Universidad Zaragoza
1474

Graphics and Imaging Lab

Trabajo Fin de Grado

Estimación de la similitud en la apariencia de materiales: análisis de una métrica basada en aprendizaje

Measuring material appearance similarity: Analysis of a learning-based metric

Autor

Ming Tao Ye

Ponente

Ana Serrano Pacheu

Directores

Julia Guerrero Viu
Santiago Jiménez Navarro

Grado en Ingeniería Informática

Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Junio de 2025

AGRADECIMIENTOS

Quiero agradecer sinceramente a mis directores, Julia, Santi y Ana, por vuestra cercanía y amabilidad, por estar siempre dispuestos a ayudar, por todos los conocimientos y consejos compartidos y, sobre todo, por todo el tiempo y esfuerzo dedicados. Gracias por haberme guiado a lo largo de todo este trabajo.

RESUMEN

Los seres humanos somos capaces de inferir una gran cantidad de información sobre los materiales que nos rodean con solo observarlos brevemente: si una superficie es rugosa, brillante, metálica o blanda. Esta percepción innata que tenemos resulta fundamental para la interacción con nuestro entorno. Sin embargo, replicarla computacionalmente es un reto complejo, ya que la apariencia de un material depende de múltiples factores como la geometría, la iluminación o el punto de vista.

En este Trabajo de Fin de Grado se estudia en profundidad un modelo de predicción de similitud en la apariencia de materiales propuesto por Lagunas et al. (2019), basado en aprendizaje profundo con redes neuronales convolucionales entrenadas mediante tripletas de imágenes. El objetivo principal es evaluar la capacidad del modelo para alinear su comportamiento con la percepción humana y, especialmente, analizar su rendimiento fuera del conjunto de datos original.

Para ello, se han seleccionado y utilizado múltiples fuentes de datos: desde imágenes sintéticas generadas con BRDFs medidas hasta imágenes de fotografías reales de materiales. Se han explorado distintos factores que pueden afectar al rendimiento del modelo, como la variación de geometría, material o iluminación, así como transformaciones en las imágenes como recortes o enmascarados. Además, se han comparado distintas métricas de similitud —como CSSIM, *Maximum Mean Discrepancy* o métricas basadas en BRDF— con el modelo de Lagunas. También se ha evaluado la robustez del modelo frente a varias transformaciones y se ha implementado una visualización de activaciones mediante Grad-CAM y PCA, que ha permitido analizar internamente en qué se fija la red al realizar sus predicciones.

Los resultados obtenidos reflejan que el modelo presenta un rendimiento competitivo, pero muestra una sensibilidad notable a las variaciones en el fondo o en las condiciones de iluminación, lo cual puede afectar negativamente a su coherencia perceptual. Las visualizaciones han revelado que el modelo tiende a fijarse en zonas con reflejos y en regiones de alto brillo o contraste, en ocasiones desplazando su atención hacia el fondo.

Por último, se ha modificado el modelo base para que ignore el fondo de las imágenes mediante la aplicación de máscaras durante el entrenamiento, eliminando así las activaciones de las regiones no pertenecientes a la geometría del objeto. Además, se ha realizado un proceso de optimización automática de hiperparámetros y se han analizado las respuestas de este modelo modificado comparado con las del modelo base, aunque sin lograr mejoras sustanciales de robustez.

En conjunto, este trabajo aporta un análisis detallado del modelo de Lagunas et al. (2019) de similitud en la apariencia de materiales, destacando tanto sus puntos fuertes como sus limitaciones y planteando futuras líneas de mejora. Entre ellas, se encuentran el diseño de nuevas estrategias de entrenamiento, la incorporación de mecanismos que enfoquen la atención en la geometría y el material, o la exploración de modificaciones arquitectónicas más adaptadas a la percepción visual de materiales.

Índice

Capítulo 1

Introducción

Los seres humanos poseemos una capacidad extraordinaria para interactuar con nuestro entorno, y una parte fundamental de esta interacción es nuestra habilidad para reconocer materiales, comparar su apariencia e incluso inferir muchas de sus propiedades clave (como el brillo o la suavidad) de forma casi instantánea, simplemente observándolos. Si bien se han propuesto numerosas técnicas para la clasificación de materiales, esta percepción de los materiales es un proceso complejo que involucra múltiples factores, y que aún a día de hoy no se comprende en su totalidad [And11; Fle14; MB10].

La *apariencia de un material* puede definirse como la impresión visual que tenemos de él. Como tal, no solo depende de las propiedades intrínsecas del material en sí (como su *Función de Distribución de Reflectancia Bidireccional* o *BRDF*), sino también de factores externos como la iluminación y la geometría del objeto, así como del juicio humano [Ade01; Fle14]. Por tanto, la similitud en la apariencia se diferencia notablemente de las nociones comunes de similitud entre imágenes (centradas en encontrar diferencias detectables entre sí [Wan+04]) o de la similitud en el espacio de las BRDF, que ha demostrado correlacionar pobremente con la percepción humana [SYG16].

En la actualidad, donde las imágenes fotorrealistas generadas por ordenador son omnipresentes y dado el auge de campos como los materiales computacionales, la utilidad de una medida de similitud para la apariencia de los materiales que se alinee con la percepción humana puede tener un valor incalculable para numerosas aplicaciones como síntesis, clasificación o recuperación[1]. A pesar de los avances en áreas como la clasificación de materiales, el desarrollo de una métrica que capture fielmente la riqueza subjetiva de la percepción humana continúa siendo un desafío fundamental y un área activa de investigación.

En este contexto, el trabajo de Lagunas et al., *A Similarity Measure for Material Appearance* [Lag+19], supone un paso significativo. Los autores presentaron un modelo basado en aprendizaje profundo diseñado para medir la similitud en la apariencia entre diferentes materiales, buscando una correlación con los juicios de similitud humanos.

[1]Hace referencia la recuperación de material (o *material retrieval* en inglés) donde, dado un material, se busca en una base de datos los candidatos más similares.

Para ello, crearon una base de datos con miles de imágenes renderizadas que representaban objetos con variaciones en materiales, forma e iluminación, y recopilaron datos sobre la similitud percibida mediante experimentos de *crowdsourcing*. Estos datos se usaron para entrenar una arquitectura de aprendizaje profundo con una novedosa función de pérdida, que aprendió a predecir la percepción de similitud de materiales de forma muy semejante a la de un humano.

Si bien el modelo de Lagunas demostró un rendimiento muy satisfactorio para predecir la percepción mayoritaria de similitud en un alto porcentaje de los casos, no incluía un análisis exhaustivo de las capacidades de generalización con, por ejemplo, imágenes realistas o con elementos totalmente nuevos para el modelo. En este contexto se enmarca el presente Trabajo de Fin de Grado (TFG), que profundiza en el análisis del modelo evaluando sus decisiones en relación con los datos ya existentes, así como comprobar su comportamiento ante diversos escenarios e imágenes sintéticas y reales nunca vistas durante el entrenamiento. Finalmente, se analizan su robustez ante imágenes con transformaciones y los efectos de un reentrenamiento con optimización de hiperparámetros.

1.1. Objetivo y tareas

El objetivo último de este trabajo es analizar y evaluar el modelo de aprendizaje profundo de Lagunas orientado a predecir la similitud percibida entre diferentes materiales en función de su apariencia visual. Para ello, se han planteado las siguientes tareas:

- Replicar los resultados obtenidos en el artículo original en el que se basa el modelo.
- Comparar los resultados obtenidos con otras métricas de similitud existentes.
- Realizar un análisis del modelo evaluando sus decisiones en relación con los datos ya existentes.
- Configurar un conjunto de datos adecuado para la evaluación y generar datos adicionales mediante transformaciones controladas.
- Evaluar el modelo con nuevos datos de entrada y analizar su rendimiento.
- Apoyar los resultados con técnicas complementarias, como visualizaciones mediante mapas de calor o proyecciones con UMAP [McI+18].
- Reentrenamiento y análisis de resultados del modelo con optimización de hiperparámetros y modificaciones.

1.2. Metodología y planificación

Para el desarrollo del trabajo, se ha desglosado el objetivo principal en una serie de tareas específicas (detalladas en el Apartado 1.1). La planificación y ejecución de estas tareas se ha articulado a lo largo de diversas semanas, con un seguimiento semanal y adaptándose constantemente a los avances y desafíos encontrados.

A continuación, se presenta un resumen cronológico de las actividades clave desarrolladas a lo largo del proyecto agrupado por semanas. De forma complementaria, se ha elaborado un diagrama de Gantt (Figura 1.1) para visualizar mejor la distribución temporal de cada tarea.

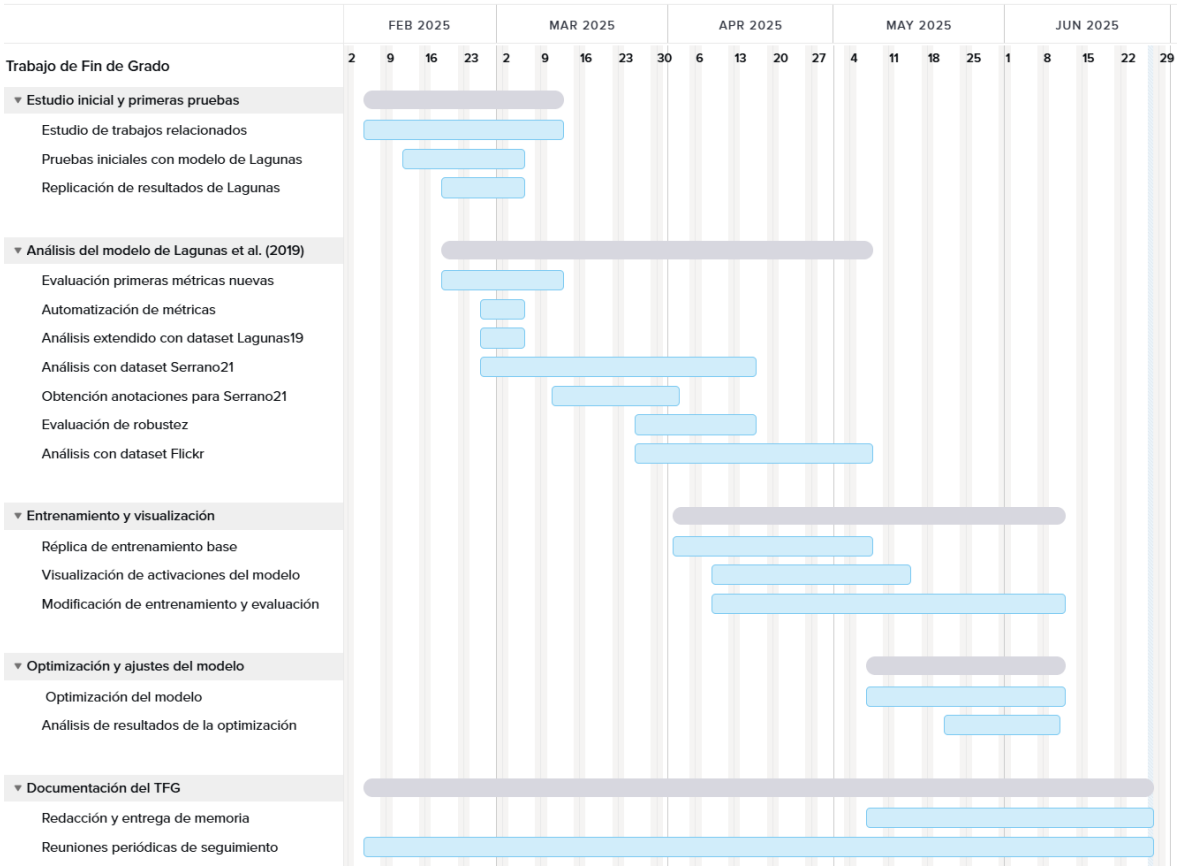


Figura 1.1: Diagrama de Gantt del trabajo

1.3. Entorno y herramientas

Para la realización de las tareas del TFG se ha requerido de un entorno computacional adecuado y el uso de diversas herramientas de software especializadas para llevar a cabo las tareas de análisis, evaluación y experimentación con modelos de aprendizaje profundo. Para ello, se han utilizado principalmente dos entornos de trabajo, que se detallan a continuación junto con el software y las librerías empleadas.

1.3.1. Hardware

Las tareas principales del TFG se han realizado en una máquina local con las siguientes especificaciones técnicas:

- **Procesador (CPU):** AMD Ryzen 7 5700X
- **Memoria RAM:** 32 GB
- **Tarjeta Gráfica (GPU):** NVIDIA GeForce RTX 3070 con 8 GB de VRAM.

Para aquellas tareas como entrenamientos prolongados de modelos, evaluaciones de grandes conjuntos de datos o aquellas que demandaban un uso intensivo de VRAM, se ha utilizado la estación de trabajo Makinon2. Esta es una máquina dedicada del grupo de investigación *Graphics & Imaging Lab*, con las siguientes características:

- **Procesadores (CPU):** Dos Intel Xeon Gold 6140 a 2.30 GHz (total 72 hilos).
- **Memoria RAM:** 256 GB.
- **Tarjeta Gráfica (GPU):** NVIDIA GeForce RTX 2080 Ti con 12 GB de VRAM.

1.3.2. Software y Librerías

El lenguaje utilizado para el desarrollo y la experimentación ha sido Python. La implementación del modelo de Lagunas y las demás funcionalidades están basadas en un conjunto de librerías especializadas, entre las que destacan:

- **PyTorch:** Framework principal para el desarrollo y la experimentación con redes neuronales profundas. Ha sido fundamental para la carga de modelos preentrenados, la definición de arquitecturas y el proceso de entrenamiento y evaluación.
- **NumPy:** Para operaciones numéricas eficientes y el manejo de arrays multidimensionales, crucial en el preprocesamiento y postprocesamiento de datos.
- **OpenCV (Open Source Computer Vision Library):** Utilizada para tareas de procesamiento de imágenes, como lectura, escritura, redimensionamiento, operaciones de color y otras manipulaciones necesarias para adaptar los datos a las entradas del modelo.
- **Scikit-image:** Complementaria a OpenCV, esta librería se ha empleado para análisis de imágenes y métricas de similitud, así como para funciones de preprocesamiento de imágenes.
- **UMAP (Uniform Manifold Approximation and Projection):** Para la reducción de dimensionalidad y visualización de espacios de características de alta dimensión, facilitando la interpretación de los *embeddings* generados por el modelo.

- **Grad-CAM (Gradient-weighted Class Activation Mapping):** Utilizada para generar mapas de calor, que permiten visualizar qué partes de la imagen son más relevantes para la decisión del modelo, ofreciendo una perspectiva sobre su interpretabilidad.
- **Optuna:** Utilizada para la optimización de hiperparámetros del entrenamiento de un modelo de Lagunas mejorado.
- Otras librerías auxiliares como **pandas** para el manejo de datos tabulares, **tqdm** para barras de progreso en operaciones largas y **matplotlib** para visualización de datos y resultados.

Este conjunto de herramientas ha permitido llevar a cabo desde la replicación de resultados hasta la implementación de nuevas métricas y la realización de análisis avanzados.

Capítulo 2

Estudio previo y conceptos teóricos

En este apartado se presentan los fundamentos necesarios para entender el trabajo desarrollado. Se abordan los conceptos clave sobre redes neuronales y su aplicación al análisis de imágenes, así como los principios sobre la apariencia de materiales y el modelo empleado para predecir similitud de apariencia de materiales.

2.1. Fundamentos de apariencia de materiales

2.1.1. Bidirectional Reflectance Distribution Function

La BRDF (o *Bidirectional Reflectance Distribution Function*) es una función que describe cómo se refleja la luz en una superficie, en función de dos direcciones: la de entrada w_i (la dirección desde la que llega la luz) y la de salida w_o (la dirección hacia la que se observa). Es decir, determina cuánta luz es reflejada en una dirección particular w_o desde un punto de la superficie x cuando incide luz desde una dirección concreta w_i (ver Figura 2.1).

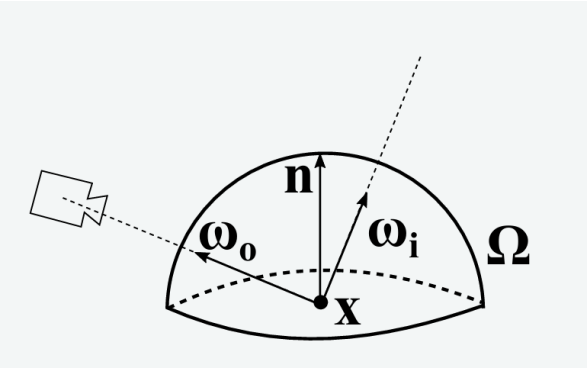


Figura 2.1: Diagrama que muestra cómo un punto x de una superficie con normal n refleja luz en función de la dirección de entrada w_i y de salida w_o.

Esta función es fundamental para modelar la apariencia de los materiales y, según la forma que tome, se pueden simular diferentes tipos de reflectancia, desde superficies difusas (como el yeso o el papel) hasta otras más especulares (como los espejos) e

incluso una mezcla de ambos (como los plásticos). En el Apartado 3.1 se explica más detalladamente el método de obtención de la BRDF de un material y su almacenamiento.

2.1.2. Apariencia de materiales

La apariencia de un material [Del+22][PR12] se refiere a la impresión visual que provoca en el observador y constituye una de las propiedades más importantes para determinar cómo percibimos un objeto. Esa percepción —ya sea el material metálico, brillante o mate— influye notablemente en cómo interactuamos con los objetos y en las expectativas que tenemos sobre su comportamiento.

En ámbitos como el cine, la fotografía o los videojuegos, la apariencia de los materiales juega un papel clave en la construcción visual de las escenas y la ambientación. Por ejemplo, la pintura descascarillada, las manchas de humedad o las tuberías oxidadas evocan espacios antiguos, mientras que las superficies lisas, limpias y los metales brillantes transmiten sensaciones de novedad o modernidad [PR12] (ver ejemplo de la Figura 2.2).

Figura 2.2: Comparación de la apariencia visual del mismo material bajo diferentes condiciones de iluminación. Ambas imágenes muestran una taza de cerámica con el mismo material base, pero bajo condiciones de iluminación diferentes. A la izquierda, la luz suave y difusa de un entorno nublado resalta el acabado mate de la taza; a la derecha, una iluminación más intensa y direccional da la impresión de que el material es más brillante.

Además, esta apariencia no solo depende de las propiedades intrínsecas del material —como su *BRDF*—, sino también de factores externos como la geometría del objeto o la iluminación de la escena. Así, dos imágenes pueden parecer muy distintas y, sin embargo, corresponder al mismo material bajo condiciones diferentes. O al contrario: pueden parecer similares visualmente, pero no compartir el mismo material. Por todo ello, capturar la apariencia no es únicamente un problema físico, sino también profundamente perceptual.

Figura 2.3: Ejemplos de la percepción de la apariencia de materiales. En todos los casos se ha usado la geometría *Havran1* y materiales del dataset MERL [Mat+03]. Izquierda: son materiales distintos, pero se perciben muy similares. Derecha: son el mismo material, pero su apariencia cambia totalmente cuando se cambia la iluminación del entorno.

2.1.3. Métricas de similitud

En el estudio de la apariencia de materiales, contar con una métrica que compare imágenes de forma cuantitativa resulta fundamental para evaluar automáticamente la similitud visual entre materiales en tareas como síntesis, clasificación o recuperación. Para ello, estas métricas buscan modelar de la forma más fiel posible el juicio humano sobre cuándo dos materiales se parecen visualmente.

El objetivo de estas métricas puede parecer simple, pero la gran dificultad a la hora de confeccionar una métrica de similitud reside en cómo definir su comportamiento ante cambios que afectan a la percepción del material, sobre todo cuando se trata de diferencias sutiles pero visualmente significativas y que afectan a la percepción del material (tal y como se puede ver en la Figura 2.3).

Por este motivo, y en línea con los objetivos de este trabajo, se han empleado y comparado distintas métricas que van más allá de la simple comparación a nivel de píxeles. Entre ellas se incluyen métricas que capturan relaciones estructurales, como el SSIM [Wan+04], así como otras que consideran propiedades físicas del material, como el error cuadrático medio (RMS, por sus siglas en inglés) [FFG12].

2.2. Modelo de predicción de similitud

En este apartado se describen los aspectos clave del funcionamiento del modelo neuronal propuesto por Lagunas et al. (2019) [Lag+19] que es utilizado a lo largo de todo este trabajo. Dicho modelo plantea el aprendizaje de una métrica de similitud de la apariencia de materiales mediante el entrenamiento de una red neuronal convolucional[1] con tripletas de imágenes, en las que se indica cuál de dos materiales es más similar al material de referencia. Estos datos provienen de un estudio realizado en la

[1]Para explicaciones más detalladas sobre fundamentos de redes neuronales (funciones de pérdida, redes neuronales y redes neuronales convolucionales), ir ver el Anexo B.

plataforma de Amazon Mechanical Turk (MTurk)[2] con la participación de 603 personas (más detalles en el posterior Apartado 3.2). Este enfoque permite que la red aprenda un espacio de representaciones en el que la distancia entre imágenes se alinee con la percepción humana de similitud visual entre materiales.

2.2.1. Arquitectura

El modelo utilizado se basa en la arquitectura *ResNet* (Residual Networks) [He+16], una red neuronal convolucional que introduce un tipo de conexión llamada *residual*, la cual permite que la información fluya más fácilmente entre capas. En lugar de aprender directamente una transformación completa de la entrada, como ocurre en redes tradicionales, cada bloque residual en ResNet aprende una diferencia (o residuo) respecto a la entrada original. Esta conexión directa entre capas —que suma la entrada a la salida del bloque— permite mantener la información a lo largo de la red y facilita el entrenamiento incluso cuando la red tiene cientos de capas.

En el modelo de Lagunas et al. (2019), se parte de la arquitectura ResNet-34 (ResNet de 34 capas), eliminando su última capa *fully-connected* (fc2) —la correspondiente a la salida del modelo para clasificar la entrada—, puesto que el objetivo es calcular la similitud entre dos imágenes. Así, se preserva su penúltima capa (fc1) —una *fully-connected*— que tiene como salida las características extraídas de la imagen en un espacio de 128 dimensiones (llamado *embedding*) que resume la apariencia del material. En la Figura 2.4 se observa un esquema del modelo.

