos médicos existentes se reduce, con ninguno, al momento de escritura, aplicado a imágenes médicas 3D.

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.

1.2. Motivación

En el caso del ámbito biomédico, dados los rápidos avances en las últimas décadas de las técnicas no invasivas y la gran cantidad de fabricantes de equipamientos, nació el estándar *DICOM* [12] en 1995 con el objeto de hacer que el intercambio de imágenes médicas se realice de forma fácil, segura y con alta calidad. Permitiendo la integración con diversos sistemas, almacenar información extra en forma de metadatos y anotaciones, así como segmentaciones que permitan la reconstrucción 3D de diferentes regiones anatómicas.

Cada vez más frecuentemente se emplean volúmenes tridimensionales, como tomografías computarizadas (TC) 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, ver Figura 1.1. Esto brinda a los médicos una comprensión más completa de la anatomía, les ayuda a identificar con mayor precisión lesiones, enfermedades o anomalías, así como también facilita la planificación quirúrgica, entre otros[13-15].

A pesar de ello, las distorsiones, que son una ocurrencia común en las imágenes cotidianas, están muy presentes en las imágenes médicas [17]. Prevalecen las distorsiones de contraste, ruido y difuminado³, que se detallarán en la

tener degradación visual o distorsiones.

³Con ruido nos referimos a pequeñas fluctuaciones no deseadas en los colores de los píxeles debido a interferencias de todo tipo. Difuminado se refiere a la pérdida de detalles

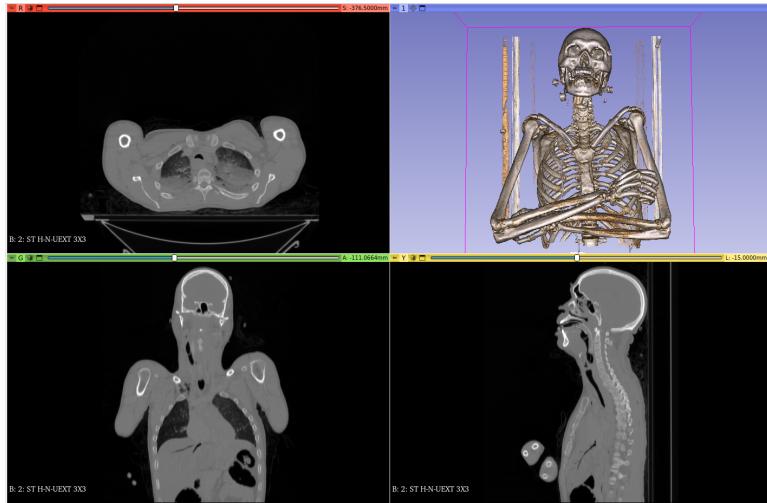


Figura 1.1: Ejemplo de visualización de un directorio *DICOM*. Se pueden observar las proyecciones axial, arriba izquierda, coronal, abajo izquierda, y sagital, abajo derecha. Además de una renderización volumétrica de los huesos, arriba derecha. Para ello se ha utilizado *Slicer3D* [16]

Sección 2.3. Estas a su vez, podrían afectar al volumen 3D que se puede generar a partir las imágenes médicas.

En [18] se estudiaron las razones por las que se suelen rechazar las radiografías, la relación con la calidad de la imagen y el valor del 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 o falsos negativos. Sin mencionar los elevados costes que supone realizar nuevas pruebas para conciliar las anteriores.

Resolver este problema, o dar pasos hacia adelante, formulando una medida de calidad puede conllevar a la mejora de los dispositivos médicos, de los algoritmos de comprensión, almacenado y transmisión de información (tanto a nivel médico como a cualquier aplicación con datos tridimensionales). Resultando en reducción de costes (menos pruebas), de tiempo de consultas y mejorando la calidad del diagnóstico médico.

en los bordes, como una pérdida de enfoque.

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 del estado del arte para la estimación de calidad de imágenes *3D*, así como de la calidad de imágenes médicas en particular.
2. Estudiar las distorsiones más comunes en el ámbito de las imágenes , en general, y analizar los patrones de distorsión que afectan la calidad de las imágenes biomédicas.
3. Analizar 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 analizados. Para ello, será necesario estudiar diferentes estrategias y métricas de evaluación objetivas[19].
5. Realizar un estudio experimental que permita validar los enfoques propuestos y extraer conclusiones sobre su aplicabilidad al problema.

1.4. Planificación del proyecto

Al planificar el proyecto, es fundamental tener en cuenta 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, lo cual permite abordar su desarrollo a través de un enfoque de ciclo de vida en cascada [20]. 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*⁴, teniendo en cuenta un estudio previo de las distorsiones médicas.
- Diseño: Consistió 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 nuevos.
- 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 (1.1) se tomó como referencia, pero no de forma estricta. Ya que se tuvo en cuenta que el autor estaba realizando prácticas de empresa, tenía una asignatura y participaba de un curso de *Google* ofrecido por la universidad. Además, se esperaba que ocurriera retrasos sobre todo en la implementación, como se puede ver en (1.2), dado la novedad de la propuesta y dificultad del problema. Un ejemplo fue a la hora de simular las distorsiones médicas, caso que fue algo iterativo y manual.

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.

⁴*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)

Fecha inicio	21/02/2022
Fecha fin	25/07/2022
Duración	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€
Total	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*. Los primeros, son problemas dónde 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[21], 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, 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[3], ver Figuras 2.1 y 2.2.

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

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[3].

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 naturales de escena (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[SpatialDomainForJPEG]

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[22-24].

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^1 , 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[25-27]. Un enfoque común es la evaluación de calidad de una nube de puntos tridimensional mediante proyecciones 2D desde diferentes perspectivas[28-30]. 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.

²La distancia de *Minkowski* es una métrica vectorial que puede considerarse como una generalización tanto de la distancia euclídea como de la distancia de Manhattan .

³*Mean squared error* o error cuadrático medio es una métrica de distancia que se calcula como la media de la suma de las diferencias al cuadrado.

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

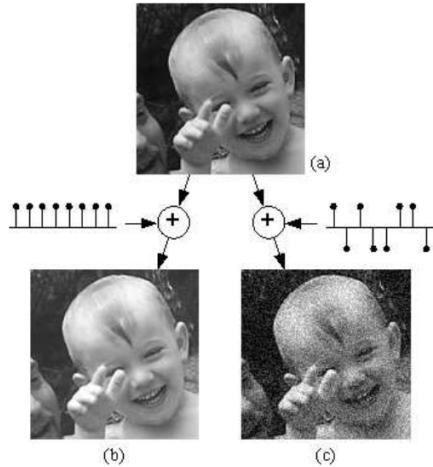


Figura 2.1: En este ejemplo, extraído de[2], vemos que sumar una constante positiva a una imagen de referencia (a) produce la imagen (b) que contiene la misma distancia $Minkowski^2$ que (c), imagen fabricada por la misma constante pero permutando signo de forma aleatoria, resultando que la percepción final es que la imagen (c) es peor que la imagen (b).

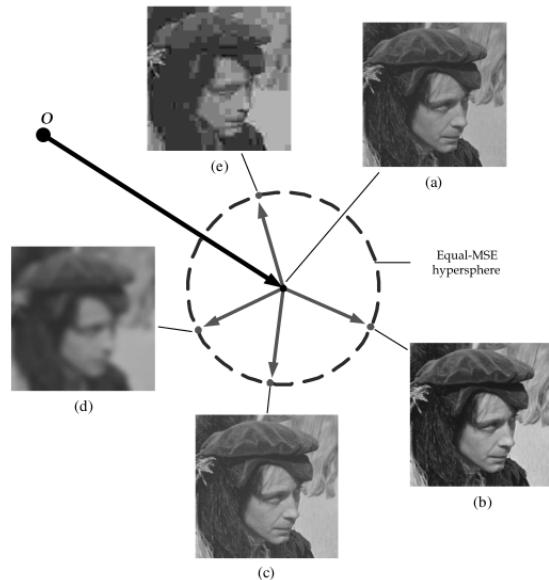


Figura 2.2: En este ejemplo, extraído de[3], la misma imagen distorsionada de distintas maneras resulta en el mismo valor $MSE^3=181$. Siendo evidente que algunas distorsiones producen efectos visuales más marcados que otras.

2.2. Aprendizaje Automático y Profundo

2.2.1. Aprendizaje Automático

El aprendizaje automático[31] o *Machine Learning (ML)* es una de las ramas que compone lo que definimos como la inteligencia artificial (*IA*).

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[32]. 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[33].

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, ver Figuras 2.3 y 2.4.

En general se suelen aplicar las técnicas de *ML* sobre grandes conjuntos de datos sobre los cuales deseamos detectar los patrones subyacentes[34].

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.

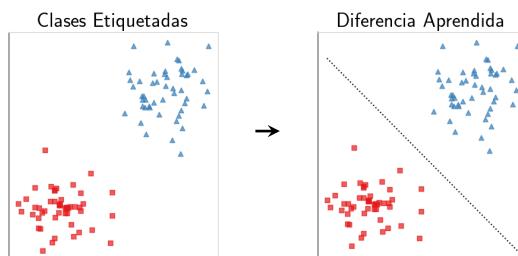


Figura 2.3: Ejemplo de aprendizaje supervisado. Vemos como a partir de un conjunto de clases etiquetadas aprendemos un hiperplano que las separa.

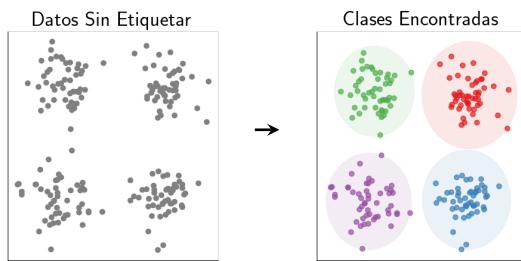


Figura 2.4: Ejemplo de aprendizaje no supervisado. Dado un conjunto de puntos aprendemos un conjunto de clases a partir de los patrones.

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[35-37]. 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)[38, 39].

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 anomalías 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.

