

Universidad Nacional de San Agustín de
Arequipa
Facultad de Ingeniería de producción y servicios
Escuela profesional de Ciencia de la computación



La tesis titulada:

**Clasificación de cerámicas arqueológicas 3D
usando *multi-view geodesic farthest point
sampling*.**

Tesis presentada por el Bachiller:
Patrick Anthony Lazo Colque

Para optar el Título Profesional de:
Licenciado en Ciencia de la Computación

Asesor:
Dr. Cristian José López del Álamo

Arequipa, Perú.
2021

Este trabajo está dedicado a mis padres.

Agradecimientos

Agradezco Dios, por darme existencia y guiar cada instante de mi vida.

Agradezco a mi familia, por todo el apoyo que me brindaron, con cuya motivación pude culminar exitosamente esta carrera profesional.

Agradezco a mi asesor Cristian José López del Álamo, por creer en mí y apoyarme incondicional en el transcurso de la elaboración de esta tesis.

Agradezco al Dr. Ivan Anselmo Sipiran Mendoza, por los conocimientos brindados y la guía permanente para alcanzar los objetivos propuestos.

Financiamiento: Este trabajo ha sido financiado por el Proyecto de Mejoramiento y Ampliación de los Servicios del Sistema Nacional de Ciencia, Tecnología e Innovación Tecnológica (Banco Mundial, Concytec), Nro. Proyecto 062-2018-FONDECYT-BM-IADT-AV.

Resumen

La clasificación de cerámicas arqueológicas es un aspecto clave en la arqueología, ya que permite identificar los diferentes tipos de cerámicas y, posteriormente, interpretar sus funciones o usos que tuvieron en la antigua sociedad. Gracias al desarrollo de tecnologías de adquisición 3D como sensores y a los esfuerzos en *computer vision* (CV) y *deep learning* es posible diseñar un método inteligente capaz de automatizar la clasificación de cerámicas arqueológicas.

En la presente tesis, se propone un descriptor similar a *Multiview-Curvature* denominado *Multiview-Geodesic* para la clasificación de cerámicas arqueológicas. *Multiview-Geodesic* está basado en la combinación de los mapas de distancias geodésicas generadas desde varias fuentes con las múltiples vistas de las cerámicas arqueológicas. El descriptor propuesto fue probado en dos conjuntos de cerámicas arqueológicas (a) *Peruvian Dataset* representada por 938 cerámicas arqueológicas recolectadas de diferentes museos en la ciudad de Lima-Peru, y (b) *3D Pottery dataset* representada por 411 cerámicas, obteniendo la mejor puntuación tanto en *Peruvian dataset* con 82,67% como en *3D Pottery dataset* con 97,56%.

|Palabras clave— Computer vision, deep learning, clasificación de cerámicas arqueológicas, multi-view, distancia geodésica

Abstract

The classification of archaeological ceramics is a key aspect in archaeology, since it allows the different types of ceramics to be identified and, later, to interpret their functions or uses that they had in ancient society. Thanks to the development of 3D acquisition technologies such as sensors and efforts in computer vision (CV) and deep learning, it is possible to design an intelligent method capable of automating the classification of archaeological ceramics.

In this thesis, a descriptor similar to Multiview-Curvature called Multiview-Geodesic is proposed for the classification of archaeological ceramics. Multiview-Geodesic is based on the combination of geodesic distance maps generated from various sources with multiple views of archaeological ceramics. The proposed descriptor was tested on two sets of archaeological ceramics (a) Peruvian Dataset represented by 938 archaeological ceramics collected from different museums in the city of Lima-Peru, and (b) 3D Pottery dataset represented by 411 ceramics, obtaining the best score both in Peruvian dataset with 82,67% and in 3D Pottery dataset with 97,56%.

keywords— Computer vision, deep learning, classification of archaeological ceramics, multi-view, geodesic distance.

Lista de símbolos	ix
1 Introducción	1
1.1 Contexto y motivación	3
1.2 Descripción del problema	4
1.3 Objetivos	5
1.3.1 Objetivo general	5
1.3.2 Objetivos específicos	5
1.4 Estructura del documento	6
2 Marco teórico	7
2.1 Descriptor de forma 3D	7
2.2 Distancia geodésica	8
2.2.1 Algoritmo <i>fast marching</i>	9

Contenido

2.3	<i>Multiview-CNN</i>	12
2.3.1	Extracción de características	13
2.3.2	<i>View pooling</i>	13
2.3.3	Clasificador	14
2.4	<i>Transfer learning</i>	14
3	Trabajos relacionados	16
3.1	<i>Voxels</i>	16
3.2	Mallas	18
3.3	Nubes de puntos	19
3.4	Proyecciones 2D (multiviews)	21
3.5	Resumen	23
4	Propuesta	24
4.1	Normalización de modelos 3D	26
4.1.1	Traslación	26
4.1.2	Escalamiento	27
4.1.3	Rotación	27
4.2	Construcción del descriptor	28
4.2.1	Cálculo de mapas de distancias geodésicas desde varias fuentes	28
4.2.2	Generación de vistas	30
4.3	Proceso de clasificación	32

5 Experimentos y resultados	35
5.1 Bases de datos	35
5.1.1 <i>Peruvian Dataset</i>	35
5.1.2 <i>3D Pottery Dataset</i>	39
5.2 Criterio de evaluación	40
5.3 Experimentos	41
5.4 Discusión	44
5.5 Código fuente	48
6 Conclusiones y trabajos futuros	49
6.1 Conclusiones	49
6.2 Trabajos Futuros	50
Bibliografía	51