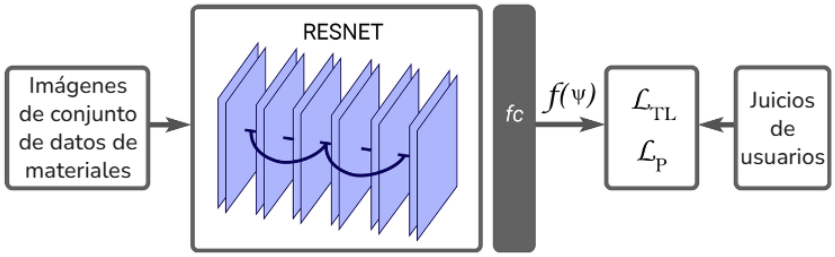


Figura 2.4: Esquema del proceso de entrenamiento del modelo de Lagunas et al. (2019). La entrada que recibe la ResNet se denomina ψ y se trata de los datos del propio conjunto de datos del artículo (posteriormente detallado en el Apartado 3.2). Además, la función de pérdida, definida con los términos L_{TL} y L_P (explicados en el Apartado 2.2.2), también recibe información de percepción de similitud proveniente de usuarios. $f(\psi)$ representa el vector de características de 128 dimensiones.

2.2.2. Función de pérdida

La función de pérdida del modelo propuesto por Lagunas et al. (2019) combina dos términos ponderados de forma equitativa:

[2]Más información en: `https://www.mturk.com/`

$$\mathcal{L} = \mathcal{L}_{\text{TL}} + \mathcal{L}_{\text{P}} \tag{2.1}$$

El primer término, denominado *Triplet Loss Term* ($\mathcal{L}_{TL}$)[3], se basa en las respuestas recogidas mediante experimentos de comparación perceptual realizados a través de la plataforma Amazon Mechanical Turk. En cada tripleta (r, a, b), r es una imagen de referencia, a es la imagen elegida por la mayoría de participantes como la más similar a r, y b es la otra opción descartada.

Este término de la función tiene como objetivo que, en el espacio de características aprendido por la red, la distancia entre los *embeddings* (vector de características) de r y a sea menor que la de r y b por al menos un margen μ. Es decir, busca que el modelo aprenda un espacio donde las imágenes consideradas más similares por los usuarios estén más cercanas entre sí que aquellas percibidas como distintas. Matemáticamente, este término se formula de la siguiente manera:

$$\mathcal{L}_{TL} = \frac{1}{|\mathcal{B}|} \sum_{(r,a,b)\in\mathcal{B}} \Big[\|f(\psi_r) - f(\psi_a)\|_2^2 - \|f(\psi_r) - f(\psi_b)\|_2^2 + \mu \Big]_+ \tag{2.2}$$

En este caso, $f(\psi_k)$ representa el *embedding* de la imagen k (véase la Figura 2.4) y $\mathcal{B}$ es el conjunto de tripletas del lote de datos actual para las que se dispone de comparaciones humanas anotadas. El operador $\|\cdot\|_2^2$ es la distancia euclídea (o norma L2) al cuadrado y $[\cdot]_+$ aplica una función ReLU (es decir, máx$(0, x)$). Así, la función asegura que solo se penalicen los casos en los que la red no respeta la relación de similitud perceptual establecida por los usuarios.

El segundo término, denominado *Similarity Term* ($\mathcal{L}_P$), tiene como objetivo maximizar la probabilidad de que el modelo escoja la misma imagen que los usuarios de las comparaciones por tripletas del MTurk. De esta manera, se introduce una forma adicional de supervisión mediante la maximización del logaritmo de la verosimilitud (en inglés, *log-likelihood*)[4].

Para ello, dado un lote de tripletas (r, a, b) del lote de entrenamiento, se define la similitud perceptual entre las dos imágenes x e y en el espacio de características como:

$$s_{xy} = \frac{1}{1 + d_{xy}} \quad \text{con} \quad d_{xy} = \|f(\psi_x) - f(\psi_y)\|_2^2 \tag{2.3}$$

donde $f(\psi_x)$ es el *embedding* de la imagen x y d_{xy} es la distancia euclídea de $f(\psi_x)$ y $f(\psi_y)$ al cuadrado. En esta expresión de similitud (s_{xy}), un valor de 1 significa que las imágenes x e y son completamente idénticas, mientras que un valor de 0 implica que no guardan ninguna semejanza.

A partir de estas similitudes, se define una distribución de probabilidad sobre la elección entre a y b, de modo que tenemos:

[3]Nótese que este término está claramente inspirado en la función *Triplet Margin Loss*, explicada anteriormente en el apartado B.2.

[4]Más información en el siguiente enlace: `https://www.statlect.com/glossary/log-likelihood`

$$p_{ra} = \frac{s_{ra}}{s_{rb} + s_{ra}}, \quad p_{rb} = \frac{s_{rb}}{s_{rb} + s_{ra}} \tag{2.4}$$

donde p_{ra} representa la probabilidad de que el modelo elija correctamente la imagen a, es decir, aquella que los humanos consideraron más similar a la referencia r. El caso de p_{rb} es análogo.

Con esta formulación, el término de pérdida $\mathcal{L}_P$ se expresa como la media de los logaritmos negativos de dichas probabilidades para todas las tripletas del lote ($\mathcal{B}$):

$$\mathcal{L}_P = -\frac{1}{|\mathcal{B}|} \sum_{(r,a,b)\in\mathcal{B}} \log p_{ra} \tag{2.5}$$

De este modo, obtenemos una función que busca reforzar el alineamiento entre las decisiones del modelo y las elecciones humanas, penalizando aquellos casos en los que el modelo otorga baja probabilidad a la opción correcta. En combinación con el término anterior ($\mathcal{L}_{TL}$), este componente aporta una supervisión probabilística que ayuda a suavizar el aprendizaje y a guiar al modelo hacia decisiones más coherentes con la percepción humana.

2.2.3. Conclusión y uso en este trabajo

El modelo propuesto por Lagunas et al. (2019) constituye una base sólida para abordar la predicción de similitud en la apariencia de materiales, gracias a su métrica centrada en alinearse con la percepción humana.

Esta métrica no solo ha sido empleada por los propios autores en Lagunas et al. (2019) para entrenar y evaluar su modelo de similitud perceptual, sino que también ha sido utilizada en trabajos posteriores como el de Lavoué et al. (2021) o el de Serrano et al. (2021), donde sirve como referencia para validar sus métricas y sus resultados.

En este trabajo, se toma este modelo como punto de partida para llevar a cabo todos los experimentos de evaluación y análisis en el posterior Capítulo 4. En particular, se ha realizado un estudio exhaustivo de su comportamiento y capacidad de generalización, incluyendo su rendimiento sobre distintos conjuntos de datos y la influencia de sus hiperparámetros, para entender con mayor profundidad las capacidades y limitaciones del modelo en diferentes contextos.

2.3. Estado del arte y trabajos relacionados

A lo largo de los últimos años, han surgido muchos trabajos que abordan la percepción de materiales desde múltiples enfoques. Aparte del modelo de Lagunas et al. (2019) [Lag+19] comentado previamente, entre las propuestas más recientes y cercanas a Lagunas se encuentra el trabajo de Serrano et al. (2021) [Ser+21]. Este introduce una métrica supervisada para predecir atributos perceptuales de los materiales y que están relacionados con el brillo (e.g. luminosidad y contraste de reflejos). Estos atributos son

aprendidos directamente desde imágenes mediante técnicas de aprendizaje profundo, lo cual en sus pruebas resulta en un rendimiento superior al de Lagunas.

Siguiendo esta misma línea, Guerrero-Viu et al. (2024) [Gue+24] presenta una métrica supervisada que mejora la predicción del brillo, uno de los atributos perceptuales más relevantes en la apariencia de materiales. En este aspecto, su propuesta supera el rendimiento del trabajo de Serrano et al. (2021), al introducir una estrategia de aprendizaje débilmente supervisado que combina un número reducido de anotaciones humanas con etiquetas generadas automáticamente. Además, sus predictores mantienen una alta coherencia con la percepción humana incluso bajo cambios de iluminación y punto de vista, consolidándose como el nuevo estado del arte en predicción de brillo.

En el campo del aprendizaje profundo también merece mención la métrica LPIPS [Zha+18], que compara activaciones internas de redes convolucionales preentrenadas para tareas de clasificación, logrando correlaciones altas con la percepción humana. Aunque su aplicación principal no está centrada en materiales, sí representa un enfoque relevante de comparación perceptual aprendida.

Por otra parte, también existen trabajos que abordan la apariencia desde otras perspectivas. Por ejemplo, métricas como SSIM [Wan+04] y sus variantes (e.g. CSSIM [HB17; Ven+21] y CCWSSIM [Ber23]) se han empleado como medidas perceptuales generales en tareas de evaluación de calidad o reconstrucción visual.

Por último, destacan también las métricas analíticas aplicadas sobre datos físicos de materiales, como las basadas en distancias entre BRDFs. Entre ellas, se encuentran aquellas que usan Root Mean Square (RMS) y sus variantes [FFG12], así como distancias avanzadas basadas en Optimal Transport o Maximum Mean Discrepancy (MMD) [Fey+19; Lav+21], que incorporan nociones de geometría en el espacio de distribuciones.

Capítulo 3

Conjuntos de datos utilizados

Con el objetivo de evaluar el modelo de Lagunas y analizar sus capacidades de generalización, se han seleccionado y utilizado distintos conjuntos de datos a lo largo del TFG para obtener resultados. Cabe destacar que en todo momento se han usado materiales isotrópicos[1] por simplicidad. En este capítulo se explican en profundidad estos conjuntos de datos.

3.1. Materiales medidos: MERL BRDF

La base de datos de materiales MERL BRDF[2] es una de las más conocidas y utilizadas en el ámbito de la apariencia de materiales. Está compuesta por mediciones precisas de la BRDF de 100 materiales reales (apreciables en la figura 3.1) que fueron obtenidas mediante un *gonio-reflectómetro* de alta velocidad diseñado específicamente para este propósito (figuras 3.2a y 3.2b). Este dispositivo es capaz de medir BRDFs con gran precisión desde múltiples direcciones de iluminación y visión, generando entre 20 y 80 millones de muestras por material antes del filtrado y procesado [Mat+03].

Tras la medición, cada BRDF se representa como una tabla muy densa de datos discretos en un espacio tridimensional (debido a la simetría isotrópica de los materiales) e incluye una gran variedad de materiales, desde plásticos y pinturas hasta telas y metales.

Gracias a la calidad y variedad de estas mediciones, este conjunto de datos resulta útil tanto para comparar materiales desde un punto de vista analítico (por ejemplo, mediante métricas basadas en BRDF), como para renderizarlos y generar imágenes sintéticas, tal y como ocurre en los conjuntos de datos Lagunas19 y Serrano21 que se describen en los siguientes apartados.

[1] Un material isotrópico presenta propiedades idénticas de absorción y dispersión de la luz en todas las direcciones. Un ejemplo sería una bola de cristal. Más información en `https://www.sciencedirect.com/topics/materials-science/isotropic-material`

[2] Estos datos pueden encontrarse en la siguiente web: `https://www.merl.com/research/downloads/BRDF`

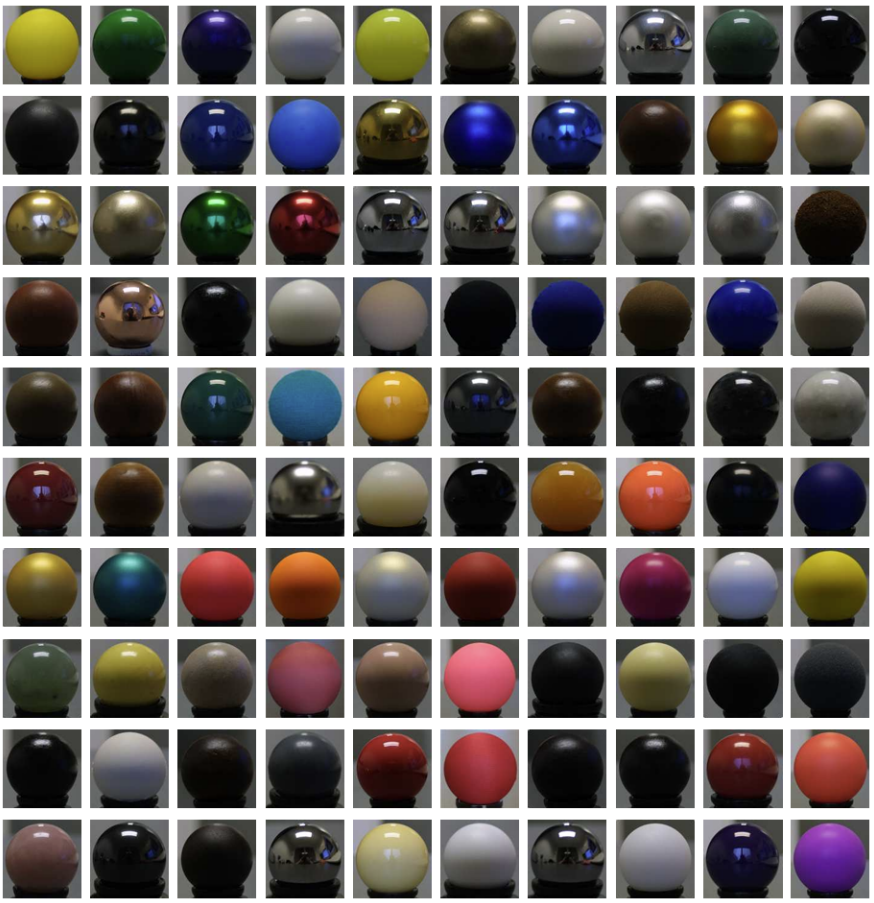
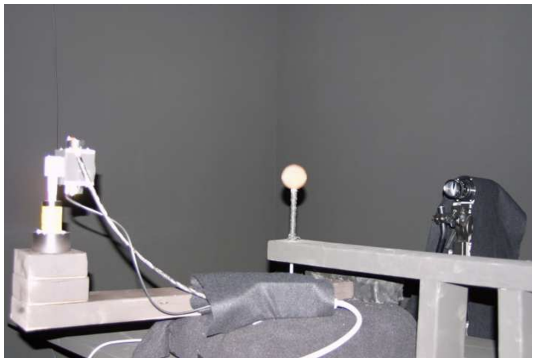
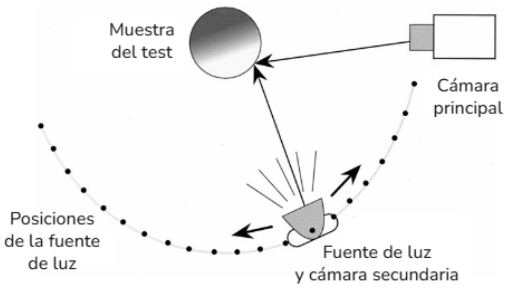


Figura 3.1: Imágenes de los 100 materiales de MERL [Mat+03].



(a) Foto del gonio-reflectómetro usado para medir BRDFs de MERL [Mat+03].



(b) Esquema de la disposición para la medida de BRDF (fuente: [Mar+00]).

3.2. Tripletas de imágenes sintéticas: Lagunas19

A diferencia del conjunto MERL BRDF, que recoge medidas físicas de materiales, este es un conjunto de imágenes sintéticas renderizadas. Dado que en el artículo original de Lagunas et al., 2019 [Lag+19] no se le asignó un nombre específico, en este trabajo

se hará referencia a él como Lagunas19, en alusión al primer autor. Este conjunto de datos está compuesto por 9.000 imágenes generadas mediante la combinación de 100 materiales del conjunto MERL, 6 mapas de entorno (o entornos de iluminación) y 15 geometrías (modelos 3D), incluyendo diferentes perspectivas. A partir de estas imágenes se generaron 25.801 tripletas, que dieron lugar a más de 114.000 comparaciones de similitud de apariencia recogidas a través de la plataforma de *crowdsourcing* MTurk, en la que participaron un total de 603 usuarios[3].

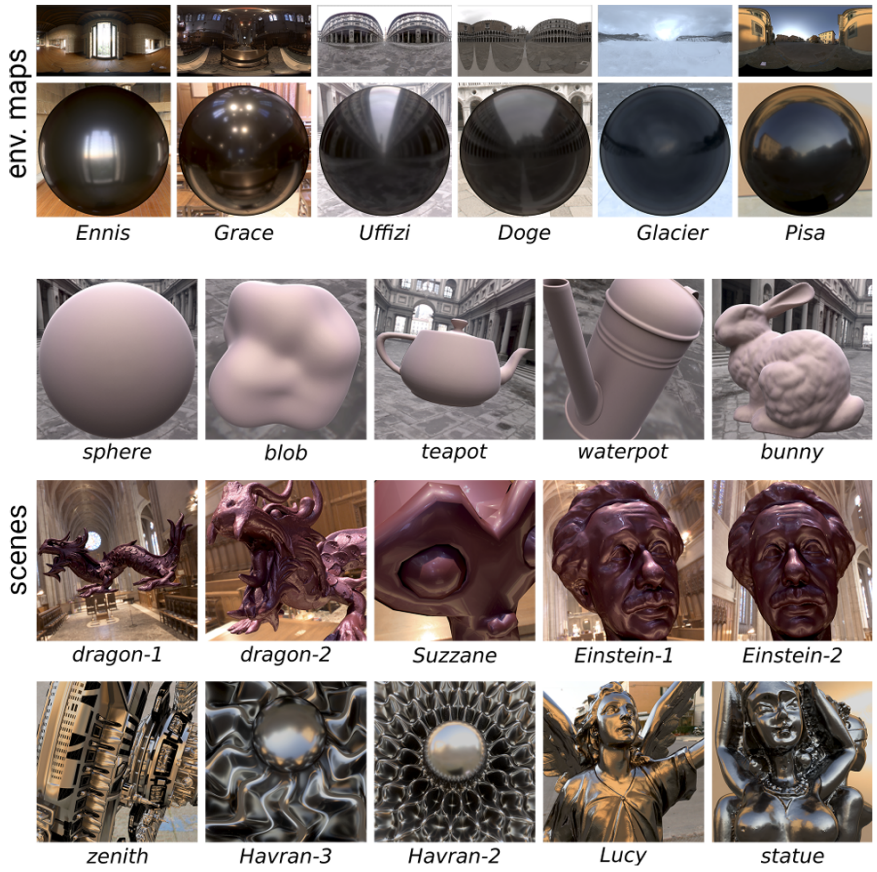


Figura 3.3: *Dataset* Lagunas19 [Lag+19]. Arriba: los seis mapas de entorno utilizados en el dataset, junto con esferas renderizadas con el material *black-phenolic*. Abajo: imágenes de muestra para las quince geometrías, cada una con distintos materiales y condiciones de iluminación.

Es importante destacar que a partir de este punto, las tripletas que se muestren tendrán la nomenclatura de (r, a, b), donde r es la imagen de referencia, a es la imagen del material que más se parece a r y b es la del que menos se parece a r. Esta colocación de las imágenes se ha obtenido gracias a las votaciones (de entre dos y cinco votos totales por tripleta) recogidas en el estudio de MTurk y es el *ground truth* que se ha empleado en Lagunas et al. (2019) para entrenar la red neuronal.

[3]La edad media de los participantes fue de 32 años, con un 46,27 % de mujeres. Los usuarios no fueron informados del propósito del experimento.

En el presente trabajo, el *dataset* se ha utilizado con varios propósitos:

1. Para reproducir y evaluar el modelo original de Lagunas
2. Como base para obtener nuevas estadísticas que permitan estudiar la relación entre el rendimiento del modelo y las decisiones agregadas de los usuarios
3. Para generar nuevos datos de prueba con los que analizar el comportamiento del modelo en distintos escenarios.

Cabe destacar que el conjunto está dividido en subconjuntos de entrenamiento y test, lo que permite evaluar la capacidad de generalización del modelo sobre imágenes no vistas. Esta partición se realiza a nivel de geometría (ver Figura 3.3): únicamente las geometrías *Havran-2* y *Havran-3* se reservan para test, mientras que las otras trece se emplean en el entrenamiento. Además, si se excluyen las tripletas con empate en los juicios de los usuarios, el conjunto final queda compuesto por 21.406 tripletas válidas para entrenamiento (generadas a partir de aproximadamente 7.800 imágenes distintas) y 2.738 tripletas válidas para test (con unas 100 imágenes).

3.3. Datos de atributos de la apariencia de materiales: Serrano21

El conjunto de datos de Serrano et al., 2021 [Ser+21], referido como Serrano21 porque no tiene un nombre específico, se utiliza en este trabajo para evaluar la capacidad de generalización del modelo propuesto por Lagunas. Este *dataset* es particularmente interesante porque comparte algunas geometrías empleadas en Lagunas et al. (2019), pero también introduce nuevos entornos de iluminación, materiales y geometrías que no estaban presentes durante el entrenamiento del modelo original.

Las imágenes del *dataset* están generadas a partir de la combinación de nueve entornos de iluminación[4] y diversas geometrías 3D, produciendo una gran variedad de escenas (figura 3.4). Adicionalmente, cada imagen tiene su correspondiente versión HDR (alto rango dinámico), un formato que conserva la información con mucha más precisión que una imagen estándar e incluye datos como el *albedo*[5] y otras propiedades físicas.

Todo este conjunto de imágenes se complementa con la recopilación mediante *crowdsourcing* de valoraciones perceptuales de atributos de apariencia, con más de 215.680 respuestas para 42.120 combinaciones distintas de material, forma e iluminación. Dichos atributos están organizados en un archivo `.csv` con valores naturales del 1 al 7 y corresponden con las categorías de: brillo, nitidez de reflejos, contraste de los reflejos, metalicidad y luminosidad.

[4] A lo largo del trabajo, se referirá a ellos como *iluminación* para simplificar.

[5] El *albedo* se define como el color puro del material, separado de sombras, reflejos u otros efectos visuales. No depende de la iluminación ni de la geometría de la escena.

Los materiales utilizados en este conjunto de datos provienen de diversas fuentes. Se emplean materiales anisotrópicos de los conjuntos de datos UTIA ([FV14]) y RGL ([DJ18]), aunque se decidió prescindir de ellos para evitar una complejidad innecesaria. Por otro lado, los materiales isotrópicos se obtienen de los conjuntos de datos MERL ([Mat+03]), RGL, versiones editadas de los BRDFs de MERL ([SJR18]) y los equivalentes isotrópicos ([Fil15]) de 50 materiales de UTIA con efectos de alta anisotropía.

Por lo tanto, este conjunto de datos permite ampliar la variedad de los datos al incluir situaciones más diversas y realistas. Pese al gran inconveniente de que no tiene directamente anotaciones de similaridad de tripletas (como Lagunas19), la combinación de elementos conocidos y no conocidos del dataset Serrano21 resulta especialmente útil para llevar a cabo análisis sobre la percepción de materiales en condiciones más desafiantes que las vistas durante el entrenamiento del modelo de Lagunas.