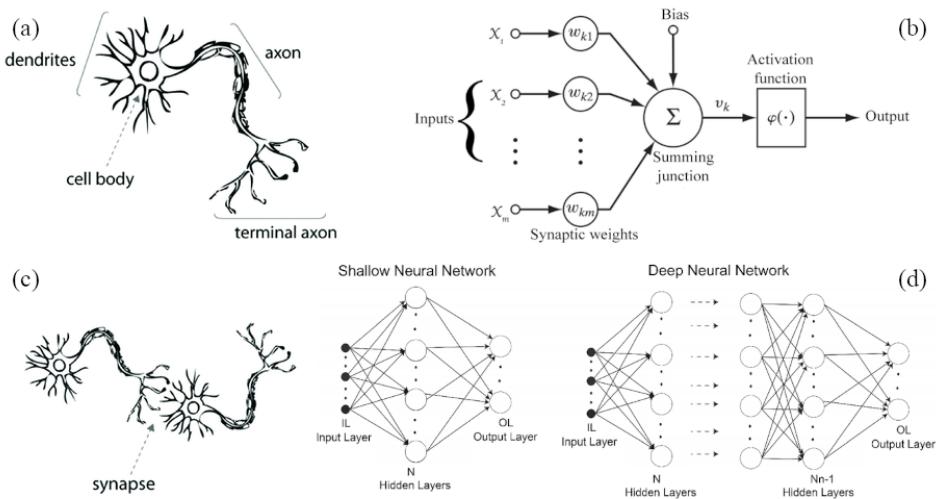


Figura 2.5: Ejemplo gráfico de una red neuronal[40-42]. En (a) vemos una neurona con su representación artificial simplificada (b). En (c) vemos una conexión entre neuronas y en (d) la representación de diferentes profundidades de conexiones artificiales.

Redes Convolucionales

Las redes convolucionales o *convolutional neural network* (CNN)[43, 44] son un tipo de arquitectura de redes neuronales diseñadas específicamente para el 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 2.6 podemos ver un ejemplo de esquema jerárquico de extracción de características para el diagnóstico médico a partir de una radiografía. Con ese ejemplo se puede observar las diferencias con respecto a las ANN, posee dos capas adicionales: capas convolucionales y capas de *pooling*.

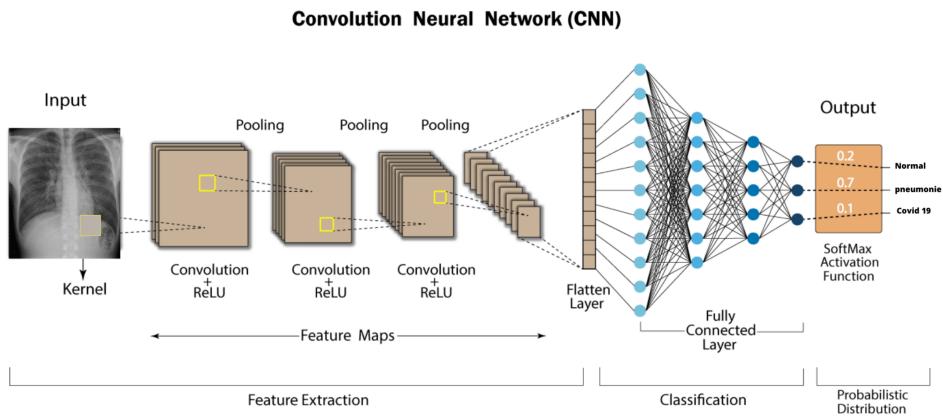


Figura 2.6: Ejemplo extraído de [45] del proceso de convolución sobre una radiografía pulmonar para la detección de enfermedades.

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 de entrada para generar el mapa de características completo. En la Figura 2.7 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 2.7 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.

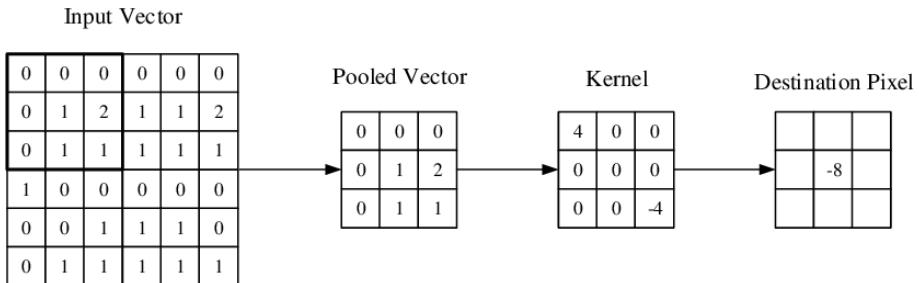


Figura 2.7: Representación visual de la operación de convolución sobre una imagen, extraída de [46].

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.

En la Figura 2.8 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.

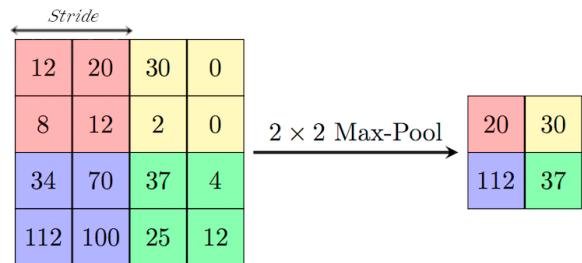


Figura 2.8: Ejemplo de operación de *max-pooling* con *stride* a 2.

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 CNN's 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*[47]. 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 (Figura 2.9).

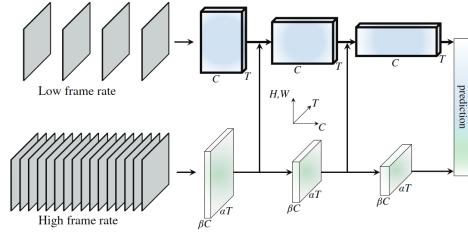


Figura 2.9: Ejemplo extraído de[47] para ilustrar la distinción entre la extracción espacial del camino “*slow path*” y de movimiento con el “*fast path*”.

Aplicadas a nubes de puntos

De forma similar al caso de los vídeos, actualmente existen modelos de 3D CNN's capaces de procesar nubes de puntos al añadir una dimensión más que representa la profundidad de los puntos. Sin embargo, la complejidad de diseño y tiempo de cómputo para estos modelos 3D crece enormemente. Ya 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*[48], habitualmente se intenta estructurar la información de las nubes de puntos mediante lo que denominamos voxéles[49]. 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 voxels y asignar valores a cada voxel 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.

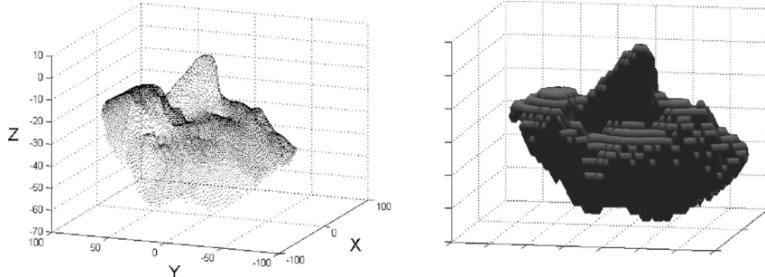


Figura 2.10: Ejemplo extraído de[49] que demuestra el resultado de transformar una nube de puntos sin estructura en una cuadrícula voxelizada.

2.2.3. Ensemble o Conjunto 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[31, 34, 50].

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 evalua el uso de un meta-modelo para la estimación de calidad de las imágenes médicas 3D sin referencia.

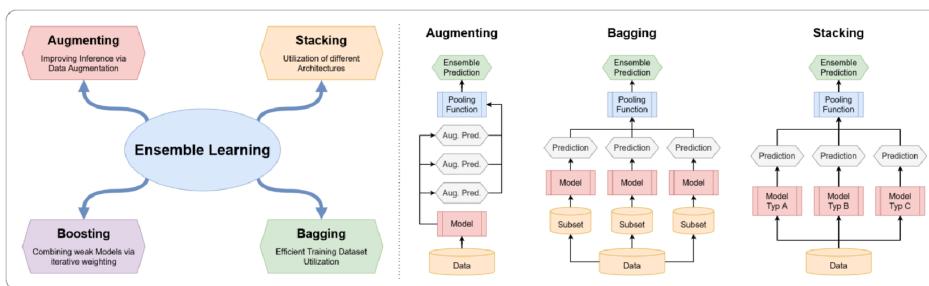


Figura 2.11: Representación de métodos de *ensemble* extraído de[50].

2.3. Imágenes médicas y distorsiones

Las tomografías computarizadas (CT) 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.

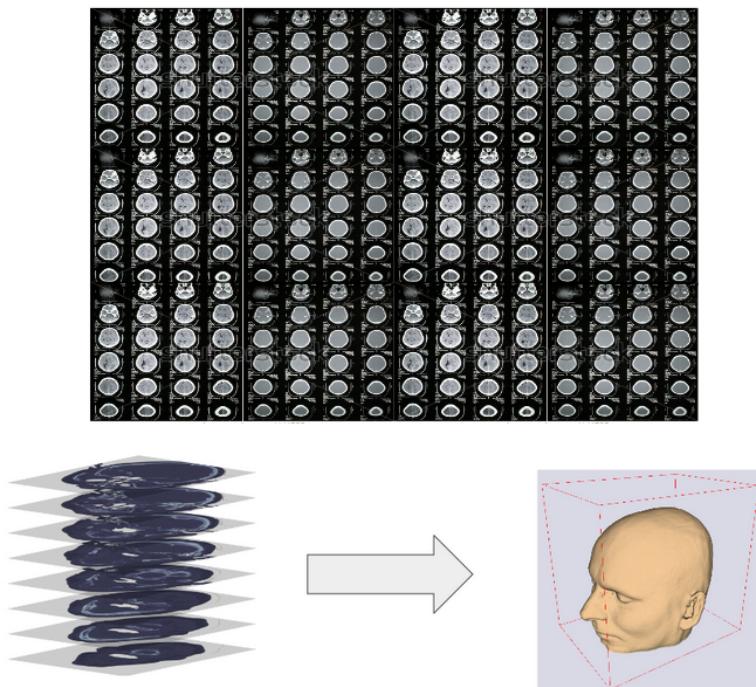


Figura 2.12: Ejemplo de tomografía computarizada y modelo 3D[51].