Abreviaturas

Abreviatura	Término
<i>CNN</i>	<i>Convolutional Neural Network.</i>
<i>3DConvNet</i>	<i>3D Convolutional Neural Network.</i>
<i>MVCNN</i>	<i>Multiview-CNN.</i>
<i>BoW</i>	<i>Bag of Words.</i>
<i>FPS</i>	<i>Farthest Point Sampling.</i>
<i>CV</i>	<i>Computer Vision.</i>
<i>BPS</i>	<i>Basis Point Set.</i>
<i>DGCNN</i>	<i>Dynamic Graph CNN.</i>
<i>MLP</i>	<i>Multi Layer Perceptron.</i>

Lista de Figuras

2-1	Ejemplos de descriptores usados para representar las características de un objeto 3D.	8
2-2	Esquemas de actualización del algoritmo de <i>dijkstra</i> y <i>fast marching</i> [Bronstein et al., 2008].	10
2-3	Modelo 3D <i>bunny</i> proyectado usando el mapa de distancias geodésicas desde un punto origen.	12
2-4	Arquitectura multiview-CNN [Su et al., 2015].	13
3-1	Cerámica representa usando <i>voxels</i> : podemos notar como se pierde la suavidad de la superficie del modelo 3D.	17
3-2	Cerámicas representadas usando <i>mallas</i> : a la izquierda se aprecia una malla totalmente conectada, a la derecha una malla desconectada.	18
3-3	Codificación de una nube de puntos usando 128 puntos fijos.	20

Lista de Figuras

3-4	Construcción del descriptor de objetos 3D Multi-View, utilizando el mapa de curvatura media [Canul-Ku et al., 2018].	22
4-1	Diagrama de Flujo del proceso de clasificación de cerámicas arqueológicas. (a) Normalización de la cerámica arqueológica aplicando traslación, escalamiento y rotación, (b) Construcción del descriptor combinando las distancias geodésicas con la arquitectura <i>multiview</i> -CNN, (c) Proceso de clasificación empleando una MLP (<i>Multi Layer Perceptron</i>).	25
4-2	A la izquierda se aprecia la cerámica 3D proyectada con el punto inicial (vértice máximo en el eje Y), a la derecha la misma cerámica con 128 puntos obtenidos del cálculo de FPS a partir del punto inicial dado.	30
4-3	Generación de vistas de una cerámica 3D proyectada usando mapas de distancias geodésicas desde 128 fuentes.	31
4-4	Arquitectura CNN VGG-11 utilizada para la extracción de características.	33
4-5	Proceso de clasificación a través de la agrupación de cada vector característico.	34
5-1	Interfaz web inicial para organizar base de datos de cerámicas peruanas por categoría.	36
5-2	Interfaz web para categorizar una cerámica en específico.	37

Lista de Figuras

5-3	Muestra aleatoria (una por clase) del conjunto de datos de cerámicas 3D peruanas.	38
5-4	Diagrama de barras de la distribución de clases del conjunto de datos Peruano. En la parte superior de cada barra se aprecia el número de cerámicas correspondientes a cada categoría.	38
5-5	Muestra aleatoria de formas elegidas (una por clase) del conjunto de datos <i>3D Pottery</i>	39
5-6	Diagrama de barras de la distribución de clases del conjunto de datos <i>3D Pottery</i> . En la parte superior de cada barra se aprecia el número de cerámicas correspondientes a cada categoría.	40
5-7	Comparación de las matrices de confusión correspondientes al método MVCNN y el método propuesto (MVCNN- <i>Geodesic</i>) usando el conjunto de pruebas de <i>Pottery Dataset</i> . Para generar la matriz de confusión de MVCNN- <i>Geodesic</i> se utilizó el experimento max-1 de la tabla 5-1	44
5-8	Comparación de las matrices de confusión correspondientes al método MVCNN y el método propuesto (MVCNN- <i>Geodesic</i>) usando el conjunto de pruebas de <i>Peruvian Dataset</i> . Para generar la matriz de confusión de MVCNN- <i>Geodesic</i> se utilizó el experimento max-128 de la tabla 5-1	46

Lista de Figuras

5-9 Análisis de las categorías que se confunden con mayor frecuencia pertenecientes a la base de datos *Peruvian*, en donde cada categoría está representada por una cerámica proyectada usando el descriptor propuesto. 47

Lista de Tablas

3-1	Resumen de los trabajos relacionados seleccionando algunos de los más relevantes y describiendo sus ventajas y desventajas.	23
5-1	Comparación de la precisión de clasificación obtenida del método propuesto (MVCNN- <i>Geodesic</i>) con diferente número de puntos FPS utilizando como punto inicial el vértice máximo y mínimo del eje Y.	43
5-2	Comparación de la precisión de clasificación obtenida del método propuesto (MVCNN- <i>Geodesic</i>) con otros métodos de clasificación sobre los dos conjuntos de datos de cerámicas arqueológicas (<i>Peruvian</i> y <i>3D Pottery</i>). Para el método propuesto MVCNN- <i>Geodesic</i> se presentó el porcentaje de precisión más alto obtenido de la Tabla 5-1	43

En el Perú existieron muchas culturas preincaicas como la cultura Nazca o Lima las cuales, a pesar de haberse desarrollado hace siglos, han dejado evidencias de sus costumbres y tradiciones siendo una de las categorías más representativas las cerámicas. El estudio de las cerámicas ayuda a que los arqueólogos puedan reconstruir patrones culturales de la antigua sociedad a la cual pertenecieron [Cintas et al., 2020].