Figura 3.4: Dataset de Serrano21. Primera fila: Mapas de entorno de las nueve iluminaciones. Segunda fila: imágenes de la geometría *bunny* con cada iluminación. Tercera fila: imágenes con la iluminación *Small Cathedral* con todas las geometrías. El material usado por las geometrías ha sido *blue-metallic-paint2* del conjunto MERL [Ser+21].

3.4. Imágenes de materiales en entornos reales: Flickr

Finalmente, la base de datos de materiales Flickr (ó FMD) [Liu+10] está compuesta por imágenes del mundo real fotografiadas desde dos tipos de encuadres: planos cercanos (close-ups) y planos más generales. Tiene diez categorías de materiales y cada una incluye un total de 100 imágenes, 50 para cada tipo de encuadre. Todas las imágenes fueron elegidas manualmente, tras un proceso de selección, para así poder asegurar una alta diversidad en iluminación, composición, color, textura, forma de la superficie, subtipos de material y asociación de objetos.

A diferencia de los conjuntos sintéticos comentados anteriormente, las imágenes de Flickr reflejan la complejidad de entornos reales, por lo que suponen un banco de pruebas especialmente exigente para seguir evaluando la capacidad de generalización del modelo en entornos nunca antes vistos.

En el trabajo original de Lagunas et al. (2019), se empleó este conjunto para mostrar que su modelo, pese a haber sido entrenado únicamente con materiales isotrópicos

y sintéticos, era capaz de recuperar materiales similares en imágenes reales. Concretamente, se seleccionaron ejemplos de las categorías de telas, metales y plásticos, y se evaluó qué imágenes eran las más cercanas en el espacio de características. Aunque los resultados fueron prometedores, los propios autores reconocen que una evaluación más profunda sobre materiales del mundo real quedaba fuera del alcance de su artículo.

Por esta razón, en el posterior Apartado 4.5 de este TFG se propone ampliar ese análisis, usando el conjunto de Flickr para llevar a cabo evaluaciones más exhaustivas del modelo.

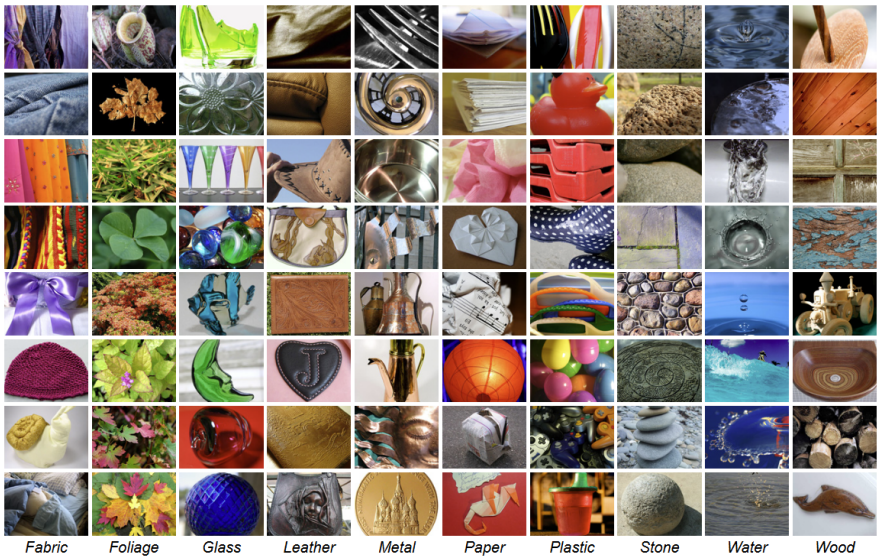


Figura 3.5: Dataset de Flickr. Fotos de las diez categorías de materiales presentes en el conjunto de datos. Fuente: `https://people.csail.mit.edu/lavanya/fmd.html`

Capítulo 4

Evaluación del modelo

En este capítulo se evalúa el comportamiento del modelo en diferentes escenarios. Para ello, se utilizan distintos conjuntos de datos, configuraciones y pruebas que permiten analizar la capacidad del modelo para generalizar y capturar la similitud en la apariencia de materiales.

4.1. Replicación de los resultados de referencia

Antes de realizar pruebas adicionales, se ha replicado la evaluación del modelo base utilizando el conjunto de test original de Lagunas et al. (2019) [Lag+19], que incluye 2.750 tripletas generadas con la geometría *Havran* [HFM16] y materiales del conjunto MERL [Mat+03] (100 en total). Todos los hiperparámetros se han mantenido por defecto, coincidiendo con los indicados en el artículo original de Lagunas.

Los resultados obtenidos (Tabla 4.1) muestran una coincidencia completa en *accuracy* para la métrica de Lagunas[1] y las tres métricas basadas en BRDF (RMS, RMS-cos, Cube-root) [FFG12]. Sin embargo, el rendimiento obtenido de SSIM destaca especialmente porque alcanza un 78,23 % de *accuracy*[2], lo cual supone una mejora de aproximadamente 14 puntos respecto al valor indicado en el artículo original de Lagunas et al. (2019). Esto último puede deberse a diferencias en la implementación concreta de la métrica SSIM o a la posible actualización de librerías empleadas, lo que podría haber influido en los resultados obtenidos.

Por otra parte, se ha comprobado el correcto funcionamiento de otras funciones proporcionadas por el modelo. La primera es el listado de las N imágenes más similares a una de referencia (ver la Figura 4.1) y los resultados son muy satisfactorios. Cabe destacar el caso de las tres imágenes rojizas de la derecha de la fila inferior (Figura

[1]Para calcular el *accuracy*, se toman como casos totales las tripletas que los usuarios del MTurk han votado (y que no han tenido empates) y se evalúan con el modelo de Lagunas et al. (2019). De las predicciones que se obtiene con Lagunas, los aciertos se dividen entre los casos totales y el resultado obtenido es el *accuracy*.

[2]El *accuracy* de SSIM se obtiene comparando las distancias positiva $Dist_{(r,a)}$ con la negativa $Dist_{(r,b)}$ de las imágenes de una tripleta (r, a, b). Claramente, se considera un acierto si $Dist_{(r,a)} < Dist_{(r,b)}$. Luego, el *accuracy* se calcula dividiendo los aciertos entre las tripletas totales.

Métrica	Accuracy (%)	
	Lagunas	Actual
Lagunas19	80,69	82,03
SSIM	64,74	78,23
RMS	64,72	65,70
RMS-cos	64,67	65,74
Cube-root	67,40	67,16

Tabla 4.1: Comparación de *accuracy* entre el modelo original de Lagunas et al. provenientes del artículo vs los resultados obtenidos de replicar los resultados para este trabajo.

4.1). Aunque difieren en color respecto a la tela azul de referencia, el modelo valora la apariencia general y por ello, las considera similares porque las características de reflectancia son parecidas.

La segunda es la visualización, mediante UMAP [McI+18][3], de los *embeddings* de características de un conjunto de datos en un mapa 2D. Para el caso en que los datos son el conjunto de test de Lagunas19 y los *embeddings* son los producidos por el modelo con los mejores pesos, los mapas de la Figura 4.2 tienen unas agrupaciones coherentes y similares a la referencia del artículo original.

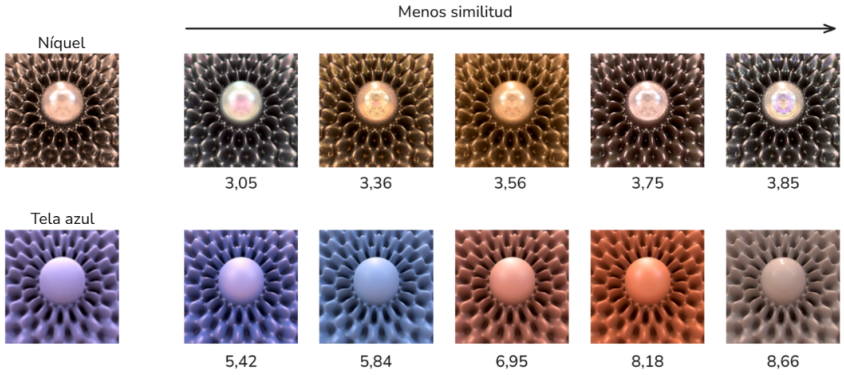


Figura 4.1: Resultados al calcular y mostrar las 5 imágenes (del *dataset* de test de Lagunas19) más similares a una imagen de referencia en cuanto a apariencia de material. Arriba: el material de referencia es el níquel. Abajo: el material de referencia es la tela azul. La geometría empleada en ambos casos es *Havran*.

4.2. Evaluación usando métricas alternativas

Además de las métricas empleadas en el Apartado 4.1, se han explorado otras opciones para evaluar la calidad de los embeddings generados por el modelo de Lagunas.

[3]Más información en `https://umap-learn.readthedocs.io/en/latest/`

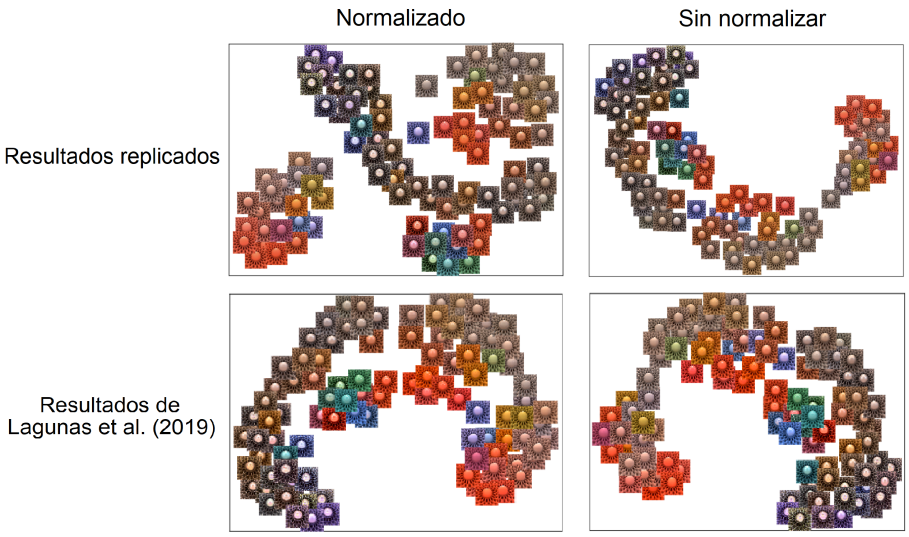


Figura 4.2: Visualización con UMAP del *dataset* de test de Lagunas19 en un espacio 2D basado en el espacio de características proporcionado por el propio modelo de Lagunas. Arriba: resultados replicados en el presente trabajo. Abajo: resultados obtenidos en Lagunas et al. (2019). En ambos casos se observa que los materiales similares están agrupados en los UMAP normalizado y sin normalizar.

En primer lugar, se ha probado una métrica alternativa para el cálculo de distancias mediante variantes del método Maximum Mean Discrepancy (MMD) con kernels diferentes: Sinkhorn, Gauss y Energy [Fey+19]. En todos los casos, los resultados han sido idénticos a los obtenidos con la métrica original de Lagunas, ya que las tres variantes parten de los mismos embeddings generados por la red.

Por otro lado, motivado por estudios como los de Venkataramanan et al. (2019) [Ven+21], Hassan et al. (2017) [HB17] y Jadhav et al. (2013) [PJD13], que demostraron buenos resultados utilizando métricas como SSIM y CSSIM en distintos espacios de color, se ha llevado a cabo una evaluación de cómo influye el espacio de color sobre estas métricas. Para el caso particular de CSSIM[4] en espacios como YCrCb y CIELab, se han empleado ponderaciones adaptadas para dar más importancia al canal de luminancia, dado su impacto perceptual en la similitud visual. En concreto, los pesos son los siguientes: CSSIM-RBG tiene los pesos [0,1664; 0,4932; 0,3404], CSSIM-YCrCb tiene los pesos [0,8; 0,1; 0,1] y CSSIM-CIELab tiene los pesos [0,5; 0,25; 0,25]

Los resultados de todas estas pruebas se muestran en la Tabla 4.2. Se observa que el mejor resultado se obtiene con SSIM aplicado en el espacio CIELab, alcanzando un 79.00 % de *accuracy*. Esto indica que utilizar un espacio de color más alineado con la percepción humana (como es CIELab) y dar más peso a la componente de luminancia, mejora la correlación con el juicio humano de similitud. Por el contrario, los resultados

[4]En la versión propuesta por Venkataramanan et al. (2019) [Ven+21], se ha aplicado SSIM a cada canal del espacio de color y se combinan los resultados según unos pesos específicos.

de CSSIM empeoran ligeramente en los espacios YCrCb y CIELab.

Métrica	**Accuracy (%)**
Lagunas	82,03
MMD	82,03
SSIM-CIELab	79,00
CSSIM-RBG	78,05
CSSIM-YCrCb	75,24
CSSIM-CIELab	73,89

Tabla 4.2: Resultados de las métricas SSIM y CSSIM aplicados sobre diferentes espacios de color.

4.3. Análisis con conjunto de test de Lagunas19

El dataset de Lagunas et al. (2019) tiene mucha información recopilada con MTurk que no se ha llegado a explotar del todo. Por ello, dado que las tripletas de test están etiquetadas según el número de votos obtenidos en el MTurk, se ha decidido usar esos votos para agrupar los resultados del modelo.

4.3.1. Métricas adicionales

Además del *accuracy*, se han añadido las siguientes métricas con el objetivo de ofrecer una evaluación más completa del rendimiento del modelo:

- En cuanto a los votos, puede haber entre dos y cinco votos por tripleta y representan los resultados de las votaciones de similitud por los usuarios de MTurk. Dado que el número de votos de usuarios por tripleta está limitado a cinco, una votación de 5-0 indica unanimidad en la elección de que la imagen A es la más parecida a la imagen de referencia R, mientras que una votación de 3-2 revelaría ciertas dudas en la elección de una opción ganadora.

- La "confianza" de la votación de una tripleta para ver el nivel de seguridad de esa percepción de similitud. Su rango de valores es de [0,5; 1,0] porque, por definición de una tripleta (R, A, B), la opción A es la más votada. Una confianza de 1,0 indica que hay unanimidad en cuanto a la imagen elegida por los usuarios.

$$confianza = \frac{votos\ imagen\ A}{votos\ totales\ tripleta} \tag{4.1}$$

- La distancia ΔDist, basada en las distancias obtenidas entre los *embeddings* del modelo de Lagunas. Puede tomar valores entre [0.0; 2.0]. El valor 0 representa que las diferencias entre la imagen de referencia R respecto a la imagen A y B

son inexistentes, es decir, ambas opciones son igual de parecidas a la referencia. El valor 2 representa que la imagen A es idéntica a la imagen de referencia R y que la imagen B no se parece nada a ella.

$$\Delta\text{Dist} = \frac{1}{N}\sum_{i=1}^{N}(dist(\text{img\_R}_i,\ \text{img\_B}_i) - dist(\text{img\_R}_i,\ \text{img\_A}_i)) \tag{4.2}$$

- La distancia ErrorΔDist está basada en ΔDist, pero en este caso se aplica solo para las tripletas en las que el modelo de Lagunas se equivoca en su predicción. Así, se mantiene el rango de valores [0.0; 2.0], donde ahora el valor 2 representaría que el modelo se ha equivocado al máximo en la similitud de las imágenes A y B respecto a la referencia.

$$Error\Delta\text{Dist} = \frac{1}{N}\sum_{i=1}^{N}(dist(\text{img\_R}_i,\ \text{img\_A}_i) - dist(\text{img\_R}_i,\ \text{img\_B}_i)) \tag{4.3}$$

4.3.2. Resultados

Los resultados obtenidos de pasar el dataset de test de Lagunas19 se pueden observar en la Tabla 4.3. Se puede comprobar que los casos con 1,0 de confianza tienen un *accuracy* mayor a los demás casos y que además, a mayor unanimidad (más cantidad de votos), mayor es el *accuracy* (e.g. las tripletas con votación 5-0 tienen 6,5 % más de *accuracy* que las de 2-0). El caso peor son las tripletas con 3-2 de votación y los resultados son los esperados, ya que en las tripletas con baja confianza es donde se agrupan todas las tripletas difíciles de juzgar (Figura 4.3), lo cual se traduce en un menor *accuracy* para el modelo de Lagunas, así como peor distancia ΔDist.

Figura 4.3: Ejemplo de tripleta (R, A, B) del conjunto de test de Lagunas19 donde la votación es de 3-2 (caso difícil de elegir entre A y B) y, por lo tanto, se considera a la imagen A como la más parecida a la referencia (imagen R).

En base a estos resultados, se ha realizado una prueba de correlación entre la confianza y el *accuracy* de las estadísticas anteriores y se ha obteniendo un coeficiente de Pearson de 0,95, indicando que sí una relación directa positiva casi perfecta. Además, al pasar la misma prueba entre la distancia ΔDist y la confianza, se revela que también hay una relación directa casi perfecta (coeficiente de Pearson de 0,98) entre estas métricas. Esto último implica que, a más unanimidad hay entre los usuarios, más acertado será el cálculo de las distancias entre *embeddings* del modelo Lagunas y más tasa de acierto tendrá.

Votos	Total	Acuerdos	Accuracy	Confianza	ΔDist	Error ΔDist
2-0	372	323	86,8 %	1,00	0,4628	0,2398
2-1	247	185	74,9 %	0,67	0,2379	0,3397
3-0	253	225	88,9 %	1,00	0,5020	0,2251
3-1	283	200	70,7 %	0,75	0,2485	0,2424
3-2	297	182	61,3 %	0,60	0,0914	0,2607
4-0	283	253	89,4 %	1,00	0,5107	0,3016
4-1	390	306	78,5 %	0,80	0,2850	0,2154
5-0	613	572	93,3 %	1,00	0,5372	0,1503
Global	2738	2246	82,03 %	1,00	0,3800	0,2490

Tabla 4.3: Tabla de estadísticas detalladas de los resultados obtenidos al usar el *dataset* de test de Lagunas19 con el modelo de Lagunas et al. (2019). En naranja están subrayadas las filas con unanimidad de votos. En rojo se ha subrayado los casos peores. Nótese que puede haber entre dos y cinco votos por tripleta debido al muestreo adaptativo usado en el MTurk.

Por otro lado, tal y como se puede apreciar en la Tabla 4.4, se ha probado el rendimiento de métricas para los datos en los que Lagunas se equivoca en su predicción y viceversa.

Lagunas (Referencia)				**SSIM (RGB)**		**CSSIM**
Votos	**Total**	**Desacuerdos**	**Tasa de fallos**	**Total**	**Accuracy**	**Accuracy**
2-0	372	49	13,17 %	49	49,0 %	55,1 %
2-1	247	62	25,10 %	62	33,9 %	41,9 %
3-0	253	28	11,06 %	28	53,6 %	57,1 %
3-1	283	83	29,33 %	83	31,3 %	38,6 %
3-2	297	115	38,72 %	115	40,0 %	45,2 %
4-0	283	30	10,60 %	30	53,3 %	60,0 %
4-1	390	84	21,54 %	84	53,6 %	61,9 %
5-0	613	41	6,69 %	41	53,7 %	53,7 %
Global	**2738**	**492**	**17,97 %**	**492**	**43,7 %**	**49,08 %**

Tabla 4.4: Tabla en la que se compara el rendimiento de SSIM y CSSIM sólo para las tripletas en las que el modelo de Lagunas ha fallado.

En primer lugar, obtenemos que SSIM y CSSIM consiguen acertar el 43.7 % y 49.08 %, respectivamente, de las tripletas en las que Lagunas no ha acertado. Es decir, son capaces de acertar una cantidad considerable de las tripletas que son difíciles de predecir para Lagunas.

De forma complementaria, se analiza en la Tabla 4.5 el caso opuesto. En esos resultados, obtenemos que el modelo de Lagunas es capaz de acertar el 53.7 % de los

errores de SSIM, mientras que CSSIM sólo el 14.8 %. Este resultado es el esperado, ya que SSIM y CSSIM comparten una naturaleza técnica similar, lo que puede explicar por qué tienden a fallar en las mismas condiciones.

SSIM (RGB)				Lagunas		CSSIM
Votos	**Total**	**Desacuerdos**	**Tasa de fallos**	**Total**	**Accuracy**	**Accuracy**
2-0	372	53	14,25 %	53	58,5 %	17,0 %
2-1	247	68	27,53 %	68	39,7 %	10,3 %
3-0	253	48	18,97 %	48	72,9 %	16,7 %
3-1	283	88	31,09 %	88	37,5 %	12,5 %
3-2	297	125	42,08 %	125	44,0 %	11,2 %
4-0	283	35	12,37 %	35	62,9 %	14,3 %
4-1	390	107	27,43 %	107	61,7 %	21,5 %
5-0	613	72	11,74 %	72	70,8 %	15,3 %
Global	**2738**	**596**	**21,77 %**	**596**	**53,7 %**	**14,8 %**

Tabla 4.5: Tabla en la que se compara el rendimiento de Lagunas y CSSIM sólo para las tripletas en las que SSIM ha fallado.

En conjunto, estos resultados indican que, aunque SSIM y CSSIM son capaces de complementar parcialmente al modelo de Lagunas en algunos casos difíciles, no existe un patrón claro que permita saber de antemano cuál de las tres métricas sería más fiable en una tripleta concreta. Este hecho limita su utilidad práctica en escenarios reales donde no se dispone de una etiqueta de referencia.

4.4. Análisis con datos sintéticos nuevos

Una vez visto el comportamiento en un entorno conocido para el modelo, el siguiente paso es evaluar cómo se desenvuelve al enfrentarse a datos nuevos. Para ello, se ha optado por el *dataset* de Serrano21, que contiene las siguientes novedades respecto a Lagunas19: 289 materiales nuevos, nueve iluminaciones nuevas y cuatro geometrías nuevas (*cylinder*, *statuette*, *buddha* y *ghost*), dando lugar a un total de más de 23.000 imágenes nuevas para Lagunas.

4.4.1. Obtención de anotaciones de similaridad de tripletas de Serrano21

Este conjunto de datos presenta un inconveniente para la continuación de este trabajo y es que, como no hace comparaciones (sino que para cada imagen se asigna un valor del uno al siete para una serie de atributos perceptuales), no hay ninguna medida de *ground truth* válida para medir la similaridad como en el Apartado 4.3. Por

este motivo, se ha creado la medida $\pi_{(x,y)}$ como métrica de similitud entre los *albedos*[5] de los materiales:

$$f(x) = ln(x+1) \tag{4.4}$$

$$g(x) = \frac{1}{1+x} \tag{4.5}$$

$$\pi_{(x,y)} = normalizar(g(\|f(albedo_x) - f(albedo_y)\|_2)) \tag{4.6}$$

donde la función $f(x)$[6] busca poder comparar los albedos de dos imágenes x e y, la función $g(x)$ busca acotar los valores entre el intervalo [0,06464; 1] (porque se ha computado el valor máximo posible) y finalmente π los normaliza entre 0 y 1. El valor máximo 1 representa que las imágenes x e y son idénticas y como valor mínimo 0 representa que las imágenes x e y no se parecen en nada.