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 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 segmentado² 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*), ver Figura 2.12.

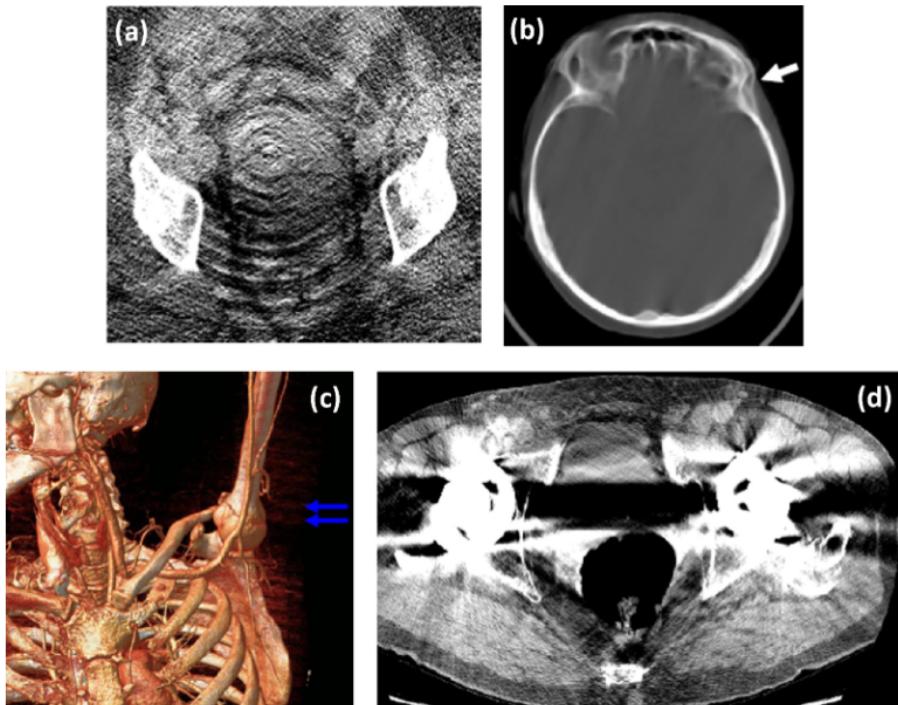


Figura 2.13: Ejemplo de artefactos sobre imágenes DICOM[52]: (a) un artefacto de anillo, (b) un difuminado provocado por movimiento, (c) artefactos producidos por interpolación helicoidal, (d) artefactos de iluminación y dispersión del haz.

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, ver Figura 2.13.

Para simularlo, partiremos de ejemplos de imágenes médicas consideradas exentas de distorsiones, con su segmentación correspondiente realizada por profesionales. A continuación, generaremos la representación volumétrica como una nube de puntos y sobre ella simularemos las distorsiones.

²La 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.

Dos de ellas serán para simular los resultados de varios algoritmos de compresión, como puede ser *octree compression*[53] 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.

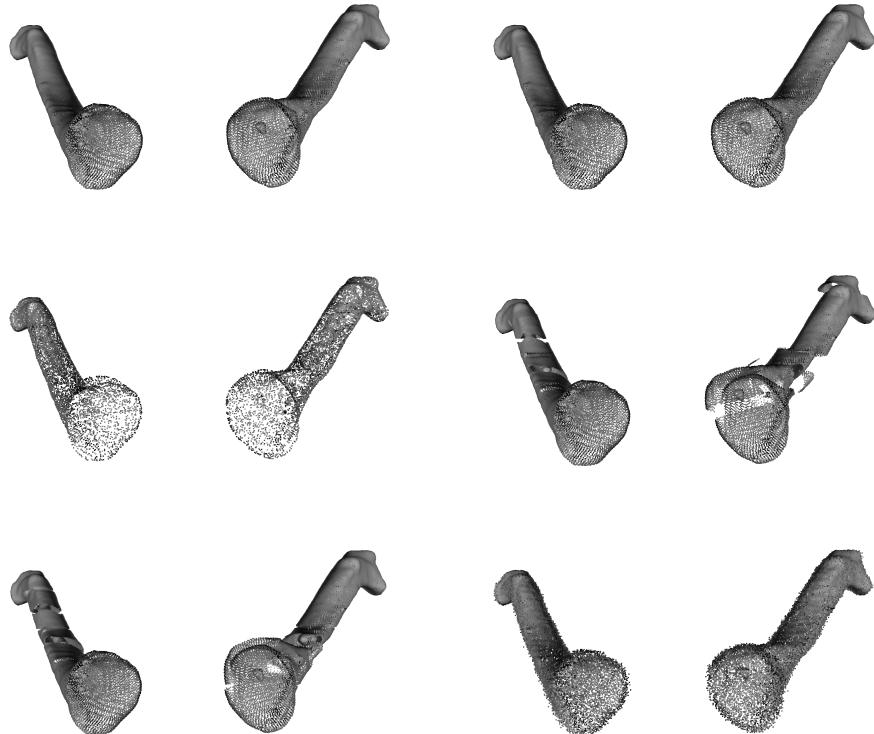


Figura 2.14: Ejemplo de distorsiones generadas sobre clavículas. Leyendo de izquierda a derecha tenemos: la imagen original, compresión *octree*, reducción de puntos por *random downsampling*, movimiento local, rotación local y ruido gaussiano.

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 inteligencia artificial en las nubes de puntos y en imágenes médicas, llegando ambas a sobreponerse a 6000 documentos a partir de 2020. Vemos que ambos incluso siguen lado a lado en número de publicaciones cuando especificamos que sean documentos relacionados con la estimación de calidad, sobreponiendo 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.

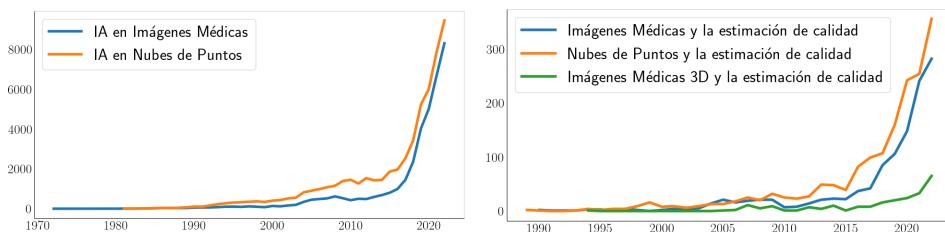


Figura 3.1: Crecimiento de interés en el campo según *Scopus*¹. A la izquierda podemos ver un incremento de publicaciones desde 1970 sobre la IA aplicada a nubes de punto, en naranja, y aplicada de forma general a imágenes médicas, en azul. A la derecha podemos ver de forma similar el crecimiento de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

¹Las búsquedas se pueden consultar en el Apéndice 7

A continuación se presenta el estado del arte. Comentaremos, brevemente, el estado actual de los métodos IQA tradicionales. A continuación, el estado de métodos en imágenes médicas 2D. Y por último, nos centraremos en el estado actual de métodos sin referencia para nubes de puntos, el campo sobre el cual trabajaremos.

3.1. Estado del arte de FR-IQA

Como ha sido mencionado anteriormente, los métodos convencionales de cuantificación del error en la evaluación de la fidelidad de imágenes, como el error cuadrático medio, la relación señal-ruido (SNR, por sus siglas en inglés) y el pico de señal-ruido (PSNR, por sus siglas en inglés), no consideran el contenido de una imagen ni las características del sistema visual humano (HVS, por sus siglas en inglés). Por lo tanto, estos métodos suelen tener una débil consistencia con la percepción humana, recordar Figuras 2.1 y 2.2. Para abordar esta limitación, se han propuesto métricas inspiradas en el HVS y conscientes del contenido de la imagen, que combinan las características del HVS con algoritmos matemáticos. Por ejemplo, Visual SNR[54] que cuantifica la fidelidad visual de las imágenes distorsionadas, mientras que PSNR-HVS[55] combina el PSNR con las características del HVS al considerar la función de sensibilidad al contraste.

Wang y Bovik afirmaron que los ojos humanos obtienen información de imagen a través de tres canales: brillo, contraste y estructura[6], y desarrollaron un índice universal de calidad de imagen (UQI)[56] y una similitud estructural (SSIM, por sus siglas en inglés)[6]. Desde entonces, se han propuesto diversas variantes basadas en SSIM, como MS-SSIM[57]. Surgieron también métodos que, basados en las respuestas del HVS, introducen la saliencia visual¹ en la evaluación de la calidad de imagen (VSI)[7], ya que se ha observado que la saliencia de la imagen desempeña un papel importante.

En los últimos años, se han aplicado métodos de aprendizaje automático. Algunos ejemplos de estos métodos son MLIQM[58], que clasifica las imágenes distorsionadas en cinco escalas utilizando máquinas de soporte vectorial (SVM, por sus siglas en inglés), y MMF[4] que utiliza una fusión multiméjido motivada por la observación de que ningún método único puede ofrecer el mejor rendimiento en todos los tipos de distorsión. Además, se ha llegado a utilizar incluso CNN's para resolver el problema, ya que son capaces de aprender características y regresión a partir de los datos brutos de las imágenes. Un ejemplo de esto es WaDIQaM-FR[59], que es una medida de calidad de imagen profunda ponderada y basada en datos sin depender de características predefinidas o conocimiento previo.

¹Cualidad estética de la forma de un objeto o una configuración que destaca.