Una tarea esencial dentro del estudio de las cerámicas es la clasificación, ya que permite que los arqueólogos puedan identificar los diferentes tipos de cerámica y, posteriormente, interpretar sus funciones o usos que tuvieron en la antigua sociedad [Hermes, 1991].

Por otro lado, con el desarrollo de tecnologías de adquisición de información tridimensional, como sensores 3D, y a los esfuerzos en *computer vision* y *deep learning*, es posible abordar el problema de clasificación de cerámicas 3D desde una

1 Introducción

perspectiva computacional [Ahmed et al., 2018].

Dentro de los diferentes modelos propuestos para el análisis y clasificación de objetos 3D, las *Convolutional Neural Network* (CNN), además de haber demostrado tener un buen desempeño para la clasificación de imágenes, han logrado ser adaptadas para la clasificación de modelos 3D [Ahmed et al., 2018].

Un enfoque simple fue transformar los modelos 3D a una cuadrícula regular en el espacio tridimensional (*voxel*), de esta forma los *voxels* pueden ser clasificados usando una 3D CNN [Wu et al., 2015], sin embargo los *voxels* son computacionalmente costosos y no conservan la suavidad de la superficie del modelo 3D. A diferencia de *voxels*, la nube de puntos es una estructura no euclidiana [Ahmed et al., 2018], por lo que, no se tiene claro cómo se deben realizar las convoluciones, Qi et al. [Qi et al., 2017a] fueron pioneros en este concepto proponiendo una red simple pero eficiente sobre nube de puntos denominada Pointnet, trabajos posteriores propusieron enfoques similares basados en la agrupación para lograr el ordenamiento de la nube puntos [Li et al., 2018], [Wang et al., 2019], [Prokudin et al., 2019].

Otro enfoque para utilizar CNN sobre modelos 3D es usando Multiview-CNN [Su et al., 2015], este enfoque representa el modelo 3D a través de proyecciones 2D desde varios puntos de vista, para luego extraer información de cada vista usando una clásica CNN para imágenes como Alexnet, Restnet o VGG. Este

enfoque permite el uso de *transfer learning* [Hussain et al., 2018] ya que se puede obtener grandes bases de datos de imágenes 2D como Imagenet [Deng et al., 2009].

Recientemente, Canul-Ku et al. propusieron un descriptor denominado *Multiview-Curvature* [Canul-Ku et al., 2018] para la clasificación de objetos arqueológicos. Este descriptor esta basado en la combinación de los mapas de curvaturas con las múltiples vistas de los objetos arqueológicos. Similarmente, en esta tesis se propone un descriptor basado en la combinación de mapas de distancias geodésicas generadas desde varias fuentes con las múltiples vistas de las cerámicas arqueológicas 3D. El descriptor propuesto fue probado en dos conjuntos de cerámicas arqueológicas *Peruvian dataset* y *3D Pottery dataset*, obteniendo la mejor puntuación tanto en *Peruvian dataset* con 82,67% como en *3D Pottery dataset* con 97,56%.

1.1. Contexto y motivación

A pesar de que existen métodos para la clasificación de modelos 3D, estos no son lo suficientemente buenos para la clasificación de cerámicas arqueológicas [Scopigno et al., 2011]. Por lo tanto, resulta interesante poder diseñar un descriptor capaz de mejorar la clasificación de cerámicas arqueológicas.

Además, crear un sistema capaz de clasificar cerámicas arqueológicas, de forma automática, podría facilitar el desarrollo de diferentes áreas, como por ejemplo en arqueología para ayudar a entender las costumbres de sociedades antiguas [Cin-

tas et al., 2020] y en la reconstrucción de cerámicas 3D, donde el proceso de clasificación puede ser usado como fase inicial de la reconstrucción [Hermeza and Sipiran, 2018], por lo que resulta importante tener un modelo robusto de clasificación.

1.2. Descripción del problema

Desde el punto de vista social, la clasificación de cerámicas 3D es una tarea difícil y compleja debido a que, generalmente, involucra cientos de cerámicas. Sin embargo, la clasificación resulta de mucha utilidad para que los arqueólogos puedan identificar la categoría a la que pertenecen y posteriormente entender la función que desempeñaba cada cerámica [Cintas et al., 2020].

Con el desarrollo de tecnologías de adquisición 3D, como sensores, y a los esfuerzos en *computer vision* y *deep learning*, es posible abordar el problema de clasificación de cerámicas 3D desde una perspectiva computacional [Ahmed et al., 2018].

Desde el punto de vista computacional, el punto crítico del proceso de clasificación es encontrar un descriptor de forma 3D (detallado en la sección 2.1), que permita generar un vector característico robusto capaz de mejorar la precisión de clasificación [Ahmed et al., 2018].

Por lo expuesto anteriormente, proponer un descriptor eficiente para la clasi-

ficación de cerámicas 3D es una tarea difícil y compleja.

1.3. Objetivos

1.3.1. Objetivo general

El objetivo de la presente tesis es proponer un descriptor basado en la combinación de distancias geodésicas con múltiples vistas, al cual se denominó *multi-view geodesic farthest point sampling*, para la clasificación de cerámicas arqueológicas 3D.