La media $\phi_{(x,y)}$ emplea las métricas de apariencia de Serrano21 y la métrica π con la siguiente fórmula:

$$\phi_{(x,y)} = \frac{\sqrt{\sum_{i=1}^{5}\left[(atr_{(i,x)} - atr_{(i,y)})^2\right] + peso \cdot (6 \cdot (1 - \pi_{(x,y)}) + 1)^2}}{maximum\_value} \tag{4.7}$$

Esta fórmula representa una distancia L2 calculada manualmente que está normalizada, donde *atr* es el valor del atributo i-ésimo (ver Apartado 3.3) de Serrano21 correspondiente a la imagen x o y. El valor final para *peso* es 2,5 y el de *maximum_value* es 16,431677. Para obtener dicho valor, dada la Ecuación 4.7: el rango de valores del sumatorio es $[0,\ 5 \cdot 6^2] = [0,\ 180]$ y el del segundo sumando es $[0;\ 2{,}5 \cdot 6^2] = [0,\ 90]$, por lo tanto, el valor máximo posible del numerador es $\sqrt{180 + 90} = \sqrt{270} \approx 16{,}431677$.

Gracias a esto, ha sido posible evaluar los datos de Serrano21 creando tripletas de imágenes y confeccionando subconjuntos de datos donde varían el tipo de material, geometría y/o iluminación. También se han creado dos métricas para calcular el *ground truth*:

- ΔL2: métrica basada en los atributos de los datos de Serrano21, cuyo rango de valores es $[-16{,}431677,\ 16{,}431677]$. Se define como la diferencia de las distancias $L2_{(r,a)}$ con $L2_{(r,b)}$. En la práctica, es la resta entre el numerador de la Ecuación 4.7 para las imágenes r y a, menos ese mismo numerador para r y b. Un valor de $-16{,}431677$ indica que r y a son idénticas mientras que r y b son completamente distintas; un valor de 16,431677 representa el caso inverso; un valor cercano a 0 sugiere que a y b son igualmente similares (o disímiles) a r.

[5] El albedo representa el color base del material, separado de sombras, reflejos u otros efectos visuales

[6] Se ha usado un logaritmo porque diverge muy lento y los valores de albedo de las imágenes especulares tienen valores de hasta 13.000 por cada canal RGB.

- $\Delta\phi$: variante normalizada de ΔL2 con rango en $[-1,\ 1]$, donde se conserva la misma interpretación relativa entre las imágenes, pero expresada en una escala acotada.

De esta manera, se han podido obtener estadísticas centradas en cada una de estas características y sus variaciones para analizar si existe alguna relación relevante entre ellas.

4.4.2. Análisis del comportamiento en tripletas con imágenes originales

Como primer paso, se ha comparado el rendimiento del modelo con elementos previamente vistos frente a elementos no vistos del conjunto de Serrano21. Para cada grupo, se generaron 3.000 tripletas para ambas categorías, fijando la geometría dentro de las imágenes de una tripleta, y se han dividido los datos de la manera indicada en la Tabla 4.6. Cabe señalar que en ambos casos se emplean las mismas iluminaciones y que Serrano21 no comparte ninguna iluminación con el conjunto Lagunas19.

Categoría	**Visto**	**No visto**
Geometrías	sphere (1), bunny (2), teapot (4), blob (5), dragon (8)	cylinder (3), surface2 / Havran2 (6), buddha (7), statuette (9)
Materiales	MERL	MERL_EDITED, RGL, UTIA
Iluminaciones	Todas	Todas
Imágenes usadas	3904	5610

Tabla 4.6: Comparación de los subconjuntos de evaluación utilizados en función de los elementos vistos y no vistos durante el entrenamiento.

Los resultados se muestran en las Tablas 4.7 y 4.8. En el caso de los elementos no vistos, se sitúa en un 54 % de *accuracy* y el bajo valor de la métrica $\Delta Dist$ ($\sim$0,052) indica que, de media, el modelo realiza las predicciones con muy poca seguridad. Sorprendentemente, el *Error* $\Delta Dist$ es relativamente alto ($\sim$0,294), lo que sugiere que el modelo se equivoca con bastante confianza.

Por el contrario, los resultados para los elementos vistos son coherentes y dentro de lo esperable: el *accuracy* aumenta hasta el 66,9 %, y el valor de $\Delta Dist$ respalda esta mejora, reflejando una mayor seguridad en las predicciones cuando el modelo ya ha sido expuesto a los datos durante el entrenamiento.

En la Figura 4.4 se muestran cuatro ejemplos de tripletas con resultados representativos e interesantes. A continuación, se analizan brevemente cada uno de ellos:

Total	Acuerdos	Accuracy	ΔL2	ΔDist	Error ΔDist
2661	1444	54,27 %	2,9841	0,0496	0,2964

Tabla 4.7: Resultados de 2.661 tripletas aleatorias de Serrano21 no vistas por el modelo de Lagunas. Dada una tripleta, la geometría usada en las imágenes es la misma, mientras que la iluminación y material cambian.

Total	Acuerdos	Accuracy	ΔL2	ΔDist	Error ΔDist
2704	1809	66,90 %	3,7969	0,2236	0,2768

Tabla 4.8: Resultados de 2.704 tripletas aleatorias de Serrano21 vistas por el modelo de Lagunas. Dada una tripleta, la geometría usada en las imágenes es la misma, mientras que la iluminación y material cambian.

Figura 4.4: Ejemplos de predicciones de tripletas con resultados interesantes para la categoría de elementos no vistos (izquierda) y elementos vistos (derecha) por el modelo de Lagunas. Se presentan también los resultados obtenidos del modelo: $Atr_{(x,y)}$ es la similitud entre x e y según sus atributos (es primera parte de la métrica del Apartado 4.4.1), ϕ es la métrica mencionada en el Apartado 4.4.1 y $Dist_{(x,y)}$ es la distancia entre los *embeddings* de x e y. En verde se indica la métrica con el resultado mejor de la tripleta. En rojo lo contrario.

En la tripleta inferior izquierda, los atributos de la imagen r y a son idénticos ($Atr_{(r,a)} = 0{,}00$), y la métrica ϕ —que también incorpora información de color— concuerda con esta similitud. Sin embargo, el modelo de Lagunas elige la opción B pero solo por una diferencia de 0,03 en $Dist$. Podría argumentarse que la opción A es más adecuada, ya que presenta un material oscuro y difuso similar al marrón difuso de la referencia. Aun así, este es un caso difícil.

En la tripleta superior izquierda, la opción B tiene una forma y color muy similares

a la referencia. No obstante, el *ground truth* (ϕ y Atr) selecciona la opción A, cuyo material es más parecido al de la imagen de referencia. En este caso, se observa que el modelo prioriza la similitud visual general por encima de la similitud de material solo.

En la tripleta superior derecha refleja una situación inversa a la observada en la parte superior izquierda. Aquí, las métricas de referencias ϕ y Atr identifican como correcta la opción A (no por mucho), que coincide con la referencia en iluminación y geometría. Sin embargo, el modelo de Lagunas elige la opción B, debido a su coincidencia en el tipo de material (ambos especulares), a pesar de sus diferencias visuales.

Por último, en la tripleta inferior derecha, se muestra un comportamiento normal. Ambas opciones tienen la misma geometría, pero difieren en el tipo de material. La opción A presenta un material difuso, mientras que la opción B es especular. Dado que la referencia también es difusa, todas las métricas coinciden en elegir la opción A como la más adecuada.

Adicionalmente, se ha realizado la misma evaluación que antes, pero esta vez sin separar los datos entre vistos o no durante el entrenamiento y desglosándolos por geometrías. Los resultados se muestran en la Tabla 4.9.

Entre los aspectos más destacables, se observa que la geometría que presenta mayores dificultades es la *statuette*, ya que presenta alta complejidad y no está entre los datos de entrenamiento del modelo. Aun así, es capaz de generalizar, con un *accuracy* del 53,7 %.

Por otro lado, el desglose del valor ΔDist reafirma que el modelo muestra mayor seguridad en sus decisiones cuando se enfrenta a geometrías vistas. La única excepción es la del *dragon*, cuya perspectiva difiere ligeramente de la usada en el entrenamiento y destaca por su teselación y geometría complicada. Por lo demás, todas las geometrías vistas superan un umbral de ΔDist mayor a 0,16. En cambio, las geometrías no vistas parten desde valores cercanos a cero y, en el mejor de los casos, alcanzan un máximo de 0,1458 con *Havran* —una geometría que ya se ha visto en el Apartado 4.3 que tiene un excelente rendimiento—.

Por último, cabe destacar que el *Error* $\Delta Dist$ se acentúa significativamente en algunas de las geometrías no vistas durante el entrenamiento, alcanzando sus valores más altos (es decir, peor desempeño) en casos como *cylinder* (0,3048) y *statuette* (0,3425).

4.4.3. Incorporación de transformaciones y máscaras

Con el objetivo de seguir evaluando la capacidad de generalización del modelo de Lagunas, se ha planteado un experimento comparativo entre su comportamiento con imágenes originales y con imágenes modificadas mediante transformaciones y máscaras.

En la Figura 4.5 se muestra un ejemplo de estas modificaciones: se emplea una transformación de tipo recorte (*crop*) y una máscara aplicada a cada tipo de geometría del conjunto de datos. El recorte se ha diseñado para resaltar las partes más representativas de cada objeto. En el caso particular de la figura *ghost* (o *Havran*), no ha sido

Tipo	Total	Acuerdos	Accuracy	ΔL2	ΔDist	Error ΔDist
sphere	283	190	67,1 %	2,839	1,938	2,839
bunny	319	218	68,3 %	29,593	2,355	2,936
cylinder	303	170	56,1 %	30,801	440	3,048
teapot	302	192	63,6 %	34,586	1,669	2,525
blob	295	191	64,7 %	31,314	1,934	2,290
havran	312	191	61,2 %	38,652	1,458	2,557
buddha	268	165	61,6 %	34,084	1,315	2,647
dragon	271	154	56,8 %	26,509	822	2,842
statuette	283	152	53,7 %	30,424	220	3,425
Global	**2636**	**1623**	**61,6 %**	**32,718**	**1,366**	**2,722**

Tabla 4.9: Resultados de 2.636 tripletas aleatorias de Serrano21 agrupadas por geometría. Dada una tripleta, la geometría usada en las imágenes es la misma, mientras que la iluminación y material cambian.

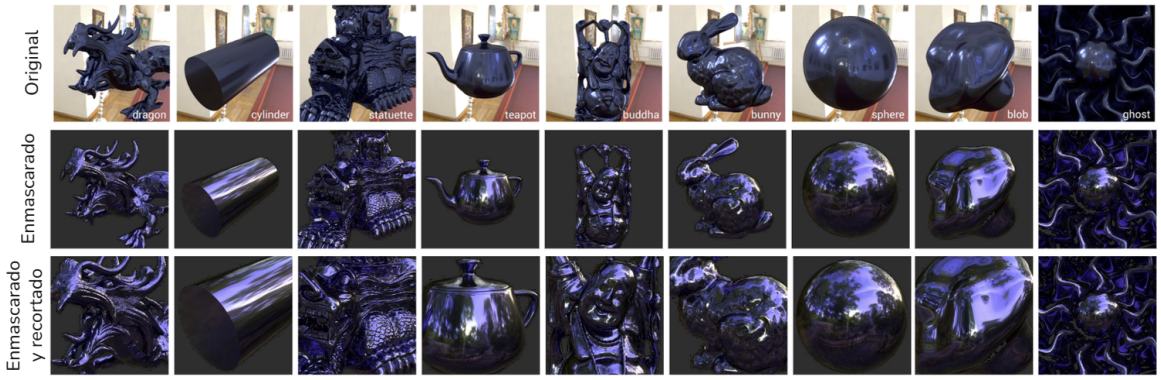


Figura 4.5: Comparación de las imágenes del *dataset* de Serrano21 con sus versiones alternativas. Se ha usado el material MERL *chrome_steel*. Arriba: versión original, iluminación *small cathedral*. Medio: versión con fondo enmascarado, iluminación *ninomaru teien*. Abajo: versión recortada y con fondo enmascarado, iluminación *ninomaru teien*.

necesario aplicar ninguna transformación ni máscara, ya que ocupa toda la imagen y su forma es suficientemente expresiva por sí sola.

Entrando en materia, el análisis (con los resultados en la Tabla 4.10) se centra en usar las mismas 2.336 tripletas (generadas a partir de más de 7.800 imágenes distintas), evaluadas en tres escenarios: usando imágenes originales, imágenes enmascaradas (con el fondo sustituido por un gris18, gracias a las máscaras provistas por el conjunto Serrano21), e imágenes enmascaradas y además recortadas.

A nivel local de la tabla, los resultados obtenidos para cada una de las tres variantes son bastante similares. A nivel global, también se mantiene una tendencia clara: las iluminaciones *Small Cathedral* y *Art Studio* ofrecen un rendimiento superior, mientras

Tipo	Accuracy (%)			$\Delta\phi$
	Original	**Enmascarados**	**Enmasc. + recorte**	
cambridge	51,15 %	46,56 %	46,56 %	0,0929
fish_eagle_hill	55,00 %	51,43 %	47,50 %	0,0867
circus_maximus	45,57 %	47,68 %	48,52 %	0,0946
small_cathedral	52,61 %	52,61 %	51,41 %	0,0929
art_studio	53,91 %	51,74 %	52,61 %	0,0852
tiber	49,60 %	47,98 %	46,77 %	0,0904
auto_service	47,04 %	45,56 %	45,93 %	0,0990
chinese_garden	50,98 %	51,96 %	48,37 %	0,0936
ninomaru_teien	52,36 %	51,97 %	49,21 %	0,0856
Global	**50,94 %**	**49,74 %**	**48,46 %**	**0,0913**

Tabla 4.10: Tabla con datos de 2.336 tripletas de Serrano21 agrupadas por tipo de iluminación. Se compara el rendimiento con los datos originales vs datos enmascarados vs datos enmascarados y recortados. Dada una tripleta, solamente cambia entre las imágenes la geometría utilizada, quedando el resto igual. En verde se indica el mejor valor de la columna. En rojo se indica el peor valor de la columna.

que la iluminación *Auto Service* muestra los peores resultados[7].

Paradójicamente, el modelo de Lagunas parece comportarse mejor en entornos con luminancia media o baja, lo cual es contradictorio porque las iluminaciones de alta frecuencia (como *Auto Service*) deberían favorecer la discriminación de materiales gracias a la aparición de sombras y reflejos más definidos[Fle14; Lav+21]. Esto se debe a que la gran mayoría de las imágenes usadas para el entrenamiento de Lagunas (Figura 3.3) presentan luminosidad media o baja, con la excepción de que, dependiendo de la geometría, los mapas *Ennis*, *Grace* y *Uffizi* tienen en ciertas ocasiones luminosidad alta.

Sin embargo, la métrica ϕ muestra un patrón opuesto: según esta, la iluminación con mejor rendimiento es precisamente *Auto Service*, mientras que *Art Studio* y *Ninomaru Teien* —que tiene características similares a *Small Cathedral*, como se muestra en la Figura 4.6— aparecen entre las peores. Esto resulta coherente teniendo en cuenta que ϕ tiene un cierto sesgo hacia atributos relacionados con el brillo.

En resumen, en las tripletas donde solo varía la geometría, las predicciones del modelo de Lagunas tienden a discernir respecto al criterio del *ground truth* definido por ϕ. Además, los resultados globales no destacan por su precisión, ya que el rendimiento disminuye con cada transformación aplicada (máscara o recorte), sin superar el 51 % de *accuracy*.

[7] *Small Cathedral* se caracteriza por una luminosidad media debida a su fuente de luz de área. *Art Studio* presenta baja luminosidad debido al contraluz. *Auto Service* se define por su alta luminosidad debido a su fuente de luz localizada [Ser+21].

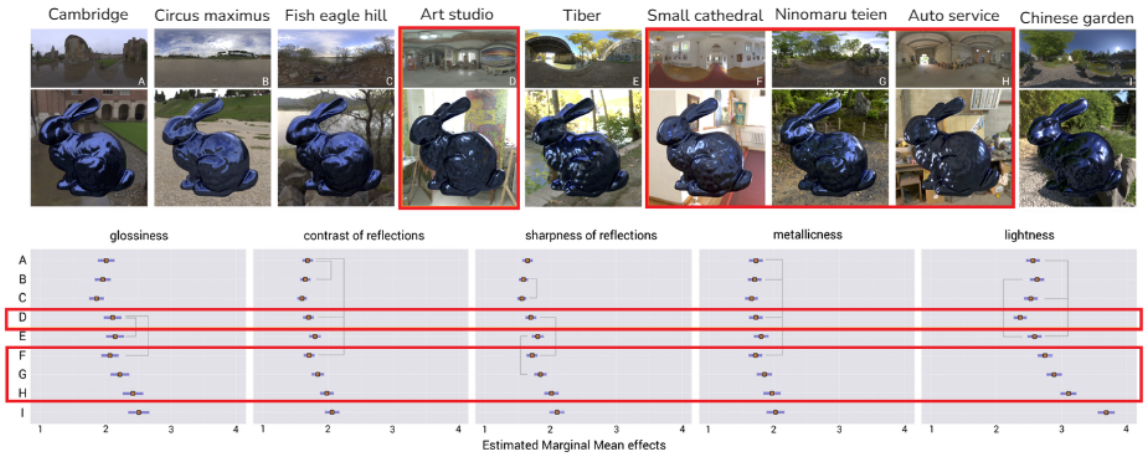


Figura 4.6: Primera y segunda fila: ejemplos de los nueve tipos de iluminación de Serrano21 con el material *blue-metallic-paint2* del dataset MERL. Tercera fila: efecto de la iluminación en los atributos percibidos de Serrano21. *Estimated Marginal Means* aproxima la media de las respuesta para cada factor, ajustado por las variables en el modelo. Los puntos naranjas marcan el valor medio y las barras azules indican un intervalo de confianza del 95 %. En rectángulos rojos están subrayadas las iluminaciones que se han empleado para el test de comparación con masqueados y recortes.

Tipo	**Original**		**Enmasc.**		**Recortados**		**Enmasc. + recorte**		$\Delta\phi$
	Accur.	**ΔDist**	**Accur.**	**ΔDist**	**Accur.**	**ΔDist**	**Accur.**	**ΔDist**	
sphere	68,40 %	0,1995	68,40 %	0,2584	65,95 %	0,2028	65,64 %	0,2233	0,2480
bunny	71,97 %	0,2444	70,70 %	0,2682	66,88 %	0,2206	71,02 %	0,2293	0,2006
cylinder	62,08 %	0,1052	63,61 %	0,1462	61,77 %	0,1125	63,30 %	0,1153	0,1922
teapot	66,67 %	0,1578	63,03 %	0,2105	68,63 %	0,1921	68,35 %	0,2133	0,2068
blob	69,23 %	0,2108	66,57 %	0,2610	59,47 %	0,1174	62,13 %	0,1241	0,2020
havran	64,00 %	0,1986	65,14 %	0,2078	64,00 %	0,1986	65,14 %	0,2078	0,2369
buddha	62,84 %	0,1272	63,14 %	0,1637	61,03 %	0,1558	59,52 %	0,1500	0,2008
dragon	62,39 %	0,1164	58,89 %	0,1434	59,77 %	0,1263	59,18 %	0,1320	0,1572
statuette	59,55 %	0,0923	57,64 %	0,0935	58,60 %	0,1032	60,51 %	0,1034	0,1873
Global	**65,23 %**	**0,1614**	**64,10 %**	**0,1950**	**62,93 %**	**0,1591**	**63,87 %**	0,1669	**0,2036**

Tabla 4.11: Tabla con datos de 3.000 tripletas de Serrano21 agrupadas por tipo de geometría. Dada una tripleta, solamente cambia entre las imágenes el material utilizado. Se compara el rendimiento con los datos originales vs datos enmascarados vs datos enmascarados y recortados. En verde se indica el mejor valor de la columna. En rojo se indica el peor valor de la columna.

Se ha realizado también una segunda comparación de datos. Esta vez, de nuevo se han generado 3.000 tripletas (a partir de más de 7.800 imágenes) en las que, dentro de

una tripleta, se varía exclusivamente el material. Se han analizado cuatro escenarios distintos: uso de imágenes originales, imágenes enmascaradas, imágenes recortadas, e imágenes con ambas transformaciones. Los resultados se recogen en la Tabla 4.11.

A nivel general, se observan diferencias moderadas entre las versiones y los valores de *accuracy* varían ligeramente a la baja en los escenarios con transformaciones. Al aplicar únicamente máscaras, se observa una ligera mejora en la métrica ΔDist, lo que sugiere un aumento en la seguridad de las decisiones correctas del modelo. En el caso de las imágenes recortadas, la seguridad del modelo en los aciertos (ΔDist) desciende de forma generalizada y el *accuracy* alcanza su mínimo para todas las geometrías excepto el *teapot* que aumenta un 2 %. Cuando se aplican tanto recorte como enmascaramiento, la seguridad global supera ligeramente a los datos originales.

En general, se hace notar que ciertas geometrías siguen siendo especialmente problemáticas para Lagunas. Es el caso de *statuette*, que de forma consistente presenta los peores resultados en precisión, acompañada por *buddha* y *dragon*. Estas geometrías, debido a su complejidad estructural y a que no han sido empleadas para entrenamiento (quitando *dragon* en otra perspectiva), parecen dificultar la tarea de predicción para el modelo.

En conclusión, mientras que la aplicación de máscaras no parece afectar demasiado negativamente al rendimiento y puede incluso mejorar la confianza del modelo, el recorte tiende a perjudicar la seguridad de las predicciones. Además, las geometrías más complejas (no vistas) o con formas menos regulares presentan una mayor sensibilidad a estas transformaciones, siendo *statuette* la geometría que más confusiones le provoca al modelo (bajo *accuracy*, bajo ΔDist). Por el contrario, *bunny* es la que mejor rendimiento y consistencia ha tenido a lo largo de todas las tablas, con un ΔDist que refleja la seguridad de Lagunas en la geometría.