Type	Metric	LIVE			CSIQ			TID2008			TID2013		
		SRCC	PLCC	RMSE	SRCC	PLCC	RMSE	SRCC	PLCC	RMSE	SRCC	PLCC	RMSE
FR	VSNR[54]	0.927	0.923	10.506	0.811	0.800	0.158	0.705	0.682	0.982	0.681	0.740	0.839
	PSNRHVS[55]	0.919	0.903	12.540	0.830	0.804	0.156	0.594	0.608	1.065	0.654	0.430	0.704
	UQI[56]	0.894	0.899	11.982	0.810	0.831	0.146	0.585	0.664	1.003	-	-	-
	SSIM[6]	0.948	0.845	8.946	0.876	0.861	0.133	0.775	0.773	0.851	0.742	0.790	0.761
	MS-SSIM[57]	0.951	0.949	8.169	0.913	0.899	0.115	0.854	0.845	0.717	0.786	0.833	0.686
	VSI[7]	0.952	0.948	8.682	0.942	0.928	0.098	0.898	0.876	0.647	0.897	0.900	0.540
	DSS[5]	0.962	0.931	9.961	0.961	0.957	0.076	0.873	0.877	0.644	0.792	0.848	0.658
	CD-MMF[4]	0.981	0.980	5.413	0.967	0.9614	0.067	0.942	0.9414	0.429	-	-	-
	WaDIQaM-FR[59]	0.970	0.980	-	-	-	-	-	-	-	0.940	0.946	-

Tabla 3.1: Tabla extraída de [60], donde vemos el progreso de las métricas FR conforme avanza los conocimientos del HVS, ML y DL.

3.2. Estado del arte de NR-IQA

Al principio, surgieron métricas para la estimación de la calidad de las imágenes que hubiesen sufrido algún tipo de distorsión específica, como imágenes borrosas[61], imágenes comprimidas en JPEG[62], imágenes con artefactos de bloque[63] y imágenes con cambios de contraste[64]. 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 estadísticas de escenas naturales (NSS, por sus siglas en inglés). Un ejemplo conocido es el evaluador sin referencia BRISQUE[65], 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[66] 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[67], y posteriormente se han realizado muchos esfuerzos para mejorar su rendimiento mediante el diseño de estructuras convolucionales más profundas. En concreto, DIQaM-NR[59], que mejora frente a redes menos profundas.

Type	Metric	SRCC	LIVE PLCC	RMSE
NR	BRISQUE [65]	0.940	0.942	-
	LGP [66]	0.957	0.954	7.901
	IQA-CNN [67]	0.956	0.953	-
	DIQaM-NR [59]	0.960	0.972	-
	Hallucinated-IQA [23]	0.982	0.982	-

Tabla 3.2: Tabla extraída de [60], donde vemos el progreso de las métricas NR conforme nos alejamos de características basadas en el conocimiento humano y de NSS utilizando ML y DL.

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[52]. Por lo tanto, el desarrollo de métodos de evaluación de calidad de imágenes sin referencia es de particular importancia en este campo[23, 59, 65-67]. 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[52]. 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 (HVS), Bhateja et al.[68] 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.[69] introdujeron una técnica semi-supervisada dedicada a la evaluación de calidad de imágenes de MRI cerebral fetal utilizando un método de profesor promedio² y la consistencia de las regiones de interés. Además, Liu et al.[70] 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[71]. Para abordar el problema de desperdiciar información espacial 3D potencialmente importante, se creó el enfoque HyS-net[72], basado en una hiper-red y es capaz de auto adaptación[72]. Así como fue expuesto anteriormente, no es posible simplemente implementar métodos IQA, es por ello que Chow y Rajagopal [73] propusieron un enfoque más reciente que adapta el evaluador de calidad de imagen más famoso BRISQUE[65].

Hay que tener en cuenta que la explicabilidad es un aspecto crucial para evaluar la fiabilidad de los sistemas de diagnóstico automático, especialmente en aplicaciones de aprendizaje profundo tipo caja negra. Dado que un clasificador profundo nos proporciona solo el valor inferido para un escaneo médico dado, es imperativo descubrir las señales que pueden ayudar a asegurar que la decisión se haya tomado en función de conjuntos de características relacionales.

²Método de aprendizaje semi-supervisado que utiliza dos modelos, uno considerado estudiante y otro profesor, que trabajan conjuntamente para aprender a resolver la tarea.

3.4. Estado del arte de FR-PCQA

Las métricas más populares para la estimación objetiva de la calidad de modelos 3D basada en puntos son *point-to-point* (Po2Po)[74] y *point-to-plane* (Po2Pl)[75]. En la primera, para cada punto del objeto distorsionado se obtiene su vecino más cercano en la versión de referencia y se calcula alguna métrica de distancia como las discutidas anteriormente (MSE, Minkowski, Hausdorff...). La principal desventaja de estos modelos es que no consideran los objetos como superficies, aparte de lo comentado la Sección 2.1.

Para solventar este problema se formuló el segundo método por Tian et al. que modela la superficie en cada punto como un plano. Ese plano es perpendicular a la normal en cada punto, que se calcula en base a la información de su vecindario. Se propuso también otra métrica como *Plane-to-Plane* (Pl2Pl)[76], que mide la similitud entre superficies asociadas a las nubes de puntos. Dentro de este mismo ámbito surge también *Point-to-Surface*[77], que mide la distancia de cada punto de la versión distorsionado respecto a su superficie correspondiente en la nube de referencia. Luego para poder tener en cuenta los colores, surgieron métricas como Po2Po PSNR donde la diferencia ya no es respecto a la posición del punto sino que al color.

De la información del vecindario de los puntos, gran cantidad de información geométrica puede ser extraída para investigar en profundidad la similitud entre las nubes de puntos. Surgieron métodos basados en la extracción de características donde la mayoría considera tanto información geométrica como los atributos lumínicos. Un ejemplo sería PointSSIM[78], 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[6]. Experimentando con combinaciones de 3 medidas geométricas y 5 comparaciones de color para encontrar el mejor vector características resultó en la métrica PCQM[79].

Otros, velaron por estudio la energía potencial en las nubes de puntos y las diferencias que emergen en presencia de distorsiones (MPED)[80]. Por otro lado, utilizando transformaciones de datos, se construyeron grafos que representan las nubes de puntos, tanto la de referencia como la distorsionada, para producir métricas de similitud como GraphSIM[81]. De igual forma 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. Se realizó incluso un estudio sobre el impacto del número de proyecciones 2D de distintas perspectivas en el rendimiento de las métricas de calidad[30, 82].

Conocer algunas de estas métricas del estado del arte de los métodos con referencia nos servirá a la hora de generar el conjunto de datos sintéticos para la evaluación del modelo sobre imágenes médicas 3D, ver Sección 5.1.2.

3.5. Estado del arte de NR-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.

Al igual que en las primeras aproximaciones de los métodos NR-IQA, existen algoritmos de estimación específicos, como el propuesto por Liu et al[83], 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.[84] utilizó distancias geométricas, curvaturas medios y niveles de color, en escala gris. Zhang et al.[26] siguiendo la misma línea extrajo características de los vectores y valores singulares de cada punto, además de utilizar características lumínicas.

El siguiente paso, al igual que en FR, se empezó a investigar la posibilidad de utilizar proyecciones 2D. En PQA-Net[85] 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[30] deciden utilizar métodos IQA aplicadas a las multi-proyecciones. Extendiendo los trabajos anteriores, tenemos VQA-PC[28] que trata las multi-proyecciones como vídeo, pudiendo así utilizar información espacial, imágenes en posiciones específicas, y información de consistencia temporal de la nube de puntos rotando.

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[19] modifican el esqueleto de PointNet[48] 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 de nubes de puntos sintético, en el momento de escritura, con su estimación de calidad. Otro método que trabaja directamente sobre la nube de puntos es SGR[86], que extraen regiones locales de la nube de puntos y analizan la calidad de los parches. Recientemente, ha salido el modelo MM-PCQA[29] que utilizan tanto información directa de la nube de puntos como información de las proyecciones. Incluso, hay métodos que han optado por utilizar un *ensemble*[87] de modelos pre-entrenado en 2D y obtuvieron muy buenos resultados.

MODELO	STJU-PCQA		WPC	
	PLCC	SRCC	PLCC	SRCC
MM-PCQA[29]	0.92	0.91	0.83	0.83
SGR[86]	0.89	0.84	-	-
VQA-PC[28]	0.8635	0.8509	0.7976	0.7968
ResSCNN[19]	0.86	0.81	0.72	0.75
GPA-Net[27]	0.806	0.78	-	-
Zhang et al.[26]	0.7382	0.7144	0.6514	0.6479
IT-PCQA [30]	0.58	0.63	0.55	0.54

Tabla 3.3: Resumen del estado del arte de modelos NR-PCQA en dos datasets muy conocidos: SJTU[88] y WPC[89, 90].

Aunque en la Figura 3.1 parece haber documentos para la estimación de calidad de imágenes médicas 3D, solamente encontré uno respecto a la calidad de segmentaciones. Los demás, imagino que son apenas documentos que contienen la palabra calidad o estimación. Así que no sé exactamente si dedicar una sección a decir que no he encontrado nada o qué decir aquí.

Capítulo 4

Materiales y Métodos

4.1. Materiales

4.1.1. Datos públicos

Para este TFG se ha utilizado diversos conjuntos de datos públicos para la evaluación y elección de los modelos anteriormente descritos. De entre ellos están SJTU[88], WPC[89, 90] y LS-SJTU-PCQA[19].

El primer de ellos, parte de 10 nubes de puntos de referencia, ver 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, ver 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.

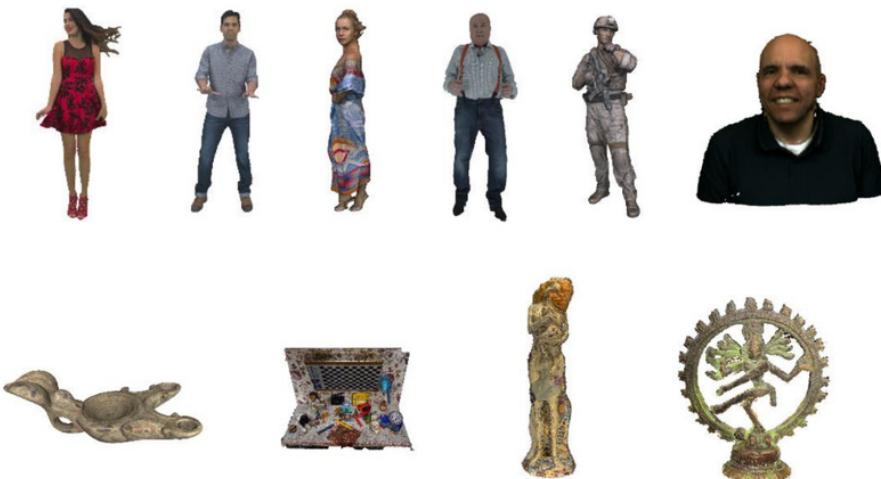


Figura 4.1: Ejemplo de conjuntos de datos SJTU[88].