1.3.2. Objetivos específicos

- Recolectar dos bases de datos de cerámicas arqueológicas 3D.
- Normalizar las cerámicas 3D recolectadas a nivel de escala, posición y orientación.
- Analizar y comparar métodos de clasificación de modelos 3D en las dos bases de datos recolectadas.
- Implementar el algoritmo para el cálculo de distancias geodésicas de modelos 3D.

1.4. Estructura del documento

En el capítulo 2 abordamos el marco teórico necesario para entender el desarrollo de esta tesis.

En el capítulo 3 se observa y analiza los trabajos relacionados más importantes desarrollados en los últimos años, se muestra una tabla resumen describiendo las ventajas y desventajas de algunos de los trabajos más relevantes acorde a su tipo de representación.

En el capítulo 4 se expone la propuesta de este trabajo de investigación.

En el capítulo 5 se presentan los experimentos y resultados de validación para este trabajo de investigación.

En el capítulo 6 se presentan las conclusiones de este trabajo de investigación y trabajos futuros.

2.1. Descriptor de forma 3D

Un descriptor de forma 3D es una representación simplificada del objeto 3D que describe las características geométricas del mismo como la superficie, textura, profundidad o cualquier otra característica [Kazmi et al., 2013]. Las características resultantes pueden ser codificadas en forma de un vector característico $x = [x_1, x_2 \dots x_n]$ el cual, dependiendo de la información recolectada, puede ser categorizado como un descriptor global o local [Canul-Ku et al., 2018].

Un descriptor global describe el objeto 3D como un todo, mientras que un descriptor local describe características alrededor de un *key-point* (vértice del objeto 3D) [Canul-Ku et al., 2018]. Por ejemplo, en la Figura 2-1 se aprecia tres formas en la cual un objeto 3D puede ser descrito, las tres representaciones pueden ser usadas para el cálculo de características globales y locales, sin embargo, tanto la opción 2-1b como 2-1c son capaces de representar mejor las características locales

ya que permiten analizar un comportamiento común alrededor de *key-point*.

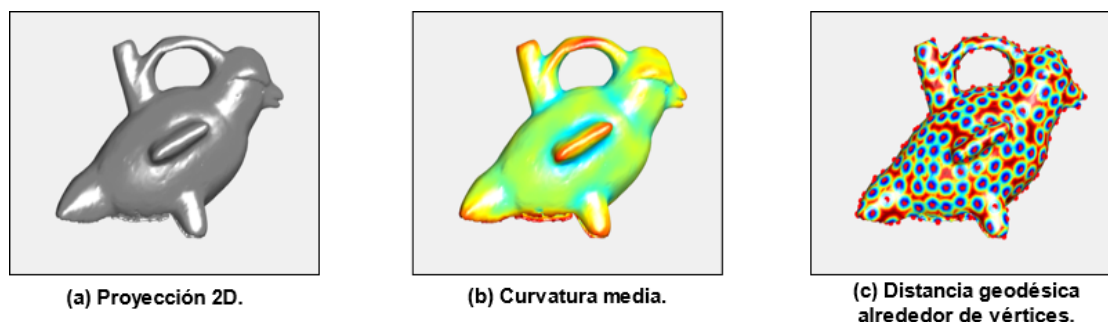


Figura 2-1: Ejemplos de descriptores usados para representar las características de un objeto 3D.

Varios descriptores manualmente diseñados (*hand-crafted descriptors*) han sido propuestos en la literatura [Tangelder and Veltkamp, 2008], estos descriptores pueden ser utilizados para realizar operaciones de clasificación mediante el enfoque BoW (*Bag of Words*) [Tangelder and Veltkamp, 2008]. Sin embargo, este enfoque sufre de limitaciones como el no poder aprender características discriminatorias. Es por ello, que *hand-crafted descriptors* han sido combinados con arquitecturas *deep learning* [Zhang et al., 2017], [Canul-Ku et al., 2018], [Barroso-Laguna et al., 2019], como es el caso de esta investigación.

2.2. Distancia geodésica

El cálculo de la distancia geodésica en mallas es importante para muchos problemas en el área de procesamiento de geometría y análisis de formas 3D, como recuperación, reconocimiento de objetos, mapeo de texturas entre otros.

La distancia geodésica puede ser definida como la ruta más corta entre dos vértices pertenecientes a una malla triangular, esta ruta más corta puede ser calculada usando una adaptación del clásico algoritmo de *dijkstra* para mallas triangulares [Mitchell et al., 1987], sin embargo este método tiene un alto costo computacional ($O(n^2 \log n)$), por ese motivo otras investigaciones abordaron el problema a través de la analogía de los fenómenos físicos como ondas de propagación y difusión de calor. El método *fast marching* pertenece a la categoría ondas de propagación [Kimmel and Sethian, 1998].