Por último, para comprobar si 3.000 tripletas son una muestra representativa del conjunto de datos, se ha realizado un test basado en la Tabla 4.11 pero con una mayor cantidad de tripletas (20.000). Los resultados obtenidos (Tabla C.1) muestran unas estadísticas bastante parecidas a la versión de 3.000 tripletas y manteniendo las tendencias: *statuette*, *cylinder*, *dragon* y *buddha* siguen siendo de las geometrías más difíciles, mientras que *bunny*, *blob* y *sphere* siguen siendo las que mejores resultados dan. Por todo esto, se puede concluir que los resultados de emplear 3.000 tripletas son igual de representativos que usando 20.000 tripletas.

4.4.4. Análisis de robustez

Las diferencias observadas en el Apartado 4.4.3 revelan que realizar transformaciones sobre las imágenes —a pesar de que éstas conservan la misma estructura y contenido esencial— puede provocar un cambio significativo en las decisiones del modelo. Esta ha sido la motivación para comparar las predicciones del modelo en una misma tripleta antes y después de aplicar una transformación visual.

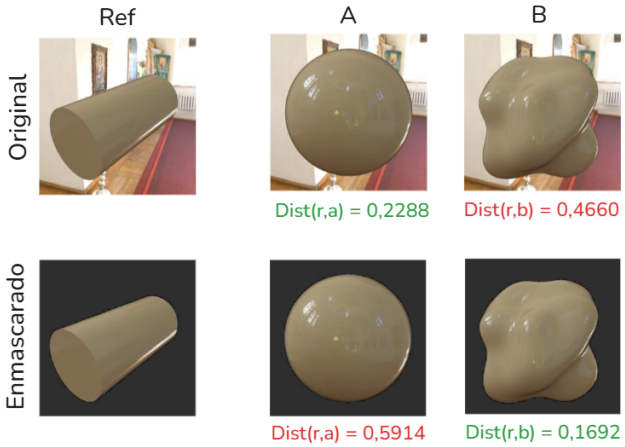


Figura 4.7: Comparativa de dos tripletas del conjunto Serrano21 con la misma iluminación y material. Menor *Dist* es mejor. En verde se indica la mejor distancia de la tripleta (imagen que ha elegido el modelo de Lagunas) y en rojo la peor. Arriba: imágenes originales. Abajo: imágenes enmascaradas.

Como puede apreciarse en la Figura 4.7, en la tripleta con datos originales (parte superior), el modelo Lagunas selecciona correctamente la opción A (la esfera), coincidiendo con el *ground truth*. Sin embargo, al aplicar la máscara de fondo (parte inferior), su respuesta cambia y pasa a preferir con gran seguridad la opción B. Este comportamiento sugiere que el modelo no es completamente invariante al contenido del fondo y que utiliza la información contextual presente en la imagen —como el entorno o el color de fondo— para tomar decisiones de similitud.

A partir de esta observación, el siguiente paso lógico ha sido plantear una prueba que evalúe la robustez del modelo frente a transformaciones. Para ello, se reutilizaron las mismas 3.000 tripletas utilizadas en el test mostrado en la Tabla 4.11, esta vez con el objetivo de comprobar en cuántas de ellas el modelo cambia de opinión tras aplicar una transformación (recorte, enmascaramiento o ambas).

Los resultados, mostrados en la Tabla 4.12, indican que el modelo cambia de decisión en al menos un 16 % de las tripletas, en promedio. Además, las geometrías con mayor número de cambios coinciden con aquellas que ya presentaban un bajo rendimiento en los tests previos: *Cylinder*, *Buddha* y *Dragon*. Curiosamente, llama la atención el caso de *Statuette* que, pese a ser una de las geometrías más difíciles para el modelo, es con la que el modelo ha tenido menos cambios de opinión. Esto podría deberse a que ocupa una gran parte del encuadre en las imágenes, por lo que las transformaciones aplicadas tienen un menor impacto sobre esta. Nótese que *Havran* tiene un 0 % en cambios porque no se le aplican ni máscaras ni recortes.

Tipo	Accuracy (%)		
	Original	**Enmascarados**	**Enmasc. + recorte**
sphere	15,34 %	13,50 %	15,64 %
bunny	15,61 %	19,75 %	22,61 %
cylinder	22,32 %	25,38 %	21,41 %
teapot	19,33 %	14,85 %	19,89 %
blob	18,05 %	25,74 %	21,89 %
havran	0,00 %	0,00 %	0,00 %
buddha	24,47 %	21,75 %	24,47 %
dragon	22,16 %	17,78 %	23,62 %
statuette	12,10 %	14,33 %	17,52 %
Global	**16,57 %**	**16,90 %**	**18,47 %**

Tabla 4.12: Tabla que indica el número de veces que el modelo de Lagunas ha cambiado de opinión al comparar las predicciones de 3.000 tripletas con imágenes originales vs imágenes con transformaciones (enmascarados, recortados o ambos). Dada una tripleta, solamente varía su material.

4.5. Análisis con imágenes reales

En el artículo de Lagunas et al. (2019), se hizo una breve evaluación con el conjunto de datos Flickr. Así, se limitó a mostrar la capacidad del modelo para encontrar los N materiales más similares para tres imágenes de referencia (tela, metal y plástico), dejando abierta la posibilidad de explorar más a fondo la capacidad de generalización del modelo en imágenes reales tomadas en condiciones no controladas.

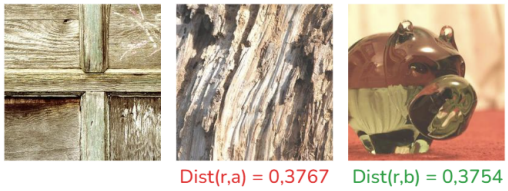
Con este objetivo, se ha analizado el comportamiento del modelo usando imágenes reales de Flickr (3.4). En cuanto al *ground truth*, este se ha incorporado de forma implícita durante la construcción de las tripletas, garantizando que la imagen de referencia y la opción A pertenezcan al mismo tipo de material, mientras que la opción B corresponda a un material distinto. Las nuevas tablas incluyen, además, una columna adicional que indica el material con el que cada clase tiende a confundirse con mayor frecuencia, lo que permite identificar patrones de error del modelo.

El rendimiento global obtenido para el conjunto de datos de Flickr alcanza un *accuracy* del 59,52 %[8]. Si bien es un valor ligeramente inferior al obtenido con datos sintéticos, resulta razonable dado el alto grado de complejidad y variabilidad inherente a las imágenes tomadas del mundo real. No obstante, la confianza del modelo en sus decisiones —con un ΔDist de 0,1099— es bastante baja y refleja una gran convicción en las predicciones realizadas. Por otro lado, pese a que la desviación típica asociada (0,4428) podría parecer alentadora a primera vista, más adelante se demostrará que

[8]Más detalles en el Anexo, Tabla C.2.

este valor no es especialmente representativo, ya que surge de una distribución con alta dispersión de los valores de las distancias.

(a) Tripleta con plástico, plástico y madera.



(b) Tripleta con madera, madera y cristal.

(c) Tripleta con tela, tela y follaje.

(d) Tripleta con cristal, cristal y agua.

(e) Tripleta con metal, metal y agua.

Figura 4.8: Ejemplos interesantes de resultados de tripletas de Flickr. En verde se indica la imagen elegida por el modelo de Lagunas (la que más se parece a la imagen de referencia de la tripleta) y en rojo la imagen que menos se parece a la referencia.

En la Figura 4.8 se presenta una serie de tripletas en las que Lagunas ha votado por la opción incorrecta (por el tipo de material de las imágenes) y en la mayoría de los casos, con bastante firmeza.

Observando las tripletas con atención, se puede analizar qué factores han podido influir en las predicciones del modelo. Por ejemplo, en la tripleta 4.8a, se puede haber decantado por elegir la madera porque esta ha sido barnizada y la luz de la escena favorece el brillo del material, haciendo que se asemeje más al plástico brillante de la imagen de referencia. En la tripleta 4.8c, es normal que haya elegido la opción B porque la suavidad de la textura del lirio y su color (amarillo) se asemejan más a la tela blanca de referencia que a un bordado de lana rosa que ocupa toda la imagen. Sin embargo, también hay casos más confusos como puede ser la tripleta 4.8b, ya que las figuras mostradas no son muy semejantes estructuralmente entre sí y Lagunas no ha sido entrenado ni con texturas de madera ni con cristal.

Un caso interesante es el del material cristal, que obtiene un rendimiento superior a la media a pesar de que el modelo nunca fue entrenado con materiales transparentes (sólo con materiales difusos o especulares).

También destaca el alto rendimiento del agua, sin embargo, su presencia provoca que el modelo lo confunda mucho con materiales como el metal o el cristal (Tabla 4.13 y Figura 4.9), ya que son visualmente parecidos en determinadas condiciones (Figuras 4.8e y 4.8d). Por este motivo, en las siguientes evaluaciones se ha optado por eliminar el agua del conjunto de datos para obtener resultados más estables y menos sesgados.

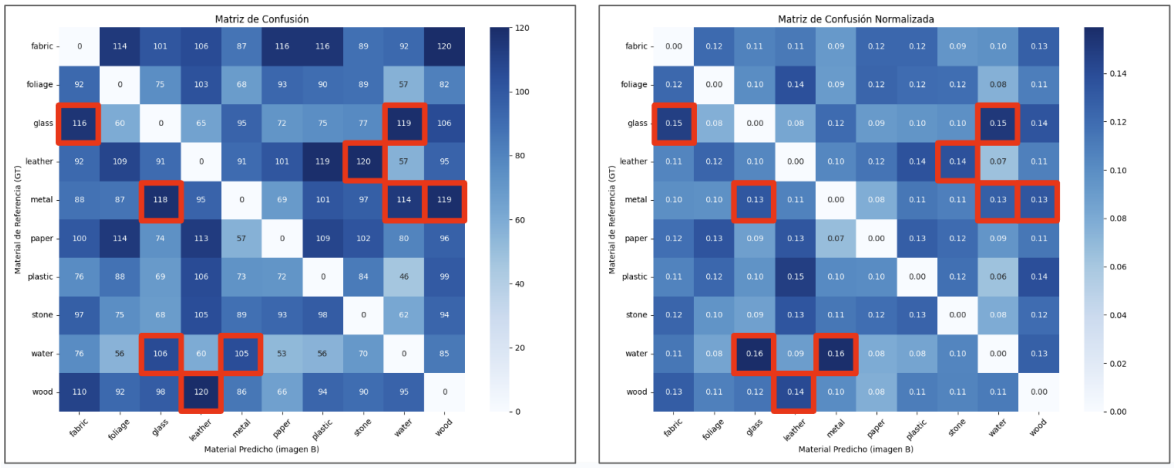


Figura 4.9: Matrices de confusión del conjunto Flickr (con agua) con los errores que ha tenido el modelo de Lagunas. Izquierda: matriz de confusión con datos absolutos. Derecha: matriz de confusión con datos normalizados para cada fila (material de referencia o *ground truth*).

Estos agrupamientos son coherentes con los resultados obtenidos en la matriz de confusión (Figura 4.10), donde se observan confusiones frecuentes entre materiales problemáticos. En particular, destacan las confusiones entre metal y plástico, tela y metal, y plástico y tela, lo que confirma la gran complejidad de este *dataset* y que el modelo tiene dificultades para distinguir materiales con apariencias ambiguas o complejas. Aun con todo, el cristal, el metal y el plástico son los materiales que mejor rendimiento tienen y los que con más confianza predije Lagunas, lo cual se ve reflejado en su *accuracy* y ΔDist superiores a los demás (Tabla 4.13).

En cuanto a las estadísticas generales, el modelo obtiene un *accuracy* del 57,5 %, ligeramente inferior al caso anterior con agua, y la confianza de Lagunas en la toma de decisiones tampoco es muy elevada ($\Delta Dist = 0,0829$). También se ha calculado la desviación típica de ΔDist, lo que indica que el modelo tiende a tomar decisiones relativamente estables. Sin embargo, al analizar la distribución de la confianza del modelo (Tabla 4.14), se observa que las tripletas están muy repartidas entre todos los niveles de confianza. En particular, destacan las pocas tripletas que hay en los niveles de confianza altos, evidenciando el resultado obtenido en *accuracy* y ΔDist.

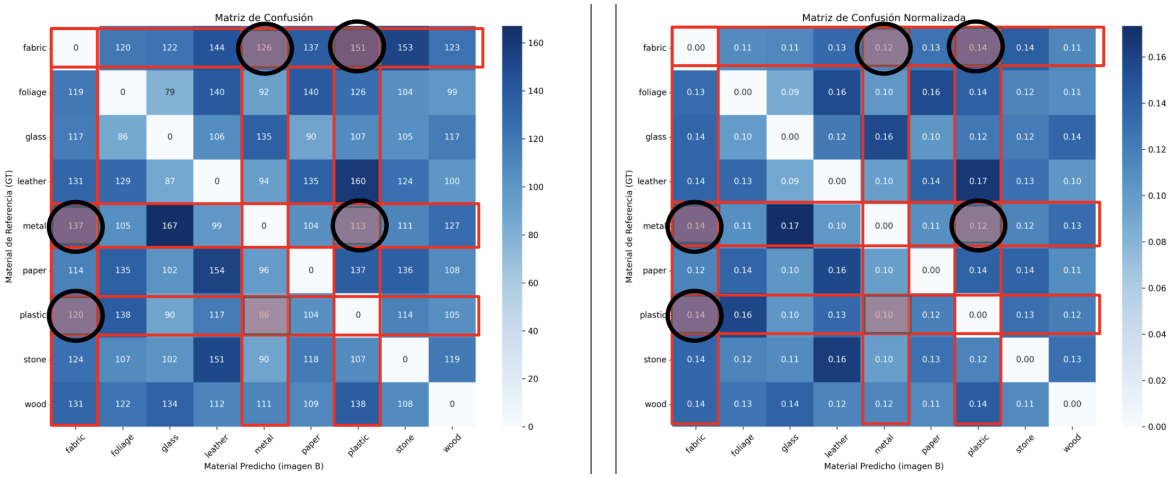


Figura 4.10: Matrices de confusión de Flickr sin agua con los errores que ha tenido el modelo de Lagunas. En rojo se indican las filas de los materiales problemáticos y los círculos indican las principales confusiones. Izquierda: matriz de confusión con valores absolutos. Derecha: matriz de confusión con valores relativos.

Material	Tripletas	Aciertos	Accuracy	ΔDist (media ± σ)	Error ΔDist
fabric	2214	1138	51,40 %	0,0056 ± 0,3918	0,3047
foliage	2200	1301	59,14 %	0,0975 ± 0,4428	0,3033
glass	2222	1359	61,16 %	0,1318 ± 0,4476	0,3037
leather	2274	1314	57,78 %	0,0895 ± 0,4068	0,2711
metal	2296	1333	58,06 %	0,0941 ± 0,4835	0,3509
paper	2193	1211	55,22 %	0,0815 ± 0,4665	0,3107
plastic	2255	1381	61,24 %	0,1077 ± 0,3759	0,2574
stone	2203	1285	58,33 %	0,0927 ± 0,4045	0,2685
wood	2143	1178	54,97 %	0,0437 ± 0,3956	0,294
Global	**20000**	**11500**	**57,50 %**	**0,0829 ± 0,4268**	**0,2966**

Tabla 4.13: Tabla de estadísticas detalladas de los resultados obtenidos al usar 20.000 tripletas del dataset de Flickr (sin incluir agua) con el modelo de Lagunas.

Nivel de confianza	Num tripletas
Muy baja *[0, 0.1)*	4476
Baja *[0.1, 0.3)*	6537
Media *[0.3, 0.5)*	4055
Alta *[0.5, 0.75)*	2972
Muy alta *[0.75, 2]*	1960

Tabla 4.14: Tabla que indica la cantidad de tripletas de la Tabla 4.13 pertenecen al nivel de confianza del modelo basándose en la distancia ΔDist de la tripleta.

En conclusión, aunque los análisis realizados permiten extraer observaciones valiosas, como las agrupaciones de materiales o las confusiones sistemáticas entre ciertas

clases, la gran variabilidad de las imágenes de Flickr genera diversas confusiones y complica la búsqueda de similitudes de imágenes con el modelo de Lagunas. A pesar de ello y vistos los casos de potenciales falsos negativos (Figura 4.8), los resultados obtenidos —lejos de ser triviales— sacan a relucir la capacidad de adaptación del modelo de Lagunas ante situaciones límites, sobre todo en un conjunto de datos tan exigente como lo es Flickr.

Capítulo 5

Análisis visual de representaciones internas

Tras haber evaluado el modelo en múltiples escenarios (Apartados 4.3 y 4.4), se ha observado que su comportamiento puede verse influido por elementos periféricos o irrelevantes desde el punto de vista semántico. Para profundizar en este aspecto y obtener una interpretación más clara de las decisiones del modelo, se ha optado por explorar visualmente sus representaciones internas mediante la técnica *Grad-CAM* [Gc21].

5.1. La técnica Grad-CAM

Esta herramienta permite generar mapas de calor que muestran qué regiones de una imagen han contribuido en mayor medida a la salida del modelo. Su funcionamiento se basa en los gradientes de las activaciones de las capas convolucionales, que se utilizan para ponderar las activaciones de una capa determinada y obtener así una visualización espacialmente coherente de la atención del modelo. Grad-CAM ha demostrado ser útil en tareas como clasificación de imágenes, segmentación semántica, detección de objetos e incluso tareas de similaridad.

En la Figura 5.1 se muestran ejemplos de resultados empleando Grad-CAM, usando, por una parte, un modelo preentrenado para tareas de clasificación y, por otra parte, usando el modelo de Lagunas. Se puede observar que para cada categoría de animal se pinta con colores más cálidos a medida que más influyente sea la zona para dicha salida.

Según el propio autor de la librería, en arquitecturas de tipo ResNet se recomienda utilizar la última capa convolucional para obtener los mapas de calor más representativos. No obstante, con el fin de validar esta elección en el contexto del modelo ResNet34 utilizado en este trabajo, se ha llevado a cabo una comparativa entre diferentes capas y bloques de la arquitectura (véase Anexo D.1 para más detalles de las capas).

Observando los resultados recogidos en la Figura 5.2, se nota claramente cómo a medida que se alcanzan capas más profundas de la CNN, la información se vuelve cada

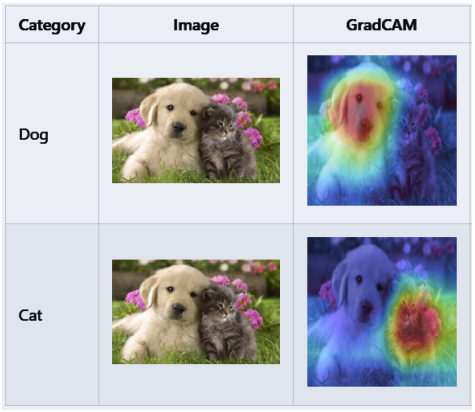
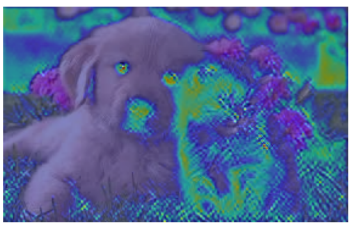
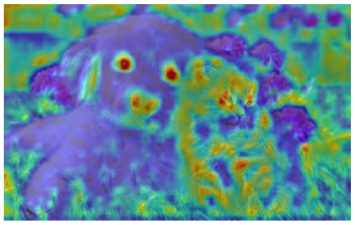


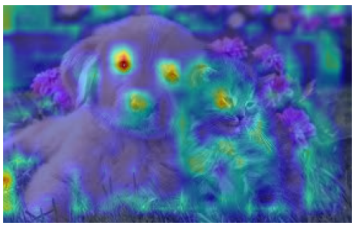
Figura 5.1: Ejemplo de uso de la librería Grad-CAM con una imagen de un perro y un gato en una tarea de clasificación con una ResNet50 con pesos orientados a realizar esta tarea.



(a) 1.ª capa convolucional



(b) 1.er bloque residual



(c) 2.º bloque residual

(d) 3.er bloque residual

(e) 4.º bloque residual

(f) Últ. capa convolucional

Figura 5.2: Ejemplo de mapa de calor con la librería Grad-CAM a partir de una imagen de un perro y un gato. Las zonas que más han influido en la activación del modelo se resaltan con colores cálidos, mientras que las regiones menos relevantes se muestran con tonos fríos.

vez más relevante y de más alto nivel. En las capas iniciales, las activaciones resultan difusas y dispersas, enfocándose de forma poco consistente en detalles del fondo o fragmentos del rostro. Sin embargo, a medida que se avanza hacia capas o bloques residuales más profundos, la atención se concentra progresivamente en el gato y en elementos relevantes a la tarea de clasificación.

En base a estos resultados, se ha optado por emplear la última capa convolucional y el cuarto (último) bloque residual como puntos de extracción para los mapas de calor en las siguientes pruebas de visualización.

5.2. Resultados de Grad-CAM

Para analizar cómo interpreta el modelo las imágenes a nivel interno, se han seleccionado diversas imágenes y tripletas de los conjuntos Flickr y Serrano21, aplicando la visualización con Grad-CAM. Los resultados pueden observarse en la Figura 5.3, de los cuales se pueden extraer varias observaciones relevantes.