Número	Tipo de Distorsión
0	OT: Compresión octree[53]
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[88].

El segundo, WPC[89, 90], 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 modelos de nubes de puntos, ver 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.

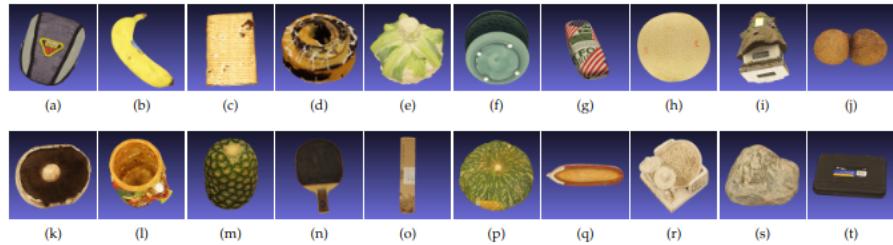


Figura 4.2: Ejemplo de conjunto de datos WPC[89, 90].

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. Además, se realizaron pruebas de ejecuciones de algunos métodos de código abierto para verificar los resultados. Sin embargo, el último es el que finalmente se utiliza para entrenar un modelo para estimar la calidad de las imágenes médicas. Esto es porque LS-SJTU-PCQA[19] 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. Ver Figura 4.3.

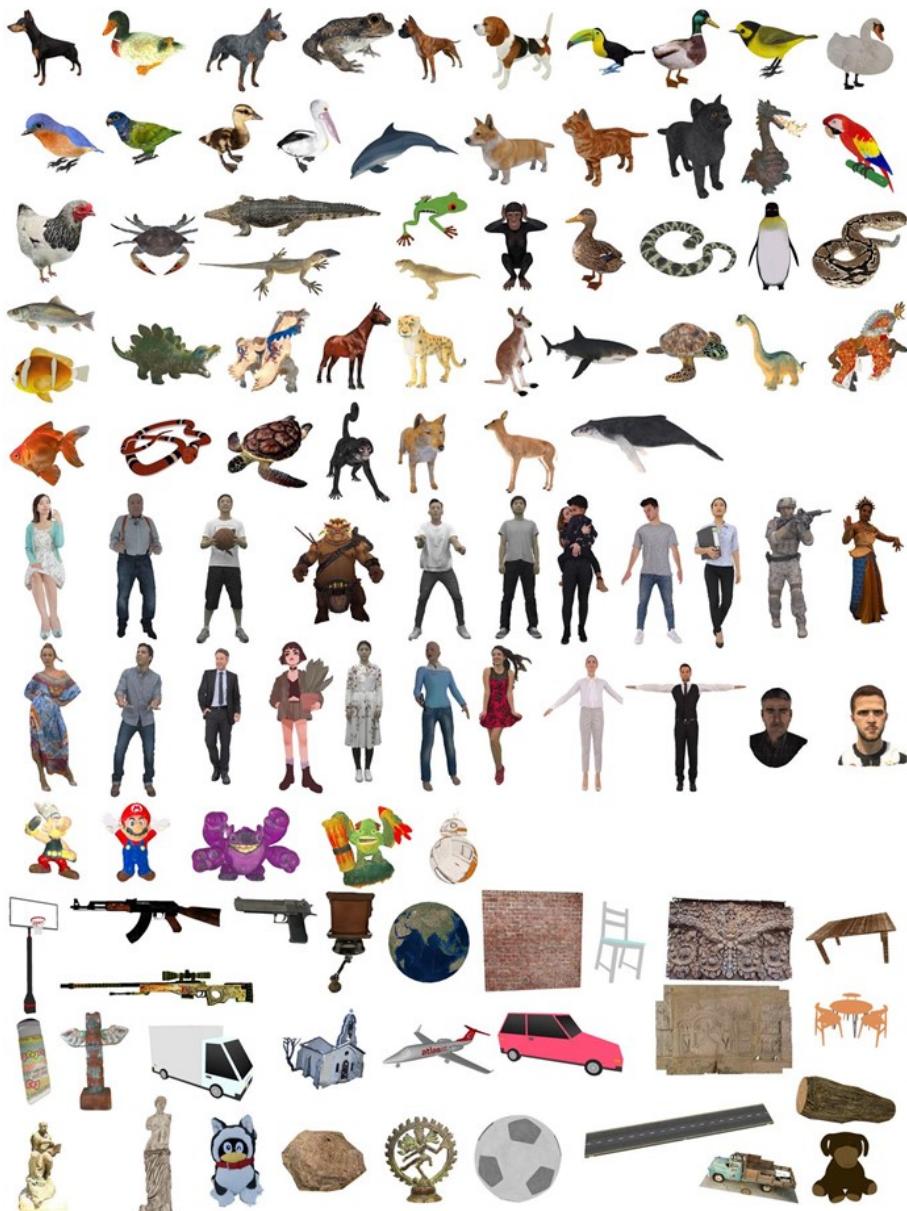


Figura 4.3: Ejemplo de conjunto de datos LS-SJTU-PCQA[19].

4.1.2. Conjunto de datos médicos

Para este TFG se tiene disponible una colección 26 directorios DICOM, de distintas partes del cuerpo, de 2 distintos individuos. De los cuales han sido segmentados las clavículas, el seno frontal y el seno Maxilar. A parte, disponemos de volúmenes del cráneo de otros 3 individuos, una pubis izquierda y una pubis derecha, ver Figura 4.4. Es decir, en total disponemos de 11 nubes de puntos de alta calidad, que representan distintos volúmenes de exámenes médicos.

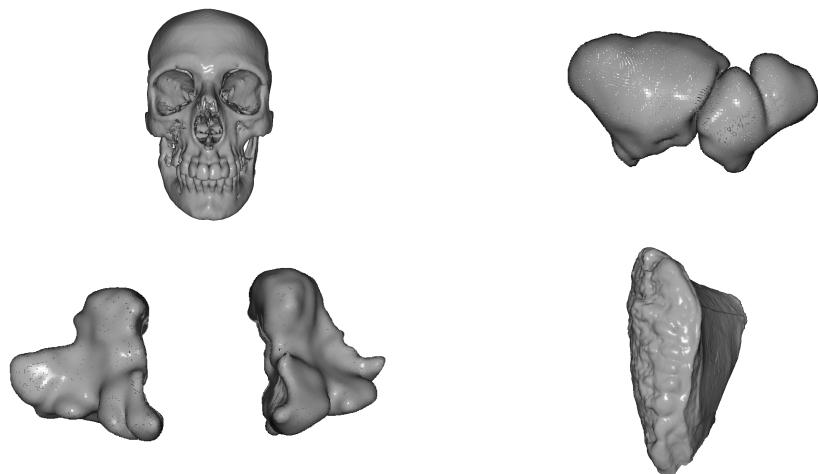


Figura 4.4: Ejemplo de nuestras imágenes médicas. Arriba a la izquierda tenemos un cráneo y a su derecha un seno frontal. Abajo a la izquierda tenemos un maxilar y a su derecha el pubis derecho.

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 la sección 5) y se eliminaron aquellos puntos aislados de todos, frutos de errores en el algoritmo de generación de las nubes de puntos a partir de segmentaciones DICOM.

4.2. Métodos

Como se pudo observar en la Sección 3, actualmente hay una tendencia, justificada, a los métodos de ML. Por otro lado, se tuvo que descartar todos los métodos que tuvieran en cuenta información de textura, cosa que no existe en los volúmenes médicos habituales. Además, necesitamos información perceptual de la imagen en su totalidad y no de regiones locales específicas. Es por ello, que optamos por evaluar el método VQA[28].

4.2.1. Zhang et al[26]

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 vectores soporte para la regresión. Para ello necesitamos definir que tipo de características queremos extraer.

Zhang et al[26] proponen utilizar características geométricas y de color. Para la primera, extraen la curvatura(4.1), anisotropia(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 λ_1 , λ_2 y λ_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 $\sigma = \frac{6}{29}$. Para cada tipo de características se normaliza como (4.9).

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

Donde F es la característica extraída y C una pequeña constante para la estabilidad numérica. Además, argumentan la alta correlación entre la entropía con la distorsión por quantización. Es por ello, que también estiman la entropía de las características. Por último, vieron que la distribución de los puntos se veía afectada con las distorsiones, y es por ello que a las características geométricas las cuantifica y asimila a la distribución gaussiana y gamma.

4.2.2. VQA-PC

Zhang et al[28] 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*[47]. Siguiendo la motivación de que las deformaciones geométricas no deseadas se presentan de forma abrupta según la perspectiva, ver Figura 4.5, 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, ver Figura 4.6. 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 características extraídos. Proponen concatenar los vectores y aprender una función por medio de una capa totalmente conectada utilizando el MSE.