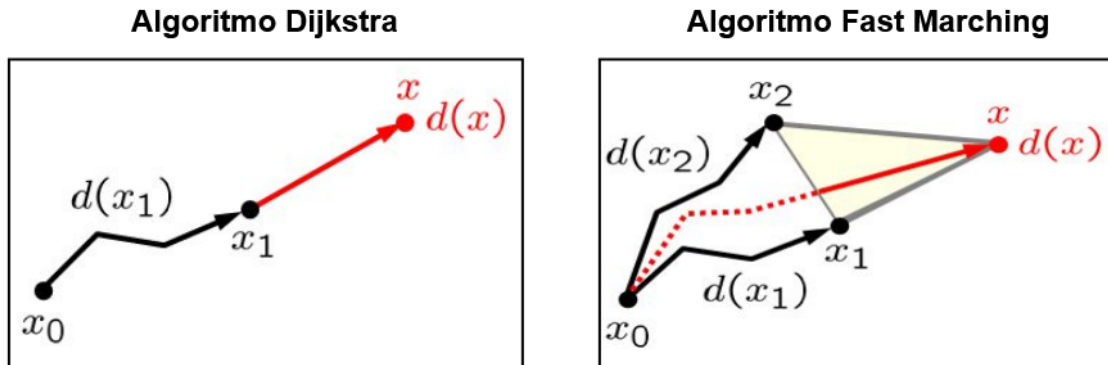
2.2.1. Algoritmo fast marching

El algoritmo de *fast marching* tiene un costo computacional de $O(n \log n)$ y su idea principal es simular la propagación de ondas sobre un conjunto discreto, excepto por el esquema de actualización que resuelve la ecuación eikonal [Kimmel and Sethian, 1998], el algoritmo de *fast marching* es similar al algoritmo de *dijkstra*. En la Figura 2-2 se muestra una comparativa de los esquemas de actualización del algoritmo de *dijkstra* y *fast marching*.

El algoritmo *fast marching* (Algoritmo 1) puede ser descrito con los siguientes pasos:

Dado un conjunto de vértices origen $s \in S$, estos son marcados como *black* (vértices visitados), a cada vértice origen se le asigna una distancia igual a 0 ($d(s) = 0$), todas las demás distancias pertenecientes a los otros vértices son ini-

Esquemas de Actualización



- El vértice x es actualizado desde el vertice adyacente x_1 .
 - La distancia $d(x)$ es computada de $d(x_1)$.
 - El camino esta restringido a los bordes del grafo.
- El vértice x es actualizado desde el triángulo (x_1, x_2, x) .
 - La distancia $d(x)$ es computada de $d(x_1)$ y $d(x_2)$.
 - El camino puede pasar sobre las caras de la malla.

Figura 2-2: Esquemas de actualización del algoritmo de *dijkstra* y *fast marching* [Bronstein et al., 2008].

cializadas con infinito $d(v) = \infty$ y marcados como *green* (vértices no visitados), d es una *priority queue* cuya prioridad es la distancia más pequeña, luego inicializamos una *queue* Q como vacía (vértices alcanzados).

Todos los vértices vecinos de los vértices *black* son insertados en Q siempre que estos sean *green*, para cada triángulo que comparte el vértice $v_0 \in Q$ utilizamos el esquema de actualización el cual actualizará la distancias (d) de los nuevos vértices.

Algoritmo 1 Fast Marching

Entrada: Malla triangular (V, F) , vértices origen $S \subset V$ **Salida:** Mapa de distancias geodésicas d

- 1: $\forall s \in S : d(s) \leftarrow 0$ y marcarlo como *black*
 - 2: $\forall v \in V : d(v) \leftarrow \infty$ y marcarlo como *green*
 - 3: Inicializar una *queue* de vertices $Q = \emptyset$.
 - 4: **while** vértices *green* $\neq \emptyset$ **do**
 - 5: Insertar en Q los vecinos de los vértices *black* si estos son *green*
 - 6: **for all** $v_0 \in Q$ **do**
 - 7: **for all** $(v_0, v_1, v_2) \in F(v_0)$ **do**
 - 8: actualización(d, v_0, v_1, v_2)
 - 9: **end for**
 - 10: **end for**
 - 11: Marcar como *black* $v_i = \arg \min d(x)$ y removerlo de Q
 - 12: **end while**
 - 13: **return** d
-

Finalmente, un nuevo vértice con el valor mínimo de d es etiquetado como *black* y removido de la lista Q , el algoritmo termina hasta que no existan vértices *green*.

El algoritmo *fast marching* retorna el mapa de distancias $d(v)$ el cual es definido como la función de distancia a un conjunto de vértices origen. Una vez obtenido el mapa de distancias este puede ser proyectado sobre el modelo 3D, para lo cual es necesario ordenar las distancias ascendentemente y interpolarlas usando una función lineal definida en un rango de $[0,1]$. Una vez obtenido las distancias interpoladas, estas son proyectadas sobre el modelo usando una escala de colores donde 0 es azul y 1 es rojo. Un ejemplo de la proyección de un modelo 3D con su mapa de distancias geodésicas se muestra en la Figura **2-3**.

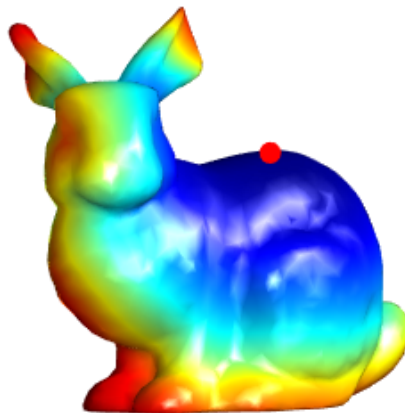


Figura 2-3: Modelo 3D *bunny* proyectado usando el mapa de distancias geodésicas desde un punto origen.

2.3. Multiview-CNN

Multiview-CNN (MVCNN) fue propuesta por Su et al. [Su et al., 2015] con el objetivo de describir de forma robusta un modelo 3D que ha diferencia de los métodos que trabajan directamente sobre representaciones 3D como mallas, *voxels* o nubes de puntos, MVCNN representa el modelo 3D en n vistas tomadas de diferentes ángulos, para luego extraer información de cada vista usando una CNN para imágenes. MVCNN ha atraído la atención de los investigadores ya que, este enfoque, permite reutilizar las arquitecturas CNN ampliamente estudiadas como Alexnet, Resnet o VGG. En la Figura 2-4 se muestra la arquitectura *multiview-*

CNN.