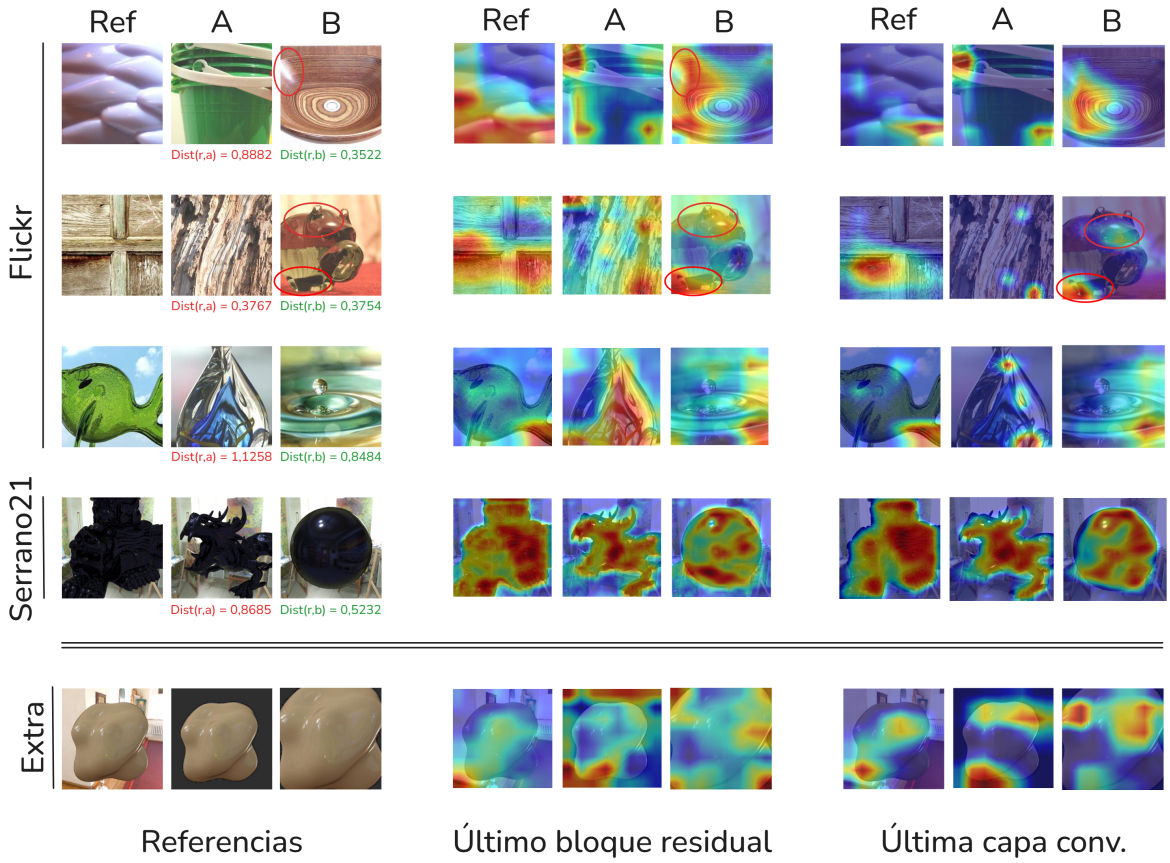


Figura 5.3: Ejemplos de activaciones con Grad-CAM. Las tres primeras filas son tripletas del *dataset* Flickr y la cuarta fila es una tripleta del *dataset* de Serrano21. Ambas cuentan con predicciones del modelo de Lagunas (la imagen con *Dist* verde es la elegida). La última fila es un *blob* del *dataset* Serrano21 en formato original, enmascarado y enmascarado + recortado.

En general podemos observar que al modelo de Lagunas presta especial atención a los reflejos y los brillos presentes en la imagen. Este es el caso del brillo de la madera barnizada de la primera fila, lo que parece haber motivado la elección de la opción B; del brillo del hipopótamo de la segunda fila, donde parece que se está fijando en la incidencia de la luz por la frente del hipopótamo y su salida por los pies; o de todas las imágenes de la tercera fila, donde el modelo parece estar fijándose en los puntos de incidencia de la luz o de mayor brillo de la imagen.

Por el contrario, también producen activaciones las zonas que destacan por su au-

sencia de luz. Este es el caso de la esquina inferior izquierda de la imagen de referencia de la primera fila, de las sombras provocadas por el relieve de la madera de la imagen de referencia de la segunda fila, o de la cola del pez de cristal y la parte central inferior de la flor de cristal de la tercera fila. Esto sugiere que el modelo también incorpora cierto contraste y profundidad como factores relevantes para su decisión.

En lo que respecta al conjunto Serrano21, podemos ver en la cuarta fila que únicamente la geometría de las imágenes produce activaciones y de forma muy satisfactoria, dejando a entender que el fondo no es relevante en este caso. Sin embargo, en la quinta fila, las activaciones no solo son débiles sobre la geometría principal, sino que se centran de forma excesiva en el fondo. Este efecto se agrava para el caso de los *blobs* con modificaciones, donde el modelo parece distraído por el entorno, evidenciando una sensibilidad no deseada a elementos irrelevantes al contexto. No obstante, su visualización con la última capa convolucional se centra bastante más en el *blob* para sus tres casos.

5.3. Resultados de PCA

Como alternativa a la visualización con Grad-CAM, se ha explorado el uso de Análisis de Componentes Principales (PCA) aplicado a las activaciones de la CNN. Esta técnica de reducción de dimensionalidad permite transformar las cientas de dimensiones del espacio de activaciones original en un nuevo espacio donde las primeras componentes capturan la mayor parte de la varianza de los datos. De esta forma, si se dibujan esas componentes principales como mapas de calor, se obtiene una visualización de las zonas que la red considera más relevantes para codificar la apariencia del material.

Los patrones observados en los resultados de la Figura 5.4 muestran parecidos con los obtenidos mediante Grad-CAM. En las filas con imágenes del dataset Flickr, se vuelve a destacar la sensibilidad del modelo a zonas con brillo y sombra: por ejemplo, la sombra en la parte inferior del pez de cristal (tercera fila), la sombra generada por el relieve de la madera (segunda fila), la zona sombreada del plástico izquierdo (primera fila) o el reflejo del barniz en la madera (primera fila).

Otra tendencia notable en el último bloque residual es que las activaciones de PCA tienden a concentrarse más, formando una única "mancha" conexa. Por el contrario, las activaciones de Grad-CAM son más dispersas y repartidas por distintas regiones de la imagen. Con la última capa convolucional de PCA, se mantiene la tendencia de la mancha conexa, pero las regiones activadas son claramente más grandes.

Otro aspecto destacable es que el PCA resalta de forma más precisa la geometría principal de la escena. En varios casos, las activaciones parecen *recortar* la silueta de los objetos, como sucede con el pez de cristal en la fila 3 o en todas las imágenes de Serrano21 en la fila 4. En la fila 5, el resultado es aceptable: las activaciones cubren buena parte del *blob* y también parte del suelo. Sin embargo, en la fila 6 el resultado es muy deficiente, ya que el objeto principal (el *blob*) aparece en azul —indicando baja activación— mientras que todo el fondo destaca en rojo, señalando que ha tenido más

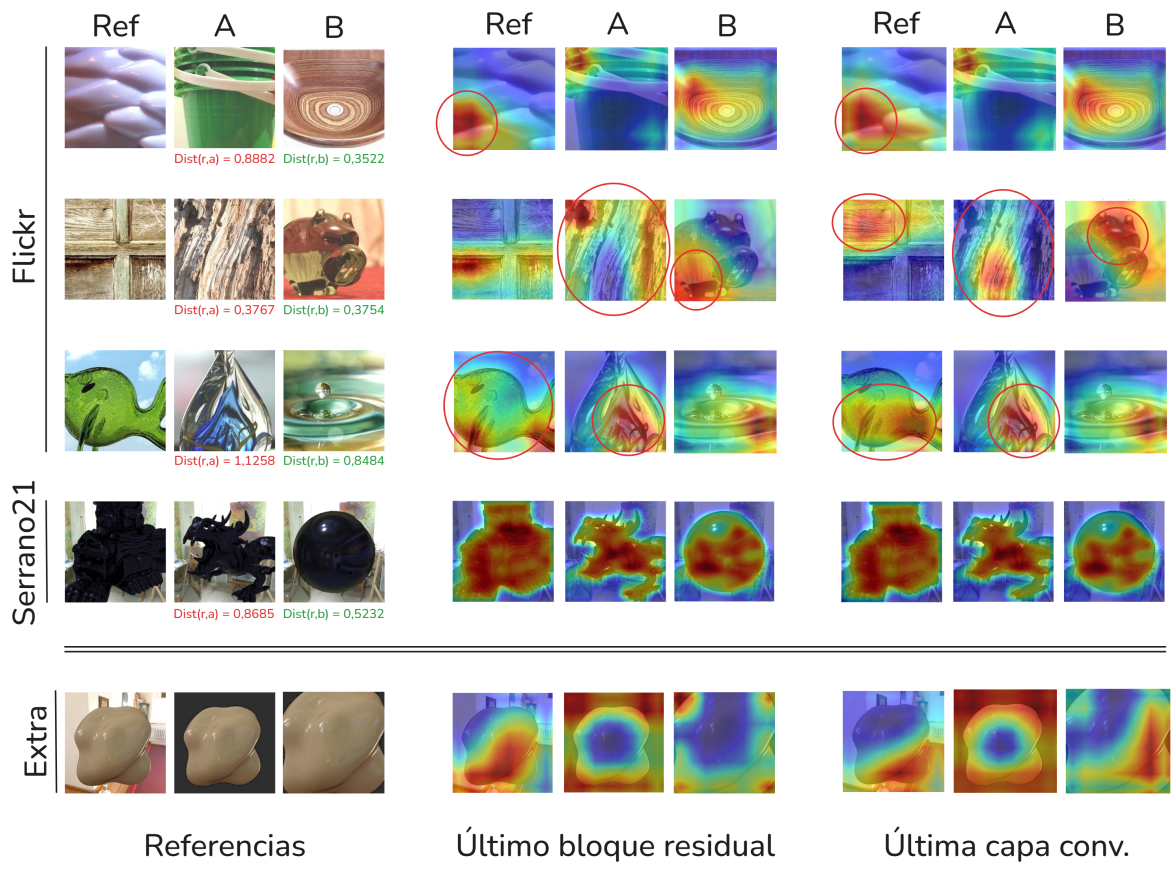


Figura 5.4: Ejemplos de activaciones con PCA. Las tres primeras filas son tripletas del *dataset* Flickr y la cuarta fila es una tripleta del *dataset* de Serrano21. Ambas cuentan con predicciones del modelo de Lagunas (la imagen con *Dist* verde es la elegida). La última fila es un *blob* del *dataset* Serrano21 en formato original, enmascarado y enmascarado + recortado.

influencia en la salida de la red, lo cual reafirma de nuevo una sensibilidad no deseada a elementos irrelevantes al contexto.

5.4. Comportamiento en función de la geometría y el material

Tras observar ciertos comportamientos indeseados en secciones anteriores —como activaciones centradas en el fondo en lugar de en la figura principal—, se ha decidido profundizar en cómo influyen dos factores clave en las visualizaciones internas del modelo: la geometría del objeto y el tipo de material aplicado. Para ello, se han realizado visualizaciones mediante Grad-CAM y PCA sobre las nueve geometrías del dataset Serrano21, usando dos materiales diferentes (uno difuso y otro especular) y fijando *Small Cathedral* como iluminación, dado que fue precisamente este fondo en el que se

dieron las problemáticas en las pruebas anteriores. Los resultados pueden consultarse en las Figuras 5.5a, 5.5b, 5.5c y 5.5d.

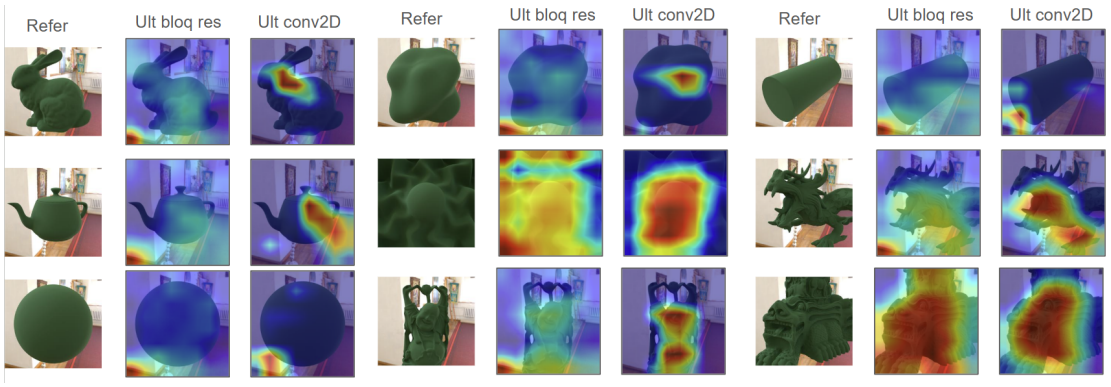
Analizando los resultados sobre las muestras difusas, se puede comprobar que el tipo de material tiene un impacto significativo en la forma de las activaciones: por ejemplo, el comportamiento observado con el *blob* en 5.5a difiere notablemente (y a peor) del observado previamente en 5.3, a pesar de usar la misma geometría y entorno.

Además, geometrías que anteriormente se visualizaban con contornos definidos —como *dragon*, *statuette* o *sphere*— aparecen ahora con activaciones más difusas y bordes menos precisos para los casos difusos, especialmente en los mapas de Grad-CAM.

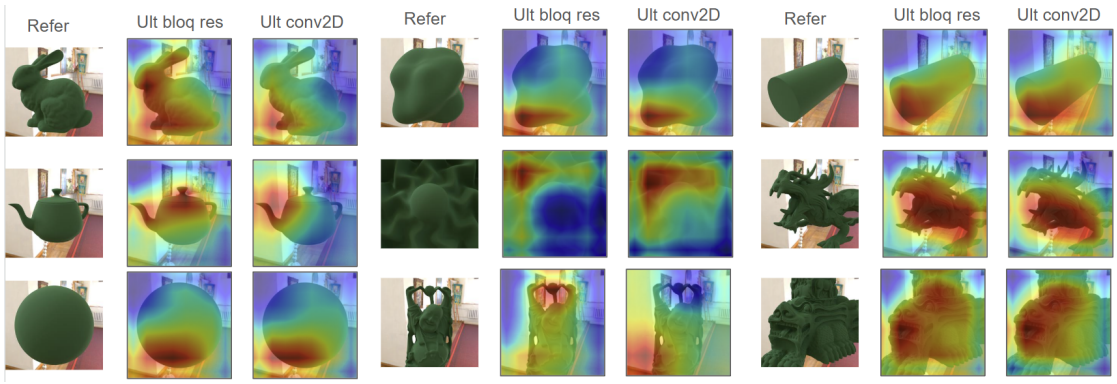
En el caso de los materiales especulares, las activaciones tienden a concentrarse en las zonas de la geometría donde se reflejan claramente elementos del entorno, lo cual es coherente con la sensibilidad del modelo a los patrones de luminancia y contraste. Esta tendencia se rompe únicamente en algunos casos concretos (por ejemplo, *statuette* con Grad-CAM y PCA o *dragon* y *buddha* con Grad-CAM), donde las activaciones se dispersan o pierden foco, probablemente debido a la complejidad de la escena.

Por otro lado, los materiales difusos muestran un rendimiento considerablemente más pobre con Grad-CAM, con excepción de las geometrías *Havran* y *Statuette*, en las que la atención se centra de forma más razonable en el objeto. En la mayoría del resto de geometrías —cuando se utiliza el último bloque residual—, el modelo tiende a focalizarse en el fondo, ignorando parcialmente o por completo la figura principal. Esta observación contrasta con los resultados obtenidos mediante PCA (Figura 5.5d), donde las regiones de mayor activación suelen coincidir con las propias figuras, lo que sugiere que PCA mantiene mejor la coherencia espacial con respecto a los elementos centrales de la escena.

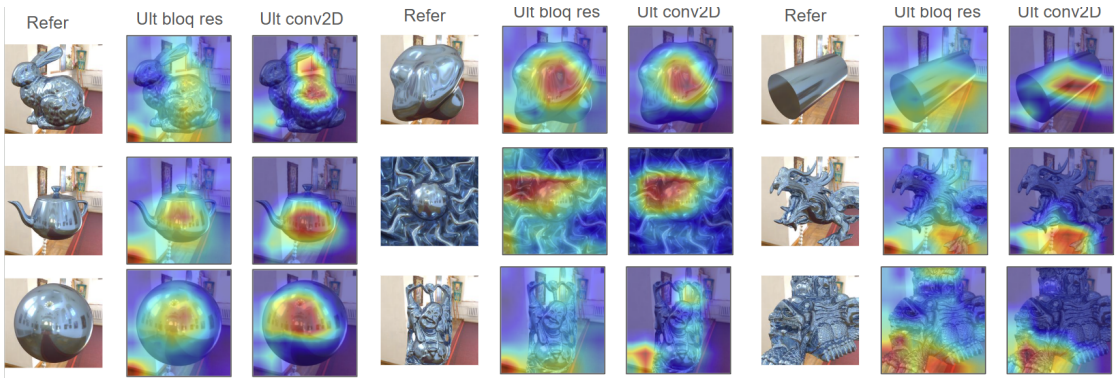
En conclusión, estos experimentos revelan que tanto la geometría como el tipo de material influyen de forma decisiva en la atención del modelo. En general, los materiales especulares producen activaciones más localizadas en las zonas con reflejos destacados, mientras que los difusos presentan un comportamiento más errático, especialmente con Grad-CAM. La técnica de PCA, en cambio, parece ofrecer una mayor estabilidad y robustez a estos factores, capturando de forma más fiable la estructura y presencia del objeto en la imagen, incluso bajo condiciones visuales complejas.



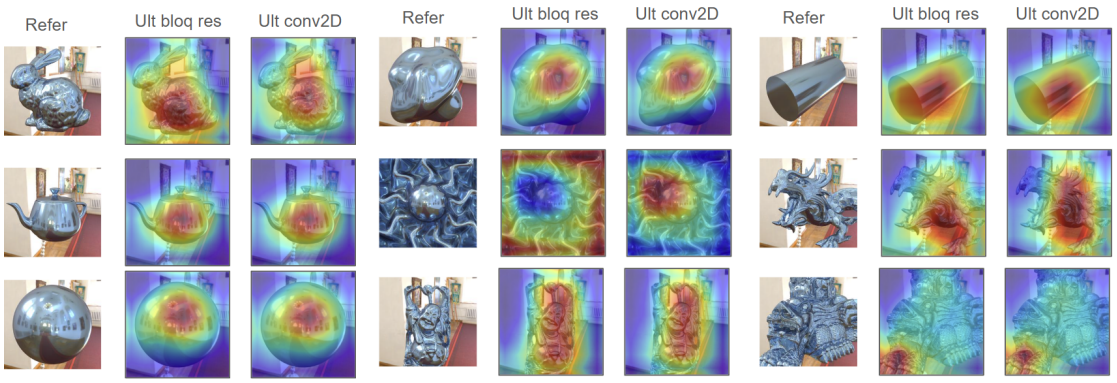
(a) Activaciones de Grad-CAM con las geometrías de Serrano21 y material difuso.



(b) Activaciones de PCA con las geometrías de Serrano21 y material difuso.



(c) Activaciones de Grad-CAM con las geometrías de Serrano21 y material especular.



(d) Activaciones de PCA con las geometrías de Serrano21 y material especular.

Figura 5.5: Visualización de las activaciones del modelo con Grad-CAM y PCA con todas las geometrías, entorno *Small Cathedral* y con materiales difuso (*UTIA_ISO-isotropic_m082_fabric119*) y especular (*RGL-chm_light_blue_rgb*).

Capítulo 6

Modificaciones al modelo base: Optimización de hiperparámetros y evaluación

Llegados a este punto, tras haber realizado múltiples evaluaciones y análisis visuales, se ha identificado un problema recurrente en el modelo de Lagunas: una falta de robustez ante variaciones en el fondo y una sensibilidad excesiva hacia las regiones periféricas de la imagen.

Con el objetivo de mitigar este problema, se ha modificado el proceso de entrenamiento de la red para forzar al modelo a ignorar el fondo, ya que este no pertenece al material estudiado. Concretamente, se ha añadido la funcionalidad de que las activaciones correspondientes al fondo se eliminan antes de la operación de *pooling*, es decir, antes de que la información sea comprimida en el *embedding* final. De este modo, se garantiza que el fondo no contribuya al aprendizaje del modelo ni influya en la salida generada.

Los datos de entrenamiento utilizados pertenecen al conjunto de datos de Lagunas19, por lo que todas las imágenes del conjunto de entrenamiento tienen las mismas dimensiones y se utilizan trece geometrías diferentes. A partir de estas imágenes, se han generado trece máscaras binarias que distinguen claramente la geometría del fondo. Para ello se ha empleado *MODNet* [Ke+22], una herramienta basada en visión por computador diseñada para la extracción de retratos en tiempo real, pero que ha demostrado ser igualmente eficaz para segmentar objetos como los presentes en esta tarea. El resultado obtenido es muy satisfactorio y se puede observar en la Figura 6.1.

Posteriormente, se ha reservado un 30 % del conjunto de entrenamiento original para validación y, para el entrenamiento, se ha implementado la aplicación a cada imagen de su correspondiente máscara binaria mediante una operación lógica AND, lo que permite eliminar el fondo de forma eficaz.

Con este planteamiento listo, se ha procedido a reentrenar el modelo. Para ello, se ha decantado por utilizar *Optuna*[1], una librería de optimización de hiperparámetros

[1]Más información en: `https://optuna.readthedocs.io/en/stable/`.

Figura 6.1: Imágenes sin fondo de todas las geometrías empleadas en el conjunto de entrenamiento de Lagunas19. Se usa el color verde para representar los píxeles transparentes.

automatizada. Se ha concedido un rango de valores de hiperparámetros con bastante libertad, como se muestra en la Tabla 6.1, con el fin de permitir una exploración más exhaustiva del espacio de búsqueda y facilitar la localización de un máximo global.

Hiperparámetro	**Optuna (rango)**	**Lagunas19**
Épocas	12	80
Tamaño del lote	8–64 (saltos de 8)	20
Tasa de aprendizaje	10^{-6} - 10^{-2}	10^{-3}
Decaimiento de pesos	10^{-6} - 10^{-2}	$5 \cdot 10^{-4}$
Momento	0,5 - 0,99	0,9
Margen de *Triplet loss*	0,1 - 1,0	0,3

Tabla 6.1: Comparación de hiperparámetros: Optuna vs configuración base de Lagunas et al. (2019)

El número máximo de épocas del estudio de Optuna se ha limitado a doce, ya que en el trabajo original de Lagunas et al. (2019) los mejores pesos se obtuvieron alrededor de la época seis. En cuanto al optimizador, se ha optado por utilizar *SGD* (Descenso de gradiente estocástico[2]), ya que en las pruebas iniciales es el que mejor rendimiento presentaba.

Para la búsqueda de hiperparámetros, se ha optado por el muestreador TPE (*Tree-structured Parzen Estimator*). Es un método de optimización bayesiana que aprende de los resultados de búsquedas anteriores para explorar de forma más eficiente el espacio de búsqueda. A diferencia de otros enfoques aleatorios, TPE tiene en cuenta los hiperparámetros que han dado buenos resultados y los que no, eligiendo así cada vez mejores combinaciones y priorizando aquellas con mayor probabilidad de mejorar el rendimiento del modelo.

Finalmente, es necesario definir un valor objetivo para guiar la optimización. En este caso, se ha elegido como valor objetivo el máximo *accuracy* obtenido a partir de la época cinco sobre el conjunto de validación. De esta forma, se evitan posibles sesgos

[2]Más información en: `https://en.wikipedia.org/w/index.php?title=Stochastic_gradient_descent&oldid=1295784793`.

derivados del subajuste característico de las primeras épocas y favorece la obtención de modelos más estables y generalizables. Con esta configuración, se ha ejecutado la optimización de hiperparámetros durante 370 iteraciones. A continuación, se detallan los resultados obtenidos.