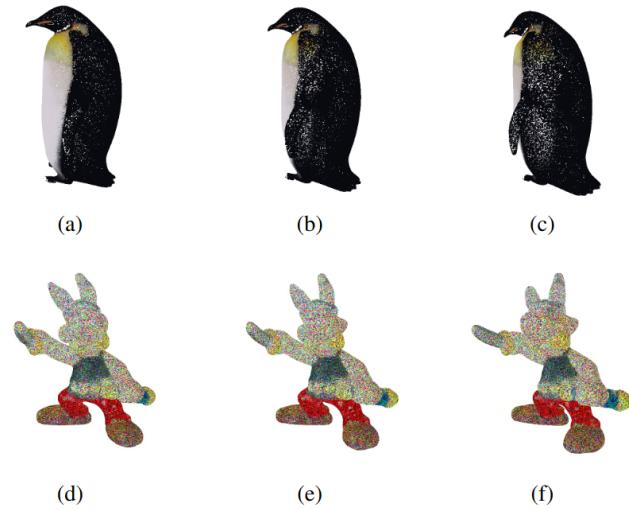


Figura 4.5: Ejemplo de distorsiones que se presentan según la perspectiva. Vemos que al girar el pingüino se empieza a observar un bajo número de puntos en su lateral izquierdo, permitiendo verse a través de él. De forma similar, en la imagen de abajo se ve cierta deformación de la cabeza.

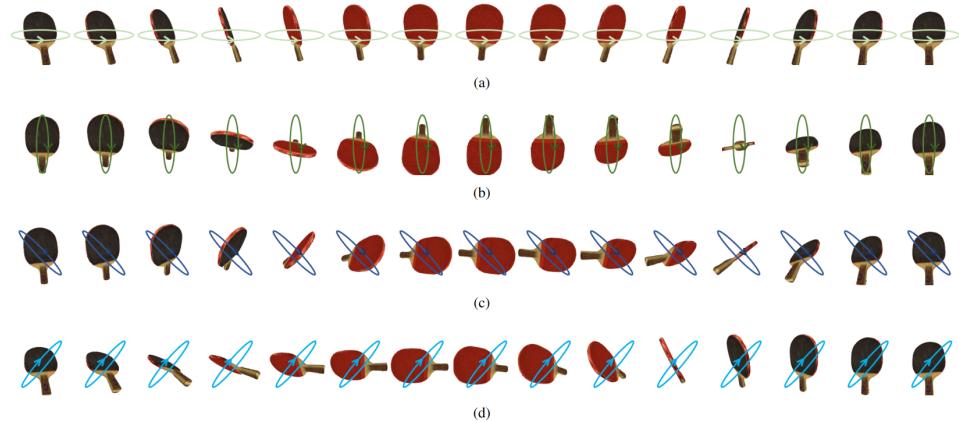


Figura 4.6: Ejemplo de las rotaciones que utiliza el modelo VQA-PC[28]. Se observa que el final de cualquier eje de rotación es la posición inicial, permitiendo así unir de suavemente una secuencia de imágenes encadenada de los ejes que genere un vídeo de rotación utilizado luego para la estimación.

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

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

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

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

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

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_σ representa la coordenada (X,Y,Z) del centro medio de la nube de punto, y σ_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 investigó variaciones de esqueletos ResNet[91]. 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.

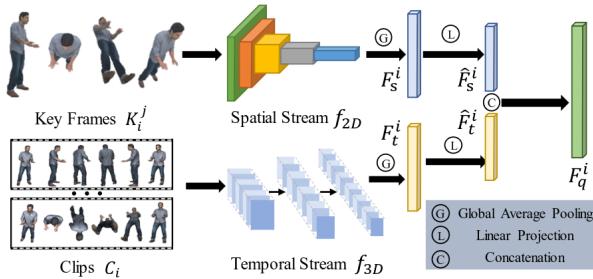


Figura 4.7: Ejemplo detallado de las etapas del método de VQA-PC

Capítulo 5

Implementación y Experimentos

5.1. 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*[53], se hizo uso de la librería PCL[92] en el lenguaje C++.

Para el desarrollo y ejecución de los modelos fue necesario el uso de librería 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[93] y Pyntcloud. 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. 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 5.1.2.

- **Document**, donde se encuentra la documentación del proyecto, incluyendo a este documento.
- **NR3DQA**, implementación y experimentos del método propuesto por Zhang et al[26].
- **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[19].
- **VQA_PC**, implementación de la variante VQA-PC[28] para la estimación de calidad de nubes de puntos y las modificaciones pertinentes sobre los métodos de fusión de características mencionados en [87].
- **imgs**, algunas imágenes relevantes sobre el proyecto.

5.1.1. Obtención de los modelos 3D

Los datos se encuentran en una carpeta del servicio UGRDrive, que provee almacenamiento en la nube para investigadores. Los modelos 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[93] 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[92] y el archivo del directorio `utils/metrics.py`.

Preprocesado de datos

El único preprocesado que sufren los datos iniciales médicos 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`.

5.1.2. 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 %, 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 voxels 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 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, 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.

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.

5.1.3. Métricas

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, por sus siglas en inglés), coeficiente de correlación de orden de rango de Kendall (KROCC, por sus siglas en inglés) y error cuadrático medio (RMSE, por sus siglas en inglés)[1].

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 (5.1).

$$PLCC(x, y) = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^m (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^m (y_i - \bar{y})^2}} \quad (5.1)$$

Como mencionado en [19] o [28], 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 (5.2).

$$Q = \beta_1 \left(\frac{1}{2} - \frac{1}{1 + e^{\beta_2(Q_s - \beta_3)}} \right) + \beta_4 Q_s + \beta_5 \quad (5.2)$$

Donde Q es el valor final normalizado, Q_s es el valor predecido y β_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 [19]. Los resultados se pueden ver en la Tabla 5.1, y las fórmulas adicionales en el Apéndice 7.

	Logística-4	Logística-5	Cúbica-4
SROCC	0.8572	0.9026	0.8957
PLCC	0.8626	0.9107	0.9044

Tabla 5.1: 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 5.3.

$$SROCC(x, y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}} \quad (5.3)$$

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

$$SROCC(x, y) = PLCC(rank(x), rank(y)) \quad (5.4)$$

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.

$$KROCC(x, y) = \frac{C - D}{\frac{1}{2}m(m - 1)} \quad (5.5)$$

Donde C alude a cuantos pares de datos, x e y, que 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.

$$RMSE(x, y) = \sqrt{\frac{1}{m} \sum_{i=1}^m (x_i - y_i)^2} \quad (5.6)$$

5.1.4. Pseudo-MOS

Para la generación de la etiqueta, se optó seguir el camino propuesto por [19]. En opción a generar un entorno controlado con los estándares ITU-R[94, 95], 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 5.2. El rendimiento es medido con el coeficiente de correlación de rangos de Spearman.

Distortion	M-p2po	M-p2pl	H-p2po	H-p2pl	PSNRyuv	H-PSNRyuv	PCQM	GraphSIM	MPED
DownSample	0.881	0.626	0.841	0.811	0.014	-0.016	0.524	0.842	0.857
GaussianShifting	0.741	0.718	0.829	0.834	0.755	0.235	0.816	0.742	0.598
LocalOffset	0.937	0.934	0.770	0.770	0.667	0.019	0.851	0.906	0.897
LocalRotation	0.819	0.712	0.831	0.734	0.327	0.004	0.657	0.723	0.742
Octree	0.779	0.788	0.819	0.752	0.523	0.108	0.676	0.757	0.710

Tabla 5.2: Tabla de métricas para generación de etiquetas de [19]. Para la definición de las métricas, se puede mirar la Sección 3. “M-p2po” sería equivalente a la métrica Po2Po[74], “M-p2pl” es Po2Pl[75]. Donde la “M” ó “H” aluden a la función de distancia utilizada, en este caso MSE ó Hausdorff.

En [19] validaron sus muestras generadas con las métricas SOTA FR por medio de un análisis subjetivo con el estándar y un entorno controlado, definido anteriormente, y obtuvieron una correlación del 90 % entre las muestras etiquetadas subjetivamente y las que fueron etiquetadas objetivamente, ver Tabla 5.3. 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 5.3: Correlación de métricas sintéticas de [19]. 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.

5.1.5. Entrenamiento de los modelos ML y DL

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 5.2.1, 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. 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[88] o WPC[89, 90] 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.

5.2. Experimentos

5.2.1. 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, ver Figura 5.1. 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 nuestro dataset de imágenes médicas, al ser pocos ejemplos se ha realizado un pliegue por modelo 3D. En el caso de LS-SJTU-PCQA[19],

se optó por utilizar cuatro pliegues aleatorios. Para las pruebas preliminares se utilizó el conjunto SJTU[88] con las condiciones indicadas por [19], es decir, con nueve pliegues, uno por modelo.

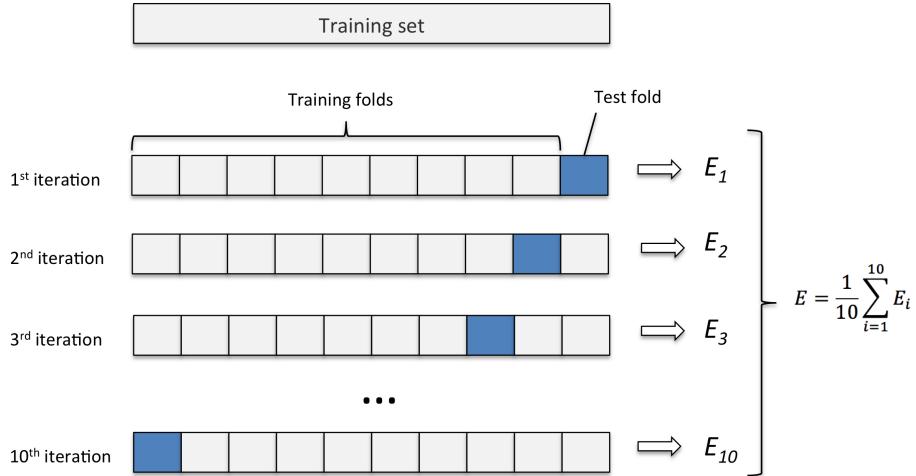


Figura 5.1: Ejemplo de uso de K-fold para la búsqueda de hiperparámetros.

5.2.2. Método con SVM

Para replicar el método de Zhang et al[26] podemos utilizar los scripts en NR3DQA/. Para ello disponemos de unos cuantos scripts para la extracción de las características y visualización de las distribuciones como indica la publicación. Utilizaremos la librería de Pyntcloud 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[88] y WPC[89, 90], los resultados son similares a los obtenidos en la publicación original, ver Table5.4. 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	0.810325	0.777403	0.608302
WPC	0.637953	0.634853	0.463578

Tabla 5.4: Resultados de prueba preliminar con SVM. En SJTU tenemos una mejora de 7% 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 5.5, que el método no es capaz del todo de determinar con precisión la calidad de imágenes, ya que la correlación entre la predicción y el valor real es muy cercano a 0.