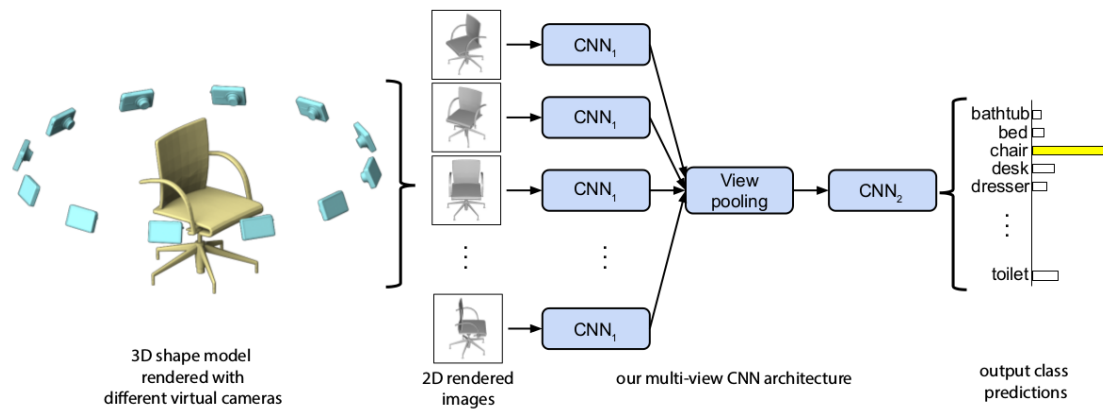


Figura 2-4: Arquitectura multiview-CNN [Su et al., 2015].

2.3.1. Extracción de características

MVCNN recibe como entrada 12 imágenes las cuales representan al modelo 3D, cada imagen es pasada a través de la primera fase de la arquitectura (CNN_1), cuyo objetivo es extraer las características de la imagen ingresada. Como resultado de CNN_1 se obtiene 12 vectores característicos correspondiente a las 12 imágenes.

2.3.2. View pooling

Una vez obtenidos los 12 vectores característicos es necesario agrupar cada vector en uno solo (view pooling), donde el vector con el valor máximo es seleccionado (*max-pooling*), también es posible utilizar otra operación como *mean-pooling* pero está usualmente tiene un menor desempeño. Para poder realizar el *view pooling* de forma correcta es necesario ingresar las imágenes en orden con respecto

a cada modelo 3D y no desordenar los vectores característicos en el proceso de entrenamiento.

2.3.3. Clasificador

Con el vector característico máximo seleccionado de las 12 imágenes este finalmente es enviado a través del clasificador (CNN_2) cuya función es reducir la dimensionalidad del mismo para posteriormente clasificarlo.

2.4. Transfer learning

Los métodos de *deep learning* son populares debido al buen desempeño que han tenido en los últimos años. Sin embargo, para que estos obtengan buenos resultados es necesario contar con un amplio conjunto de datos, ya que estos métodos no pueden asegurar el correcto funcionamiento cuando se ingresa nuevos datos (*overfitting problem*) [Ying, 2019].

Una forma de utilizar *deep learning* en pequeños conjuntos de datos es usando *transfer learning*. *Transfer learning* es una técnica de aprendizaje automático en la cual se entrena un modelo para una tarea y luego se reutiliza el mismo modelo en una tarea relacionada, *transfer learning* es aplicado generalmente cuando hay un nuevo conjunto de datos más pequeño que el conjunto de datos original [Hussain et al., 2018]. Debido a que la arquitectura MVCNN puede ser utilizada con alguna

2 Marco teórico

de las clásicas CNNs para imágenes, es posible utilizar los modelos pre-entrenados de ImageNet [Deng et al., 2009], de esta forma se evita iniciar el proceso de aprendizaje con pesos aleatorios.

La clasificación de modelos 3D es un tema desafiante que ha atraído la atención de la comunidad de *computer vision* en los últimos años. En la literatura existen diferentes métodos los cuales pueden ser divididos acorde a su tipo de representación: *voxels*, mallas, nube de puntos y proyecciones 2D. Donde las proyecciones 2D han sido comúnmente empleadas para la clasificación y recuperación de objetos arqueológicos [Hou et al., 2020], [Canul-Ku et al., 2018], [Benhabiles and Tabia, 2016].

3.1. *Voxels*

Los datos 3D se pueden transformar a una cuadrícula regular en el espacio tridimensional (*voxel*). Esta representación es análoga a la cuadrícula 2D de una imagen, ya que métodos que se aplican en cuadrículas 2D se pueden extender a cuadrículas 3D como la clásica *Convolutional Neural Network* (CNN).

Li et al. [Wu et al., 2015] fueron pioneros en este concepto, presentaron una CNN que procesa formas voxelizadas para su clasificación y completación. Posteriormente, Maturanad y Scherers [Maturana and Scherer, 2015] presentó VoxNet, una arquitectura CNN 3D para una clasificación eficiente en *voxels*. A pesar de la simplicidad de la representación y su capacidad para codificar información sobre la forma 3D, los *voxels* son computacionalmente costosos y requieren un uso significativo de memoria. Para mitigar esto, se presentó una representación volumétrica denominada *octree* [Tatarchenko et al., 2017]. La representación *octree* esta basada en la descomposición recursiva de la raíz de los *voxels* similar a la estructura *quadtree* [Samet, 1984]. Sin embargo, tanto *voxels* como *octree* no conservan la geometría de los objetos 3D como la suavidad de la superficie. En la Figura 3-1 se muestra la diferencia entre una representación normal (mallas) y una usando *voxels*.