6.1. Resultados del estudio de optimización de hiperparámetros

En primer lugar, en la Figura 6.2 se muestra la evolución de los valores objetivo obtenidos a lo largo de las 370 ejecuciones del estudio de optimización de hiperparámetros con Optuna. Es importante señalar que, a partir del intento número 97, todas las ejecuciones posteriores fueron interrumpidas prematuramente mediante el mecanismo de podado del optimizador, al considerar que no alcanzarían un rendimiento prometedor en términos de *accuracy* de validación.

El valor objetivo comienza en 0,695 y experimenta una subida escalonada hasta llegar a su máximo de 0,7776 en el intento 80. A partir de este punto —a poco de alcanzar una cuarta parte del total de ejecuciones— no se consigue superar dicho valor objetivo en ninguna prueba posterior. Este comportamiento, junto con el elevado número de ejecuciones podadas, podría indicar que el optimizador ha convergido hacia un máximo local.

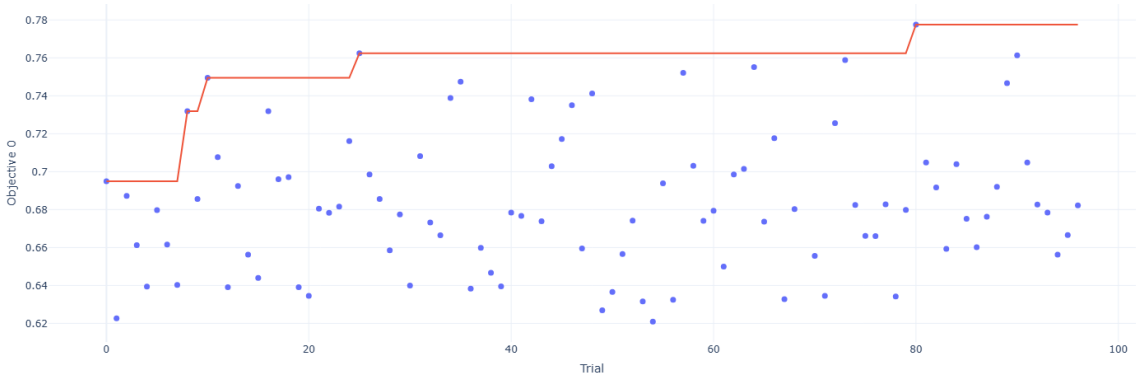


Figura 6.2: Diagrama de historial de optimización de Optuna. El eje Y indica el valor objetivo alcanzado por el modelo de red neuronal del intento correspondiente en el eje X.

Los hiperparámetros correspondientes al mejor modelo obtenido se muestran en la Tabla 6.2. A pesar de las diferencias con respecto a los utilizados en el modelo original de Lagunas, resultan coherentes desde el punto de vista técnico. Por ejemplo, el incremento en el tamaño de *batch* va acompañado de una mayor tasa de aprendizaje, lo cual es razonable dado que al procesar más muestras por iteración, el modelo puede realizar ajustes más amplios.

Hiperparámetro	Valor óptimo
Tamaño de *batch*	40
Margen de *Triplet loss*	0,5819
Momento	0,8863
Decaimiento de pesos	$1,1294 \cdot 10^{-6}$
Tasa de aprendizaje	$7,8629 \cdot 10^{-3}$

Tabla 6.2: Mejores hiperparámetros encontrados por Optuna

Pasando a la siguiente visualización (Figura 6.3), se trata de una gráfica de relaciones paralelas que permite analizar cómo interactúan entre sí los distintos hiperparámetros de alta dimensionalidad evaluados por Optuna. En ella se observan las configuraciones exploradas durante la optimización, así como las combinaciones que conducen a un mejor rendimiento.

Puede apreciarse que el tamaño del *batch* se distribuye de forma uniforme a lo largo de todos los valores posibles. La tasa de aprendizaje, por otro lado, muestra una clara preferencia por los valores altos (aproximadamente entre 0,005 y 0,00998) dentro del rango considerado. El momento[3] medio-alto (entre 0,75 y 0,9) es el intervalo de valores predominante y el margen de la *Triplet Loss* (correspondiente al μ de la Ecuación 2.2) tiene una preferencia por los valores medios (entre 0,6 y 0,65) y bajos (entre 0,2 y 0,5). Del mismo modo, el decaimiento de pesos del modelo favorece los valores medio-bajos.

Pasando al análisis de la importancia de cada hiperparámetro en el rendimiento del modelo, según el evaluador *fANOVA* [HHL14], elegido por su especialización en espacios de búsqueda no lineales y consideración de la varianza de los hiperparámetros, los más determinantes son: la tasa de aprendizaje, con un peso del 62 %, el momento del SGD, con un 19 % y el margen de *Triplet loss*, con un 14 %. Por el contrario, el resto de parámetros (tamaño del lote y decaimiento de pesos) tienen una importancia prácticamente marginal, sumando apenas un 6 % en total.

Por último, se ha obtenido la función de distribución empírica (EDF) de los 370 intentos del estudio de Optuna. En ella se observa que aproximadamente el 50 % de las ejecuciones superan un valor objetivo de 0,68, lo que indica una frecuencia considerable de modelos con un rendimiento aceptable. Sin embargo, solo el 20 % logra superar el umbral de 0,71, y únicamente un reducido 5 % alcanza valores por encima de 0,76. Estos datos sugieren que obtener un modelo con rendimiento destacado requiere una configuración de hiperparámetros muy específica y poco frecuente dentro del espacio de búsqueda, lo cual refuerza la importancia de contar con herramientas de optimización como Optuna.

[3]El momento del *SGD* es un término que ayuda a acelerar el descenso del gradiente acumulando gradientes pasados, lo que permite avanzar más rápido en direcciones consistentes y reducir oscilaciones.

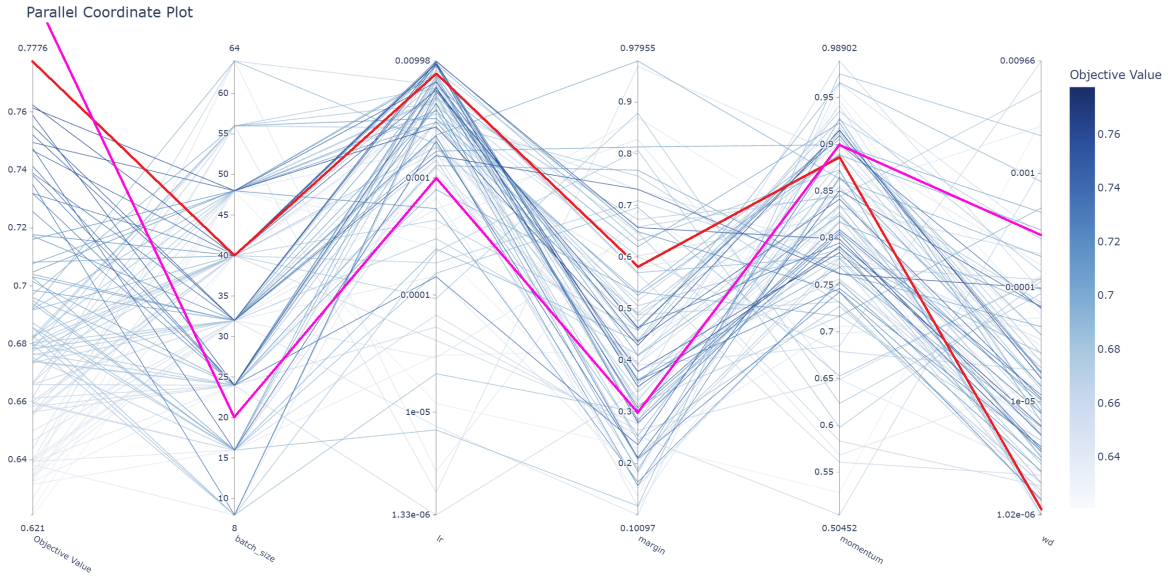


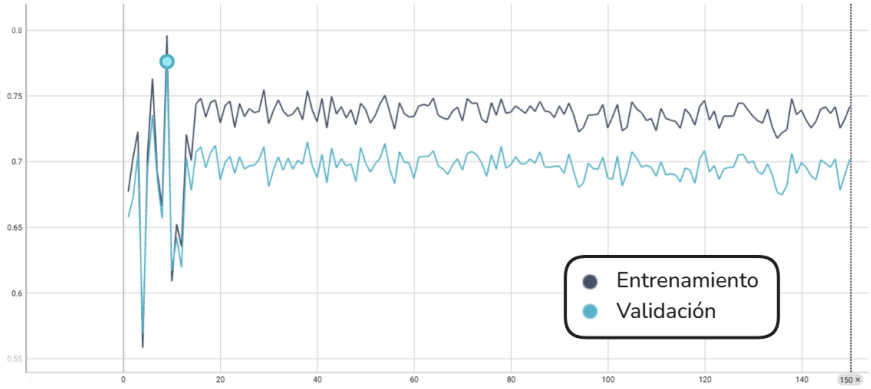
Figura 6.3: Gráfica de relaciones paralelas de hiperparámetros de alta dimensionalidad de Optuna. Las líneas rojas indican los hiperparámetros del mejor modelo obtenido en la optimización. Las líneas fucsia corresponden con los hiperparámetros del mejor modelo que obtuvo Lagunas en su artículo.

6.2. Modelo modificado

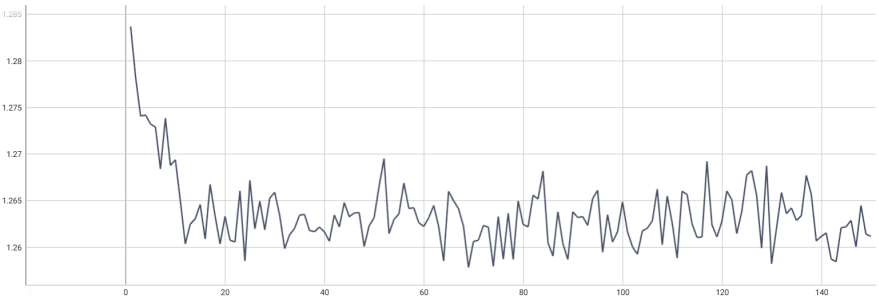
En la Tabla 6.2 se han presentado los parámetros óptimos encontrados mediante la optimización con Optuna, cuyo valor objetivo alcanzado ha sido de 0,7776 en la época nueve. Posteriormente, se ha llevado a cabo el entrenamiento completo del modelo durante 150 épocas, dando lugar al modelo final modificado, el cual alcanza un *accuracy* máximo en test de 0,7639 para el conjunto de test de Lagunas19. Esto se debe a que el conjunto de validación (el que se ha usado para el valor objetivo) son tripletas aleatorias que se han extraído del conjunto de entrenamiento (luego algunas geometrías coincidirán), mientras que el conjunto de test está conformado por la geometría Havran, inédita para el modelo en entrenamiento.

Al analizar la evolución del *accuracy* durante el entrenamiento (Figura 6.4a), se observa que tras alcanzar el pico de 0,796 en la época nueve, ya no se vuelve a superar dicho valor. Este patrón sugiere que el modelo alcanza su mejor desempeño temprano en el proceso de entrenamiento, ya que coincide con la estabilización de la gráfica de pérdida (Figura 6.4b) alrededor de valores del intervalo [1,258; 1,269]. Además, tras alcanzar este punto máximo, la precisión tiene un desplome importante, tras el cual se mantiene constante sin mostrar incrementos adicionales. Esto refleja que el modelo se ha estabilizado y no va a haber mejoría aparente, lo cual podría ser un indicio de sobreajuste o bien de una limitación en la capacidad del modelo para seguir aprendiendo con el conjunto de datos utilizado.

El objetivo principal de este nuevo entrenamiento es reducir la sensibilidad del modelo con el fondo de las imágenes. Para evaluar este aspecto, se ha realizado el



(a) Gráfica del *accuracy* en entrenamiento y validación durante el entrenamiento del modelo a optimizar con Optuna. El punto azul indica el *accuracy* máximo en validación (0,7776 en la época 9).



(b) Gráfica de la función de pérdida durante el entrenamiento del modelo modificado.

Figura 6.4: Gráficas de *accuracy* y de la función de pérdida del modelo modificado con la mejor configuración de hiperparámetros encontrada con Optuna.

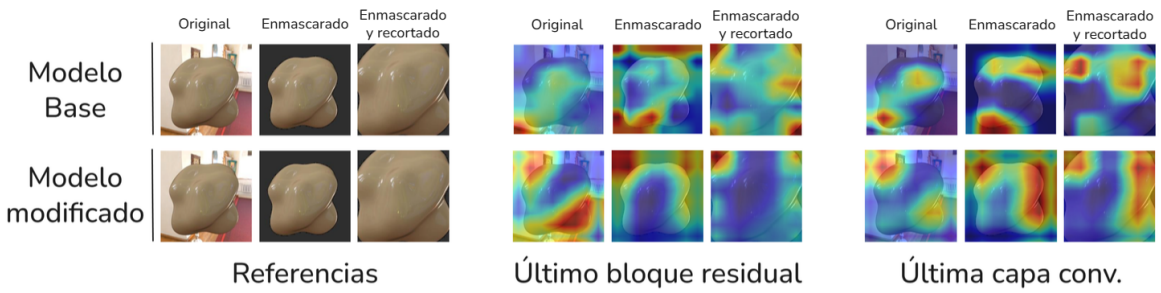


Figura 6.5: Comparación de la visualización con PCA del modelo base de Lagunas vs modelo modificado con pesos obtenidos tras la optimización con Optuna.

test de robustez (Tabla 6.3a) y, aunque los resultados finalmente no superan a los del modelo base, este análisis ha ayudado a entender mejor cómo responden las activaciones internas ante cambios del fondo. De hecho, las visualizaciones con PCA (Figura 6.5) revelan una mejoría del modelo en ciertos casos concretos (como es el caso de la

imagen original del último bloque residual), atendiendo menos a las zonas del fondo y focalizando las actividades en áreas de más importancia para determinar la apariencia del material.

Geometría	Modelo base	Modelo modificado
sphere	15,34 %	24,23 %
bunny	15,61 %	31,95 %
cylinder	22,32 %	35,47 %
teapot	19,33 %	39,78 %
blob	18,05 %	33,43 %
havran	0,00 %	0,00 %
buddha	24,47 %	32,33 %
dragon	22,16 %	34,99 %
statuette	12,10 %	15,29 %
Global	**16,57 %**	**27,51 %**

a) Tabla que indica el número de veces que el modelo ha cambiado de opinión al enmascarar el fondo de las mismas tripletas a gris.

Geometría	Modelo base	Modelo modificado
sphere	68,40 %	59,82 %
bunny	71,97 %	64,97 %
cylinder	62,08 %	61,16 %
teapot	66,67 %	57,98 %
blob	69,23 %	56,80 %
havran	64,00 %	61,14 %
buddha	62,84 %	59,21 %
dragon	62,39 %	57,14 %
statuette	59,55 %	60,51 %
Global	**65,23 %**	**59,80 %**

b) Tabla con resultados del test de similitud del modelo optimizado vs modelo base de Lagunas.

Tabla 6.3: Tablas de test de robustez (a) y similaridad (b) de 3.000 tripletas de Serrano21 (dada una tripleta, solo varía el material) donde se compara el rendimiento del modelo optimizado de Optuna vs modelo base de Lagunas.

Adicionalmente, el modelo modificado ha sido evaluado utilizando un conjunto de 3.000 tripletas del *dataset* Serrano21 (similar a la Tabla 4.11), donde solo varía el material. En esta prueba (con los resultados en la Figura 6.3b), el modelo optimizado alcanza un 59,8 % de *accuracy*, un resultado ligeramente más limitado respecto al modelo original. No obstante, el rendimiento se mantiene competitivo, con una capacidad razonable de generalización a nuevos escenarios sintéticos.

En definitiva, el estudio de optimización con Optuna ha permitido encontrar una configuración de hiperparámetros que maximiza el rendimiento sobre el conjunto de validación, alcanzando una precisión máxima en test de 0,7639. Aunque no ha superado al modelo base en todos los escenarios, los resultados obtenidos revelan aspectos clave del comportamiento del modelo ante distintos tipos de transformaciones. Este análisis abre la puerta a futuras líneas de mejora que podrían incluir enfoques más robustos de enmascarado o el uso de estrategias de aprendizaje más específicas, todo ello con el objetivo de avanzar hacia modelos perceptuales más sólidos y generalizables.

Capítulo 7

Conclusiones

El principal objetivo de este Trabajo de Fin de Grado ha sido analizar en profundidad el modelo de similitud perceptual propuesto por Lagunas et al. (2019), evaluando sus capacidades de generalización y robustez en distintos contextos, tanto sintéticos como realistas. Para ello, se han abordado múltiples enfoques complementarios: desde el análisis cuantitativo con nuevas métricas y desgloses estadísticos, hasta visualizaciones detalladas de las activaciones internas del modelo mediante técnicas como Grad-CAM y PCA.

Uno de los logros más destacables ha sido la recopilación y el estudio sistemático de resultados que el artículo original no profundizaba, como el comportamiento del modelo frente a tripletas con niveles de ambigüedad variables, o la influencia del fondo y la iluminación en la predicción de similitud. Asimismo, se ha evaluado el modelo con datos completamente nuevos —como el *dataset* de Flickr y tripletas generadas a partir del conjunto de datos de Serrano21—, permitiendo comprobar la capacidad del modelo para enfrentarse a situaciones no vistas durante su entrenamiento.

Adicionalmente, se ha llevado a cabo un proceso de optimización de hiperparámetros con la herramienta Optuna, en combinación con una propuesta de entrenamiento que excluye el fondo de las imágenes, con el objetivo de mejorar la robustez del modelo. Aunque el modelo resultante no ha superado al original en términos de robustez, este proceso ha permitido evaluar en detalle cómo influyen distintas configuraciones de entrenamiento sobre el rendimiento general y ha ofrecido información útil sobre las limitaciones actuales del enfoque y sobre qué aspectos merecen una revisión más profunda en trabajos futuros.

7.1. Trabajo futuro y limitaciones

A pesar del alcance logrado, este trabajo también presenta ciertas limitaciones y abre diversas líneas de investigación futuras. Una de las limitaciones es la influencia de la iluminación en las predicciones del modelo. Como se observa en el Apartado 4.4.3, los mejores resultados se obtienen en entornos con luminancia media o baja, lo que parece estar relacionado con las condiciones del conjunto de entrenamiento y

las transformaciones utilizadas. Como posible mejora, se podría aumentar la variedad de entornos de iluminación, modificar las transformaciones de aumentación de datos o experimentar con distintos *tonemappers* para reforzar la generalización del modelo ante cambios en la apariencia de la escena.

Además, aunque se ha trabajado con transformaciones simples como enmascarado y recorte, podrían explorarse variaciones más complejas (como cambios de escala, oclusiones o iluminación adversa) para evaluar la robustez del modelo con mayor profundidad. También sería interesante estudiar el impacto del rendimiento de la red ante cambios en la arquitectura.

Particularmente, respecto al tratamiento del fondo, aunque se ha introducido una propuesta que filtra sus activaciones, los resultados indican que el modelo sigue siendo sensible a este tipo de información. Por ello, podrían explorarse estrategias más efectivas para centrar la atención sobre la geometría o el material mediante, por ejemplo, modificaciones arquitectónicas como mecanismos de atención espacial (e.g. CBAM [Woo+18]).

Por último, se podría combinar la métrica de similitud con otras características de la apariencia del material, como rugosidad o translucidez, para manejar representaciones más completas. Integrar predictores de atributos, como los propuestos en trabajos recientes del estado del arte, podría mejorar la generalización del modelo y hacerlo más interpretable. Así, se aumentaría la completitud de las características de los materiales y se podría conseguir una percepción de similitud más fiel a la de los humanos.

En definitiva, este TFG ha demostrado que el análisis detallado de un modelo de similaridad de apariencia de los materiales ofrece múltiples oportunidades de mejora y revela aspectos críticos que no siempre son visibles con simples métricas agregadas. Las bases sentadas en este trabajo abren la puerta a mejoras sustanciales en modelos futuros, tanto en precisión como en capacidad de generalización.

Bibliografía

[Mar+00] Stephen Marschner et al. "Image-based bidirectional reflectance distribution function measurement". En: *Applied Optics* 39 (jun. de 2000), págs. 2592-2600.

[Ade01] Edward H. Adelson. "On seeing stuff: the perception of materials by humans and machines". En: *Proceedings of the SPIE* 4299 (2001), págs. 1-12.

[Mat+03] W. Matusik et al. "A Data-Driven Reflectance Model". En: *ACM Transactions on Graphics (TOG)* 22.3 (jul. de 2003), págs. 759-769. URL: `https://www.merl.com/publications/TR2003-83`.

[Wan+04] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". En: *IEEE Transactions on Image Processing* 13.4 (2004), págs. 600-612.

[Liu+10] Ce Liu et al. "Exploring features in a Bayesian framework for material recognition". En: *CVPR*. 2010, págs. 239-246.

[MB10] Laurence T. Maloney y David H. Brainard. "Color and material perception: Achievements and challenges". En: *Journal of Vision* 10.9 (2010), págs. 1-6.

[And11] Blake A. Anderson. "A framework for understanding the visual perception of materials". En: *Current Opinion in Neurobiology* 21.4 (2011), págs. 589-595.

[Den12] Li Deng. "The mnist database of handwritten digit images for machine learning research". En: *IEEE Signal Processing Magazine* 29.6 (2012), págs. 141-142.

[FFG12] Adrià Forés, James Ferwerda y Jinwei Gu. "Toward a Perceptually Based Metric for BRDF Modeling". En: *Final Program and Proceedings - IS and T/SID Color Imaging Conference* CIC'12 (ene. de 2012), págs. 142-148.

[PR12] Pierre Poulin y Holly Rushmeier. "Material Appearance". En: *IEEE Computer Graphics and Applications* 32 (mar. de 2012), págs. 22-23.

[PJD13] Narayan Pisharoty, Manisha Jadhav y Yogesh Dandawate. "Performance Evaluation of Structural Similarity Index Metric in Different Colorspaces for HVS Based Assessment of Quality of Colour Images". En: *International Journal of Engineering and Technology* 5 (abr. de 2013), págs. 1555-1562.