Etiqueta Sintética	Modelo	Escalado	SROCC
Valor de la métrica	SVM	RobustScaler	0.1776
Valor normalizado	KNNRegressor	RobustScaler	0.1882
Valor en escala 0-5	DecisionTree	StandardScaler	0.196713

Tabla 5.5: Resultados de prueba preliminar con SVM. 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 predecidos y el valor real.

Incluso utilizando el conjunto SJTU[88] para ayudar al entrenamiento del SVM, no obtenemos mejoras significativas, sino que obtenemos un 0.205. Algo similar ocurre cuando intentamos utilizar más características geométricas como se indica en la publicación[96]. Se argumenta que, 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 5.6, no son sustanciales.

Dataset	PLCC	SROCC	KROCC
SJTU	0.853709	0.820057	0.649406
WPC	0.642356	0.62917	0.455562

Tabla 5.6: Resultado de mejores sobre el método SVM. Se observa mejoras, no sustanciales, sobre los conjuntos SJTU[88], WPC[89, 90] e imágenes médicas. No obstante, todavía sigue por detrás de los métodos DL, como el método VQA-PC que se discutirá más adelante.

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. Posiblemente venga a juego la dificultad del análisis de las estadísticas de escena, tanto en imágenes como en nubes de puntos, a la hora de elegir qué métricas deben extraer para generar un vector de características adecuado para un modelo ML. Los resultados obtenidos con el modelo ML del estado del arte de características NSS 3D indican que, dado el marco temporal para el desarrollo de este proyecto, es conveniente hacer uso de modelos más complejos y permitir que la extracción de características sea automática. A continuación se discutirán los resultados haciendo uso del modelo DL elegido en la Sección 5.1.5.

5.2.3. Experimentos Preliminares DL

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 [28], seguimos el experimento en las mismas condiciones descritas por ellos en el conjunto de datos SJTU[88]. Como ese posee 10 modelos, ver Sección 4.1.1, se realiza un 9-fold. Se ha utilizado los mismos hiperparámetros, estructura de red convolucional y transformaciones de aumentación de datos, ver Tablas 5.8 y 5.7.

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 5.7: Hiperparámetros de prueba preliminar, establecidos por [28].

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 $\sigma = [0.229, 0.224, 0.225]$. Valores con los cuales se normalizaron las imágenes con las que entrenó ResNet[91]. 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 5.2, el modelo parece estar aprendiendo, ver Table 5.9.

Se debe tener en cuenta que, aunque el error cuadrático medio 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.

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

Tabla 5.8: Descripción de la arquitectura ResNet50[91].

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
Promedio	46623.94	0.8813

Tabla 5.9: Resultados de experimento preliminar. 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. El error enseñado es el error MSE que intenta minimizar el modelo, sin utilizar la regresión logística.

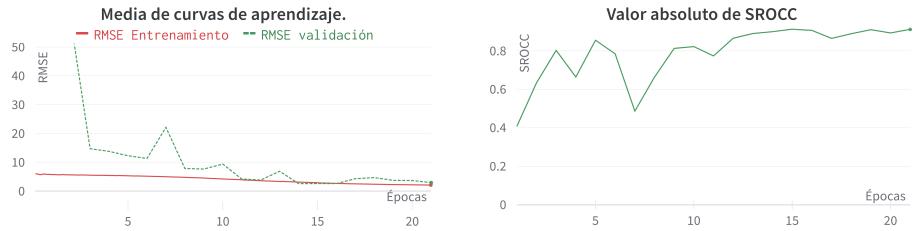


Figura 5.2: Curvas de aprendizaje del test preliminar. Es el comportamiento medio en cada fold. Como vemos el error de validación tiende a bajar, aunque con cierta variabilidad entre épocas en contra del error de entrenamiento que es más estable. A la derecha podemos ver el comportamiento de la métrica SROCC en valor absoluto, ya que resolver el problema inverso sería equivalente.

5.2.4. Propuestas de mejoras

La publicación [87] propone 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 los demás métodos.

La fusión por multiplicación 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 normalizar, si no son ya iguales, a la misma dimensión los vectores. De forma similar, la fusión por convolución permite realizar una proyección lineal de ambos vectores a uno de menor dimensionalidad. Por último, la fusión por *Bi-linear pooling* 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 sin un alto coste computacional, y luego proyecta el vector resultante a una menor dimensión utilizando algún algoritmo de proyección, como *Count Sketch*. También, a fin de evitar la búsqueda exhaustiva de ratios de aprendizaje de forma semi-manual, se hará uso del método de Leslie S. [97], comúnmente aceptado para la rápida convergencia de modelos haciendo uso de una curva de ratios de aprendizaje según el número de iteraciones y épocas. Todos los demás hiperparámetros se mantienen, ver Tabla 5.7.

Para realizar la comparativa de estas mejoras, se utilizará el conjunto de datos LS-SJTU-PCQA[19], ya que es el mayor de todos ellos y con los tipos de distorsiones adecuados para las imágenes médicas. Se entrenará 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). Los resultados promedios se pueden observar en la siguiente Tabla ??

a	b
c	d

Tabla 5.10

Capítulo 6

Conclusiones y Trabajos Futuros

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

En primer lugar, se realizó un estudio de la literatura relativa a la estimación de calidad de imágenes 2D, desde los métodos basado en extracción de características de escenas NSS y modelos de ML, hasta la extracción automática con DL. De aquí, se pudo observar que hay una tendencia, justificada, a los métodos de DL debido a su capacidad de generalización, hecho observado en los resultados obtenidos en diversas pruebas realizadas. También se dió a conocer las diferencias en complejidad entre la estimación de calidad con referencia, donde se posee la imagen original a comparar con la versión distorsionada, frente a los métodos sin referencia, donde la versión original no está disponible para la comparación. A continuación, se estudió el uso de estos métodos sobre imágenes médicas 2D, donde se presenta la dificultad de resolución del problema, parcialmente por ser el subproblema número 2, sin referencia, y debido a que los métodos más conocidos de imágenes genéricas 2D no son directamente aplicados a imágenes médicas 2D. También se observa una tendencia a métodos de DL. Posteriormente, se estudia el estado del arte de métodos dedicados a representaciones tridimensionales, en concreto las nubes de punto que son la representación habitual en el ámbito biomédico. Aquí se presenta la complejidad computacional que impone las representaciones 3D y la extracción de características sobre ellas. Se observa también la misma tendencia que en el ámbito 2D, un salto de ML a DL

debido a los resultados obtenidos sobre diversos dataset. Por último, se concluye que no existe hasta el momento otra investigación que haya tomado el enfoque novedoso de estimar la calidad de representaciones biomédicas 3D directamente.

Se realizó un análisis de los métodos SOTA de estimación de calidad de objetos 3D, tanto desde la perspectiva de ML como del DL. El primero hace uso de características extraídas manualmente utilizando conocimiento humano sobre el 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 comparativa, de los modelos ML se utilizó un modelo capaz de extraer información estática y dinámica de nubes de puntos haciendo uso de múltiples proyecciones 2D y de un video del objeto 3D rotando. De esta forma, podemos simular el HVS. Para ambos modelos se proponen ajustes y pequeñas mejoras basadas en recientes publicaciones y se comparan los resultados. Al igual que la tendencia del estado del arte, sale exitoso el modelo DL.

Para la validación sobre un conjunto de datos médicos se necesitó el desarrollo 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 las etiquetas, se hizo uso de las métricas SOTA del problema con referencia, que es el subproblema más avanzado y que en diversas publicaciones se demuestra la alta correlación de esas métricas y el HVS, pudiendo así justificarse el uso de esas etiquetas generadas antes que la ejecución de un experimento de estimación manual para la obtención del MOS según los estándares, experimento que se sale del marco temporal y planificación de este proyecto. 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.

En proceso: escribir los resultados aquí.

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 repositorio de GitHub https://github.com/CodeBoy-source/TFG_NRPCQA, a excepción de las imágenes médicas ya 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 proponer obtener un MOS manual según los estándares 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 etiquetas sintéticas como lo hacen en la publicación original, permitiendo de un conjunto pequeño extraído manualmente obtener uno 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 video fueran alteradas. 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.

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[29]. Actualmente, ha crecido el número de publicaciones de adaptaciones de PointNet[48] y PointNet++[98] para resolver distintos problemas de nubes de puntos, por lo que quizás se podría adaptar para la resolución de este problema y evitar así la inevitable pérdida de información al proyectar en 2D.

Capítulo 7

Bibliografía

- [1] Y. Ding, *Visual Quality Assessment for Natural and Medical Image*. 2018, págs. 1-4.
- [2] K. Seshadrinathan et al., «Image Quality Assessment,» *The Essential Guide to Image Processing*, 2009.
- [3] Z. Wang y A. C. Bovik, «Modern Image Quality Assessment,» en *Modern Image Quality Assessment*, 2006.
- [4] 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.º 5, págs. 1793-1807, 2013.
- [5] 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.
- [6] 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, 2022.
- [7] 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.
- [8] 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.
- [9] Netflix, *Video Multi-Method Assessment Fusion*, <https://github.com/charlespwd/project-title>, 2016.
- [10] 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.