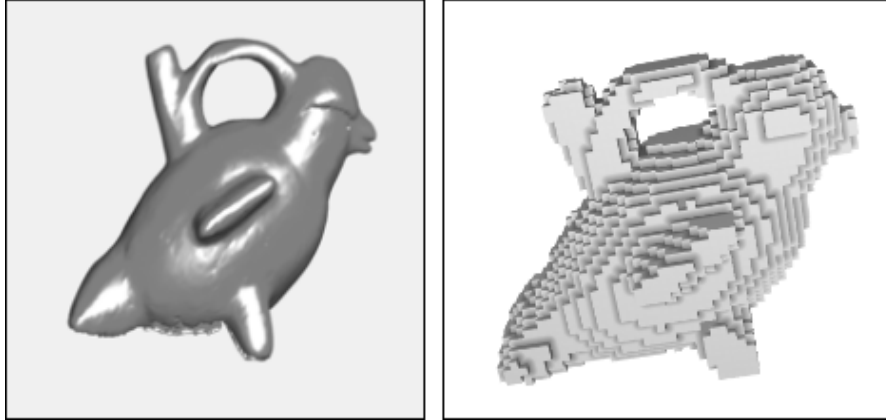


Figura 3-1: Cerámica representa usando *voxels*: podemos notar como se pierde la suavidad de la superficie del modelo 3D.

3.2. Mallas

Las mallas 3D son las representaciones más populares de formas 3D, sin embargo, en un aspecto global, las mallas son datos no euclidianos haciendo inviable el uso de arquitecturas CNN clásicas. Para superar este problema, MeshCNN [Hancock et al., 2019] y MeshNet [Feng et al., 2019] utilizaron las propiedades de las mallas para un análisis directo a través de un modelo de aprendizaje profundo, creando operaciones como *meshconv* y *meshpool*.

Estos trabajos tuvieron buenos resultados en conjuntos de datos generales, pero están restringidos a modelos 3D *manifold* [Huang et al., 2018], lo que implica que tienen que cumplir ciertas propiedades, una propiedad en particular requiere que la malla este completamente conectada. Debido a que estos métodos trabajan sobre modelos *manifold*, restringen su aplicabilidad en conjuntos de datos arqueológicos donde se presentan mallas 3D desconectadas, un ejemplo de una malla desconectada se aprecia en la Figura 3-2.

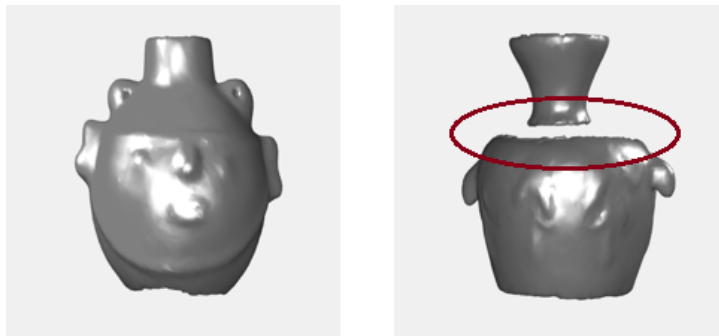


Figura 3-2: Cerámicas representadas usando *mallas*: a la izquierda se aprecia una malla totalmente conectada, a la derecha una malla desconectada.

3.3. Nubes de puntos

Las nubes de puntos son la forma más sencilla de representar objetos 3D sin pre-procesarlos como *voxels* o proyecciones 2D y sin tener las limitaciones de las mallas. De manera similar a las mallas, las nubes de puntos son datos no euclidianos, lo que significa que no es claro cómo se deben realizar las convoluciones. Qi et al. fueron pioneros en este concepto proponiendo una red simple pero eficiente sobre nubes de puntos denominada Pointnet [Qi et al., 2017a]. Sin embargo, Pointnet solo considera las características globales e ignora las características locales. En su siguiente trabajo introdujeron una red neuronal jerárquica que aplica Pointnet de forma recursiva denominada PointNet++ [Qi et al., 2017b], este enfoque permite capturar mejor las características locales.

Las CNNs son muy eficientes para procesar datos representados en cuadrículas, sin embargo las nubes de puntos generalmente no son cuadrículas, lo que dificulta el aprendizaje de la información local. Un método simple para superar este problema es construir redes neuronales en grafos [Bruna et al., 2013]. En ese sentido, DGCNN [Wang et al., 2019] explota las estructuras geométricas locales construyendo una vecindad local de grafos y aplicando convoluciones sobre grafos. Recientemente, Li et al. [Li et al., 2018] presentaron PointCNN, que extiende la noción de convolución de una cuadrícula local a una convolución \mathcal{X} sobre puntos que residen en una vecindad euclidiana.