[FV14] Jiří Filip y Radomír Vavra. "Template-Based Sampling of Anisotropic BRDFs". En: *Computer Graphics Forum* 33 (oct. de 2014), págs. 91-99.

[Fle14] Roland W. Fleming. "Visual perception of materials and their properties". En: *Vision Research* 94 (2014), págs. 62-75.

[HHL14] Frank Hutter, Holger Hoos y Kevin Leyton-Brown. "An Efficient Approach for Assessing Hyperparameter Importance". En: *Proceedings of the 31st International Conference on Machine Learning*. Vol. 32. 1. Jun. de 2014, págs. 754-762.

[Fil15] Jiří Filip. "Analyzing and Predicting Anisotropic Effects of BRDFs". En: *ACM Symposium on Applied Perception* (sep. de 2015), págs. 25-32.

[HFM16] V. Havran, J. Filip y K. Myszkowski. "Perceptually Motivated BRDF Comparison using Single Image". En: *Comput. Graph. Forum* 35.4 (jul. de 2016), págs. 1-12.

[He+16] Kaiming He et al. "Deep Residual Learning for Image Recognition". En: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, págs. 770-778.

[SYG16] Ana Serrano, Zerrin Yumak y Diego Gutierrez. "Edit propagation using segmentation-aware edge detection". En: *Computer Graphics Forum*. Vol. 35. 2. 2016, págs. 1-10.

[HB17] Mohammed Hassan y Mazen Bashraheel. "Quality Assessment from Grayscale to Color Images". En: *World Journal of Computer Application and Technology* 5 (dic. de 2017), págs. 56-64.

[DJ18] Jonathan Dupuy y Wenzel Jakob. "An Adaptive Parameterization for Efficient Material Acquisition and Rendering". En: *Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 37.6 (nov. de 2018), 274:1-274:18.

[McI+18] Leland McInnes et al. "UMAP: Uniform Manifold Approximation and Projection". En: *Journal of Open Source Software* 3.29 (2018), pág. 861.

[SJR18] Tiancheng Sun, Henrik Jensen y Ravi Ramamoorthi. "Connecting measured BRDFs to analytic BRDFs by data-driven diffuse-specular separation". En: *ACM Transactions on Graphics*. Vol. 37. Dic. de 2018, págs. 1-15.

[Woo+18] Sanghyun Woo et al. "CBAM: Convolutional Block Attention Module". En: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sep. de 2018.

[Zha+18] Richard Zhang et al. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". En: *CVPR*. 2018.

[Fey+19] Jean Feydy et al. "Interpolating between Optimal Transport and MMD using Sinkhorn Divergences". En: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. Vol. 89. PMLR, abr. de 2019, págs. 2681-2690.

[Lag+19] Manuel Lagunas et al. "A Similarity Measure for Material Appearance". En: *ACM Transactions on Graphics (SIGGRAPH 2019)* 38.4 (2019).

[Gc21] Jacob Gildenblat y contributors. *PyTorch library for CAM methods*. `https://github.com/jacobgil/pytorch-grad-cam`. 2021.

[Lav+21] Guillaume Lavoué et al. "Perceptual quality of BRDF approximations: dataset and metrics". En: *Computer Graphics Forum* 40.2 (2021), págs. 327-338.

[Ser+21] Ana Serrano et al. "The effect of shape and illumination on material perception: model and applications". En: *ACM Trans. on Graph.* 40.4 (2021).

[Ven+21] Abhinau K. Venkataramanan et al. "A Hitchhiker's Guide to Structural Similarity". En: *IEEE Access* 9 (2021), págs. 28872-28896.

[Del+22] Johanna Delanoy et al. "A Generative Framework for Image-based Editing of Material Appearance using Perceptual Attributes". En: *Computer Graphics Forum* 41.1 (2022), págs. 453-464.

[Ke+22] Zhanghan Ke et al. "MODNet: Real-Time Trimap-Free Portrait Matting via Objective Decomposition". En: *AAAI*. 2022.

[Nep22] Neptune AI. *PyTorch Loss Functions: The Ultimate Guide*. Último acceso: 03-06-2025. 2022. URL: `https://neptune.ai/blog/pytorch-loss-functions` (visitado 03-06-2025).

[Ber23] Simone Berasi. "CCW-SSIM: a Novel Color Enhanced Structural Similarity Metric applied to Autoencoders for Anomaly Detection". Laurea Magistrale thesis. Politecnico di Milano, jul. de 2023. URL: `https://www.politesi.polimi.it/retrieve/da295bf9-2556-40e2-a11a-ffdf6f299767/2023_07_Berasi_Executive%20Summary_02.pdf`.

[Gue+24] Julia Guerrero-Viu et al. "Predicting Perceived Gloss: Do Weak Labels Suffice?" En: *Computer Graphics Forum* 43.2 (2024), e15037.

[IBM24] IBM. *What are Convolutional Neural Networks?* Último acceso: 05-06-2025. 2024. URL: `https://www.ibm.com/think/topics/convolutional-neural-networks`.

[LJY25] Fei-Fei Li, Justin Johnson y Serena Yeung. *Neural Networks: Part 1*. Último acceso: 05-06-2025. 2025. URL: `https://cs231n.github.io/neural-networks-1/`.

Lista de figuras

Lista de tablas

Anexo A

Terminología

A continuación se listan una serie de términos, conceptos y acrónimos usados a lo largo de la memoria, junto a sus traducciones, sinónimos, forma extendida o breve explicación.

Término o Acrónimo	**Término equivalente o forma extendida**
Dataset	Conjunto de datos
Deep learning	Aprendizaje profundo
Loss function	Función de pérdida
Epoch	Época
Batch	Lote de datos
Cluster	Agrupación
MAE	Mean Absolute Error
MSE	Mean Squared Error
CNN	Red neuronal convolucional
Métrica BRDF-based	Métrica basada en BRDF
Albedo	Color puro de un material (sin efectos visuales, reflejos, etc.)
Ground truth	Datos de referencia para evaluar o comparar que se consideran verdad
Embedding	Representación vectorial de una entrada en un espacio de menor dimensión

Anexo B

Fundamentos de redes neuronales

B.1. Redes neuronales

Una red neuronal es un modelo computacional inspirado en la estructura y el funcionamiento del cerebro humano (Figura B.1). Su objetivo es aprender a resolver tareas complejas, como clasificar imágenes, traducir texto o, como en el caso de este trabajo, estimar la similitud visual entre materiales. Estas redes están formadas por unidades básicas llamadas nodos o neuronas artificiales, que se organizan en capas y se conectan entre sí para transformar una entrada en una salida mediante un proceso de aprendizaje.

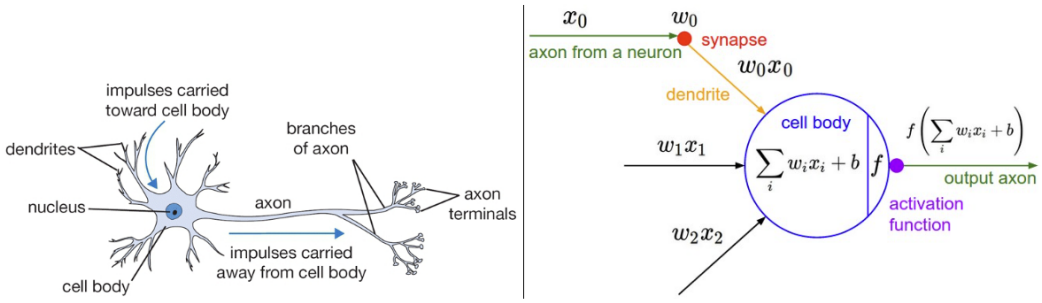


Figura B.1: Una neurona biológica (izquierda) y una neurona artificial representada con un esquema matemático (derecha) [LJY25].

Tal y como se puede observar en la Figura B.1, una neurona artificial se inspira en el funcionamiento de una neurona biológica. Una neurona artificial recibe múltiples entradas (x_0, x_1, x_2), cada una de las cuales se multiplica por un peso asociado (w_0, w_1, w_2) y a los productos se les suma un sesgo (*bias*). A esa salida de la neurona (la suma ponderada más el sesgo) se le aplica una función de activación f, que determina la salida final de la neurona antes de transmitirse al siguiente nivel de la red.

Esta función de activación es una transformación no lineal cuyo objetivo es que la red pueda aprender relaciones y patrones complejos que no podrían capturarse con funciones lineales. Para ello, adapta los valores resultantes a un rango que tenga sentido en el contexto del problema a tratar.

Por ejemplo, una neurona podría recibir entradas como "velocidad del viento", "dirección del viento" y "temperatura". Cada una se multiplica por un peso, se suma un sesgo y se pasa por una función de activación $ReLU$[1], que deja pasar solo los valores positivos y convierte en cero los negativos. Esto podría servir para decidir si activar una alerta de condiciones adversas, donde solo las señales fuertes y positivas disparan la respuesta.

Para construir una red neuronal artificial, estas neuronas se organizan en tres tipos de capas que procesan la información de forma secuencial (ver ejemplo de la Figura B.2). La primera de ellas es la capa de entrada, que recibe los propios datos de entrada de la red (por ejemplo, los píxeles de una imagen o las características de un conjunto de datos). A continuación, se encuentran las capas ocultas (opcionales), que son responsables de transformar progresivamente la información mediante operaciones lineales (producto escalar con pesos y suma de sesgos) y funciones de activación. El número y tamaño de estas capas determinan la capacidad del modelo para aprender representaciones complejas. Por último, la capa de salida proporciona la predicción o resultado de la red, y su estructura depende del tipo de tarea.

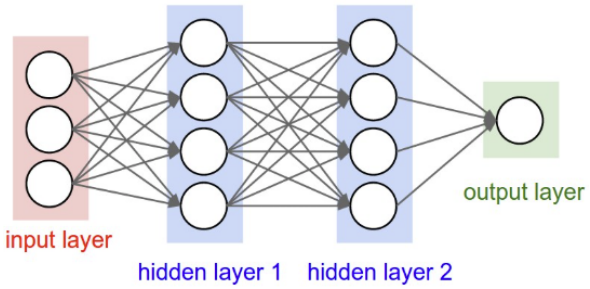


Figura B.2: Dibujo de una red neuronal de tres capas [LJY25]. Tiene tres entradas, dos capas ocultas de cuatro neuronas cada una y una capa de salida. Nótese que hay conexiones (sinapsis) entre neuronas de diferentes capas, pero no dentro de una misma capa.

Finalmente, una vez que se tiene la red neuronal lista, se procede a entrenar la red con los datos de entrenamiento a lo largo de múltiples iteraciones o épocas (en inglés, *epoch*). En cada época, la red procesa todo el conjunto de datos, ajustando sus pesos mediante un algoritmo de optimización (e.g., descenso del gradiente) con el objetivo de minimizar una función de error. A medida que se avanzan las épocas, la red va aprendiendo progresivamente los patrones presentes en los datos. Sin embargo, un número excesivo de épocas puede llevar al modelo a memorizar los datos en lugar de generalizar; es lo que se llama sobreajuste u *overfitting*. Por otro lado, también se puede dar el caso contrario: un número insuficiente puede hacer que el modelo no aprenda lo suficiente; es lo que se llama subajuste o *underfitting*. Además del número de épocas, factores como la complejidad de la red o la cantidad y diversidad de datos disponibles también influyen significativamente en estos fenómenos.

[1] `https://www.geeksforgeeks.org/relu-activation-function-in-deep-learning/`

B.2. Función de pérdida

Las funciones de pérdida se usan para medir el error entre la salida predicha por el modelo (en este TFG, el modelo es una red neuronal convolucional) y la salida objetivo esperada. En esencia, nos están diciendo cuán lejos está el modelo de acertar la respuesta correcta.

En la Figura B.3 se ilustra el cálculo de la función de pérdida. Sea la función de pérdida J que tiene 2 parámetros de entrada: y (salida objetivo) y $\hat{y}$ (salida predicha por el modelo).

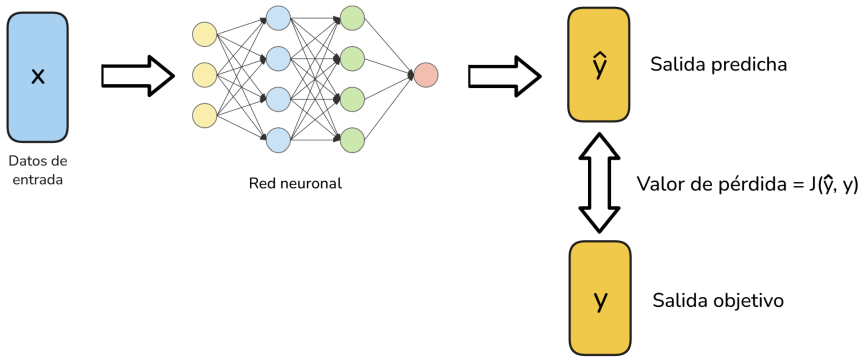


Figura B.3: Ilustración de la función de pérdida de una red neuronal [Nep22]

Esta función determinará el rendimiento del modelo comparando su salida predicha ($\hat{y}$) con la salida esperada (y). Por lo tanto, si estos dos parámetros distan mucho entre sí, el valor de la función de pérdida será muy alto. Por el contrario, si se trata de parámetros cercanos o casi idénticos, el valor será muy bajo.

Las funciones de pérdida cambian en base al problema que está tratando de resolver nuestro modelo. Algunos ejemplos de funciones son las siguientes:

- *Mean Squared Error Loss (L2)*: es la media de las N distancias al cuadrado entre el valor objetivo (y) y el valor predicho ($\hat{y}$). Penaliza los errores grandes y favorece los errores pequeños. Comúnmente, suele ser la opción por defecto.

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{B.1}$$

- *Margin Ranking Loss*: se emplea para comparar pares de valores de la entrada (x_1 y x_2) y reforzar una relación de orden entre ellos. El signo y puede tomar los valores -1 o 1. Si es 1, la primera entrada (x_1) tiene un valor mayor y, por tanto, es de mayor ranking que la segunda entrada. En caso de que sea -1, ocurre exactamente lo contrario. Por ello, esta función penaliza cuando la diferencia entre x_1 y x_2 no supera un margen m.

$$\mathcal{L}_{\text{ranking}} = \max\left(-y \cdot (x_1 - x_2) + m,\, 0\right) \tag{B.2}$$

- *Triplet Margin Loss*: se basa en grupos de tres elementos (ancla, positivo, negativo) y busca que el positivo esté más cerca del ancla que el negativo, por al menos un margen μ. Es muy utilizada en problemas de aprendizaje por similitud o embeddings.

$$\mathcal{L}_{\text{triplet}} = \max\left(distancia(ancla, positivo) - distancia(ancla, negativo) + \mu,\ 0\right) \tag{B.3}$$

B.3. Redes neuronales convolucionales

Un caso particular de las redes neuronales son las redes neuronales convolucionales (*CNN*, por sus siglas en inglés). Estas redes surgieron para abordar el problema de escalabilidad que presentan las redes neuronales tradicionales al trabajar con imágenes, cuya alta dimensionalidad supone un desafío computacional importante. Por este motivo, las CNN están especialmente diseñadas para procesar imágenes y la principal diferenciación reside en la arquitectura que tienen.

En una CNN, las neuronas se organizan en volúmenes tridimensionales con dimensiones de altura, anchura y profundidad. Esto permite que no todas las unidades estén conectadas entre sí, reduciendo así el número de parámetros y facilitando la generalización [LJY25]. Estas redes están compuestas principalmente por tres tipos de capas:

- **Capas convolucionales**: aplican filtros sobre la entrada para detectar patrones locales como bordes, texturas o formas. A medida que se avanza en profundidad en la red, estas características se combinan para identificar estructuras más complejas (ver Figura B.4). Tras cada convolución, se suele aplicar una función de activación no lineal como ReLU, que introduce no linealidad al modelo y mejora su capacidad de aprendizaje.

- **Capas de agrupamiento (*pooling*)**: reducen la dimensionalidad de los mapas de características generados, manteniendo la información más relevante. Esto disminuye la complejidad computacional y ayuda a evitar el sobreajuste.

- **Capas completamente conectadas (*fully-connected*, FC)**: situadas al final de la red, toman las características extraídas por las capas anteriores y generan la salida final del modelo, como una clasificación o una predicción.

Un ejemplo de red neuronal convolucional puede verse en la figura B.5. En este caso, la entrada es una imagen monocromática de un "2" manuscrito[2]. La imagen pasa primero por una capa convolucional que genera un volumen de activaciones de tamaño 28×28×12. A continuación, se aplica una función de activación ReLU, seguida de una capa de pooling que reduce la dimensionalidad a 14×14×12. Finalmente, el volumen se aplana y se introduce en dos capas totalmente conectadas (fully connected), a partir de las cuales se obtiene la predicción final: la clasificación del dígito manuscrito en una de las 10 clases posibles (números del 0 al 9).

[2]Proveniente del conjunto de datos MNIST[Den12]

Figura B.4: Ejemplo de detección jerárquica de patrones en una CNN [IBM24]. En la base de la pirámide se detectan características de bajo nivel (ruedas, manillar), que se combinan progresivamente para reconocer objetos de alto nivel como una bicicleta completa.

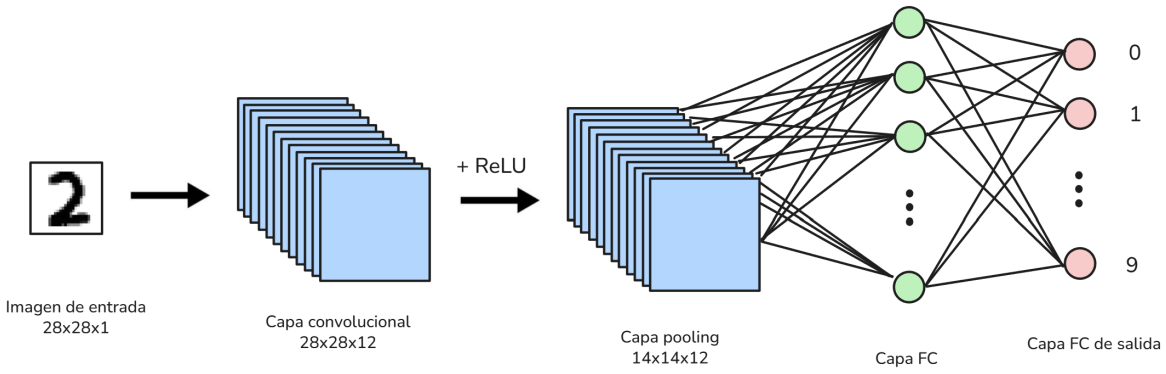


Figura B.5: Arquitectura de una red convolucional sencilla que recibe como entrada una imagen del dataset MNIST.

Anexo C

Análisis y estadísticas extra

En el apartado de datos sintéticos nuevos, se ha realizado un test de 20.000 tripletas de Serrano21 para comprobar si usar una cantidad de tripletas inferior es igual de representativa. Los resultados del test de 20.000 tripletas aleatorias a partir de 26.857 imágenes diferentes y donde solamente varía el material (dada una tripleta) se pueden ver en la Tabla C.1.

Tipo	**Total**	**Acuerdos**	**Accuracy**	**ΔDist**	**Error ΔDist**	$\Delta\phi$
sphere	2244	1560	69,52 %	0,2065	0,1992	0,2245
bunny	2228	1529	68,63 %	0,2190	0,2224	0,1975
cylinder	2215	1360	61,40 %	0,0914	0,1547	0,1971
teapot	2178	1432	65,75 %	0,1566	0,1758	0,2104
blob	2237	1498	66,96 %	0,2043	0,2098	0,2104
havran	2270	1434	63,17 %	0,1537	0,2328	0,2186
buddha	2276	1402	61,60 %	0,1207	0,2446	0,2043
dragon	2171	1374	63,29 %	0,1301	0,2332	0,1711
statuette	2181	1260	57,77 %	0,0867	0,2700	0,1843
Global	**20000**	**12849**	**64,24 %**	**0,1523**	**0,2172**	**0,2022**

Tabla C.1: 20000 tripletas de Serrano21. Dada una tripleta, solamente varía el material. Agrupado por geometría.

Se ha hecho el mismo test para el caso del *dataset* de Flickr, tal y como se puede ver en la Tabla C.2.

Material	**Tripletas**	**Aciertos**	**Accuracy**	**ΔDist (media $\pm$ σ)**	**Error ΔDist**
fabric	2023	1082	53,48 %	0,0290 $\pm$ 0,4040	0,3033
foliage	1942	1193	61,43 %	0,1389 $\pm$ 0,4541	0,2918
glass	1965	1180	60,05 %	0,1128 $\pm$ 0,4505	0,3145
leather	1992	1117	56,07 %	0,0932 $\pm$ 0,4226	0,2734
metal	2012	1124	55,86 %	0,0708 $\pm$ 0,4770	0,3493
paper	2038	1193	58,54 %	0,1282 $\pm$ 0,4921	0,3196
plastic	1980	1267	63,99 %	0,1329 $\pm$ 0,3803	0,2603
stone	2058	1277	62,05 %	0,1312 $\pm$ 0,4226	0,27
water	2018	1351	66,95 %	0,2120 $\pm$ 0,4882	0,3192
wood	1972	1121	56,85 %	0,0498 $\pm$ 0,3918	0,2956
Global	**20000**	**11905**	**59,52 %**	**0,1099 $\pm$ 0,4428**	**0,3003**

Tabla C.2: Tabla de estadísticas detalladas de los resultados obtenidos al usar el dataset de Flickr con el modelo de Lagunas19.

Anexo D

Puntualizaciones sobre la visualización con Grad-CAM

En la figura D.1 se muestra la arquitectura simplificada de la ResNet34 junto con las capas empleadas para la visualización de los mapas de calor.

Grad-CAM no realiza ningún tipo de clusterización ni PCA y para obtener su mapa de calor de salida, hace una interpolación de la imagen de entrada con las activaciones resultantes.

En cuanto a la salida que produce el modelo de Lagunas, debido a que la primera capa convolucional de cada bloque residual y la primera capa convolucional de la arquitectura tienen `stride = 2`, se está reduciendo cada vez a la mitad la resolución de la imagen. Por este motivo, si procesamos la imagen del perro y el gato, lo que obtendremos en la última capa convolucional será una imagen de 9x6 píxeles[1].

[1]Si la resolución inicial es 284x177, la resolución final se calcula de la siguiente manera: $284/2^5+1 = 9$; $177/2^5 + 1 = 6 \Rightarrow$ 9x6 de resolución final.

Figura D.1: Arquitectura simplificada de la ResNet34 usada en este trabajo y en Lagunas et al. (2019). En azul están subrayadas las capas convolucionales y bloques residuales empleados en la visualización con mapas de calor.