- [11] Z. Wang, «Applications of Objective Image Quality Assessment Methods [Applications Corner],» *IEEE Signal Processing Magazine*, vol. 28, n.^o 6, págs. 137-142, 2011.
- [12] C. Parisot, «The DICOM standard,» *The International Journal of Cardiac Imaging*, vol. 11, n.^o 3, págs. 171-177, 1995.
- [13] K. H. Höhne, H. Fuchs y S. M. Pizer, «3D imaging in medicine : algorithms, systems, applications,» 1990.
- [14] O. H. Karatas y E. Toy, «Three-dimensional imaging techniques: A literature review,» *European Journal of Dentistry*, vol. 8, págs. 132-140, 2014.
- [15] 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.
- [16] 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, Quantitative Imaging in Cancer.
- [17] Y. Sun y G. Mogos, «Impact of Visual Distortion on Medical Images,» *IAENG International Journal of Computer Science*, vol. 49, págs. 36-45, 2022.
- [18] 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, 2022.
- [19] Y. Liu, Q. Yang, Y. Xu y L. Yang, «Point Cloud Quality Assessment: Dataset Construction and Learning-based No-Reference Metric,»
- [20] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. Palgrave Macmillan, 2005.
- [21] 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*, vol. 4, 2002, págs. IV-3313-IV-3316.
- [22] 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.
- [23] K.-Y. Lin y G. Wang, *Hallucinated-IQA: No-Reference Image Quality Assessment via Adversarial Learning*, 2018.

- [24] 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.
- [25] W. Zhou, Q. Yang, Q. Jiang, G. Zhai y W. Lin, *Blind Quality Assessment of 3D Dense Point Clouds with Structure Guided Resampling*, 2022.
- [26] 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.
- [27] Z. Shan et al., *GPA-Net:No-Reference Point Cloud Quality Assessment with Multi-task Graph Convolutional Network*, 2023.
- [28] Z. Zhang et al., *Treating Point Cloud as Moving Camera Videos: A No-Reference Quality Assessment Metric*, 2022.
- [29] Z. Zhang et al., «MM-PCQA: Multi-Modal Learning for No-reference Point Cloud Quality Assessment,» 2022.
- [30] Q. Yang, Y. Liu, S. Chen, Y. Xu y J. Sun, *No-Reference Point Cloud Quality Assessment via Domain Adaptation*, 2022.
- [31] S. Russell y P. Norvig, *Artificial Intelligence: A Modern Approach*, 3.^a ed. Prentice Hall, 2010.
- [32] Y. S. Abu-Mostafa, M. Magdon-Ismail y H.-T. Lin, *Learning From Data*. AMLBook, 2012.
- [33] T. M. Mitchell, «Machine Learning,» en *Machine Learning*, McGraw-Hill, 1997, cap. 1, pág. 2.
- [34] O. Maimon y L. Rokach, eds., *Data mining and knowledge discovery handbook*, 2.^a ed. Boston, MA: Springer Science+Business Media, LLC, 2010.
- [35] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*. MIT Press, 2016.
- [36] Y. LeCun, Y. Bengio y G. Hinton, «Deep learning,» *Nature*, vol. 521, n.º 7553, págs. 436-444, 2015.
- [37] J. Schmidhuber, «Deep learning in neural networks: An overview,» *Neural Networks*, vol. 61, págs. 85-117, 2015.
- [38] C. Bishop, *Neural networks for pattern recognition*. Oxford University Press, USA, 1995.
- [39] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

- [40] Z. Meng, Y. Hu y C. Ancey, «Using a Data Driven Approach to Predict Waves Generated by Gravity Driven Mass Flows,» *Water*, vol. 12, 2020.
- [41] 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, 2018.
- [42] A. Bakiya, K. Kamalanand, V. Rajinikanth, R. S. Nayak y S. Kadry, «Deep neural network assisted diagnosis of time-frequency transformed electromyograms,» *Multimedia Tools and Applications*, vol. 79, n.º 15, págs. 11 051-11 068, 2020.
- [43] Y. LeCun et al., «Backpropagation Applied to Handwritten Zip Code Recognition,» *Neural Computation*, vol. 1, págs. 541-551, 1989.
- [44] 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.
- [45] 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.
- [46] K. O'Shea y R. Nash, «An Introduction to Convolutional Neural Networks,» *ArXiv e-prints*, 2015.
- [47] C. Feichtenhofer, H. Fan, J. Malik y K. He, *SlowFast Networks for Video Recognition*, 2019.
- [48] 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.
- [49] B. Gokberk, H. Dutagaci, A. Ulaş, L. Akarun y B. Sankur, «Representation Plurality and Fusion for 3-D Face Recognition,» *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 38, págs. 155-173, 2008.
- [50] D. Müller, I. Soto-Rey y F. Kramer, *An Analysis on Ensemble Learning optimized Medical Image Classification with Deep Convolutional Neural Networks*, 2022.
- [51] *Medical Visualization and Volume Rendering*, <https://sbme-tutorials.github.io/2019/CG/notes/7-week7.html>, Visitada: 01-06-2023.
- [52] L. S. Chow y R. Paramesran, «Review of medical image quality assessment,» *Biomedical Signal Processing and Control*, vol. 27, págs. 145-154, 2016.
- [53] R. Schnabel y R. Klein, «Octree-based Point-Cloud Compression.,» 2006, págs. 111-120.

- [54] 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.
- [55] K. Egiazarian, J. Astola, V. Lukin, F. Battisti y M. Carli, «A NEW FULL-REFERENCE QUALITY METRICS BASED ON HVS,» enero de 2006.
- [56] Z. Wang y A. Bovik, «A universal image quality index,» *IEEE Signal Processing Letters*, vol. 9, n.^o 3, págs. 81-84, 2002.
- [57] Z. Wang, E. Simoncelli y A. Bovik, «Multiscale structural similarity for image quality assessment,» en *The Thirtieth Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2, 2003, 1398-1402 Vol.2.
- [58] C. Charrier, O. Lézoray y G. Lebrun, «Machine learning to design full-reference image quality assessment algorithm,» *Signal Processing: Image Communication*, vol. 27, n.^o 3, págs. 209-219, 2012.
- [59] 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.^o 1, págs. 206-219, 2018.
- [60] 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.
- [61] 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, ISSN: 0925-2312.
- [62] 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.
- [63] C. Yim y A. C. Bovik, «Quality Assessment of Deblocked Images,» *IEEE Transactions on Image Processing*, vol. 20, n.^o 1, págs. 88-98, 2011.
- [64] 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.^o 7, págs. 838-842, 2015.
- [65] A. Mittal, A. K. Moorthy y A. C. Bovik, «No-reference image quality assessment in the spatial domain,» *IEEE Trans Image Process*, vol. 21, n.^o 12, págs. 4695-4708, 2012.

-
- [66] 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.
 - [67] 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*, 2014, págs. 1733-1740.
 - [68] V. Bhateja, M. Nigam, A. S. Bhadauria 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.
 - [69] J. Xu et al., *Semi-Supervised Learning for Fetal Brain MRI Quality Assessment with ROI consistency*, 2020.
 - [70] S. Liu, K.-H. Thung, W. Lin, D. Shen y P.-T. Yap, «Hierarchical Non-local 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.
 - [71] T. Iqbal y H. Ali, «Generative Adversarial Network for Medical Images (MI-GAN),» *Journal of Medical Systems*, vol. 42, n.º 11, 2018.
 - [72] K. Qi et al., *Blind Image Quality Assessment for MRI with A Deep Three-dimensional content-adaptive Hyper-Network*, 2021.
 - [73] L. S. Chow y H. Rajagopal, «Modified-BRISQUE as no reference image quality assessment for structural MR images,» *Magnetic Resonance Imaging*, vol. 43, 2017.
 - [74] R. Mekuria, Z. Li, C. Tulvan y P. Chou, «Evaluation criteria for PCC (Point Cloud Compression),» en *ISO/IEC MPEG Doc. N16332*, Geneva, Switzerland, jun. de 2016.
 - [75] D. Tian, H. Ochiaimizu, C. Feng, R. Cohen y A. Vetro, «Geometric distortion metrics for point cloud compression,» en *IEEE International Conference on Image Processing (ICIP)*, Beijing, China, sep. de 2017.
 - [76] E. Alexiou y T. Ebrahimi, «Point Cloud Quality Assessment Metric based on Angular Similarity,» en *IEEE International Conference on Multimedia & Expo Workshops*, San Diego, CA, USA, jul. de 2018.
 - [77] P. Cignoni, C. Rocchini y R. Scopigno, «Metro: measuring errors on simplified surfaces,» *Computer Graphics Forum*, vol. 17, n.º 2, págs. 167-174, ene. de 1998.
 - [78] E. Alexiou y T. Ebrahimi, «Towards a Point Cloud Structural Similarity Metric,» en *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, London, UK, jul. de 2020.

- [79] 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)*, Athlone, Ireland, 2020.
- [80] 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.
- [81] 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.
- [82] 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)*, Berlin, Germany, jun. de 2019.
- [83] 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.
- [84] 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.
- [85] 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.
- [86] W. Zhou, Q. Yang, Q. Jiang, G. Zhai y W. Lin, *Blind Quality Assessment of 3D Dense Point Clouds with Structure Guided Resampling*, 2022.
- [87] 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.
- [88] 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.
- [89] 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*, págs. 1-1, 2022.
- [90] 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.

- [91] K. He, X. Zhang, S. Ren y J. Sun, *Deep Residual Learning for Image Recognition*, 2015. arXiv: 1512.03385 [cs.CV].
- [92] R. B. Rusu y S. Cousins, «3D is here: Point Cloud Library (PCL),» en *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, mayo de 2011.
- [93] Q.-Y. Zhou, J. Park y V. Koltun, «Open3D: A Modern Library for 3D Data Processing,» *arXiv:1801.09847*, 2018.
- [94] ITU-R, «Methodology for the Subjective Assessment of the Quality of Television Pictures,» International Telecommunication Union - Radio-communication Sector (ITU-R), inf. téc. BT.500-13, 2012.
- [95] J. Zhou et al., «Subjective quality analyses of stereoscopic images in 3DTV system,» 2011.
- [96] 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,
- [97] L. N. Smith y N. Topin, *Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates*.
- [98] C. R. Qi, L. Yi, H. Su y L. J. Guibas, *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*, 2017.
- [99] *3D Slicer Image Computing Platform*, <https://www.slicer.org/>, Visitada: 01-06-2023.

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”))

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 β_i , a , b , c y d son parámetros a aprender, Q es el valor normalizado y Q_s es el valor predecido.