3 Trabajos relacionados

Otro enfoque para resolver la estructura no euclidiana de la nube de puntos es codificarla usando un descriptor, dicho descriptor es utilizado como entrada de un modelo aprendizaje como por ejemplo una MLP (*Multi Layer Perceptron*) o una 3D ConvNet. En ese sentido Ben-Shabat et al. propusieron 3DmFV-Net [Ben-Shabat et al., 2017] una arquitectura 3D ConvNet que usa como descriptor una modificación de *fisher vectors* en forma de cuadrícula 3D, dicho descriptor es utilizado por una 3D ConvNet para la posterior clasificación. Similarmente, BPS (*Basis Point Sets*) [Prokudin et al., 2019] codifica un objeto 3D como distancias mínimas a un conjunto de puntos fijos (BPS). BPS es un forma eficiente de describir una nube de puntos, especialmente BPS en forma esférica uniforme al azar genera un descriptor 1D dimensional el cual, puede ser usado como entrada de una MLP haciendo de este un método rápido y eficiente, un ejemplo de BPS en forma esférica uniforme al azar se muestra en la Figura 3-3.

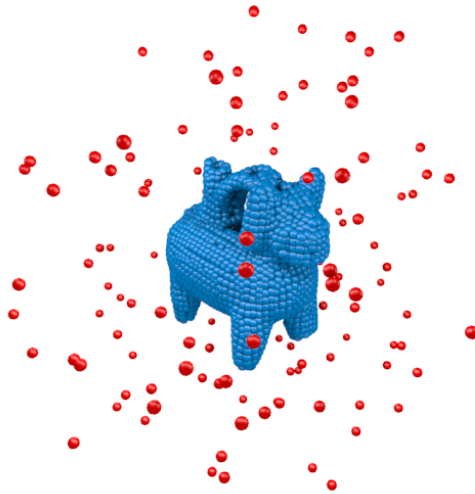


Figura 3-3: Codificación de una nube de puntos usando 128 puntos fijos.

3.4. Proyecciones 2D (multiviews)

Un objeto 3D se puede representar a través de sus proyecciones 2D desde varios puntos de vista para posteriormente, adaptar las técnicas y arquitecturas existentes aplicadas en el dominio de la imagen 2D como las arquitecturas CNN 2D. Su et al. [Su et al., 2015] fueron los primeros en aplicar una CNN multivista para la tarea de clasificación de formas, presentando un descriptor compacto de vistas 2D con una nueva arquitectura denominada *multiview-CNN*.

Con el fin de explorar enfoques volumétricos y basados en vistas, Qi et al. [Qi et al., 2016] compararon estos dos enfoques y observaron una superioridad con el método *multiview* en comparación con los métodos disponibles en ese momento.

En ese sentido, diferentes investigaciones han utilizado *multiviews* como descriptor de objetos arqueológicos. Benhabilesh y Tabiah [Benhabiles and Tabiah, 2016] usaron una CNN para la recuperación de objetos arqueológicos, diseñaron un descriptor de vistas 2D, donde cada descriptor está asociado con una BoW (*Bag of Words*).

Recientemente, Canul-Ku et al. [Canul-Ku et al., 2018] propusieron un descriptor *multiview* 3D enfocado en las curvaturas de los objetos arqueológicos (Figura 3-4). Este método obtuvo la puntuación más alta en comparación con otros descriptores y fue probado usando dos bases de datos de objetos arqueológicos

3 Trabajos relacionados

3D pottery content-based retrieval benchmark dataset [Koutsoudis et al., 2010] y *non-public Mexican dataset*.

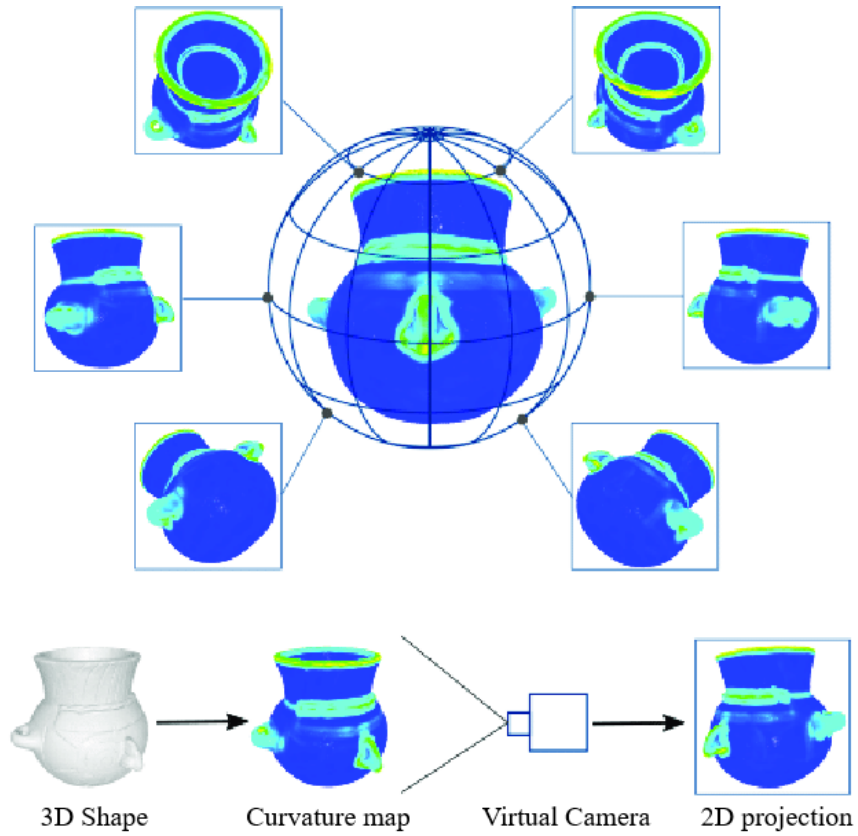


Figura 3-4: Construcción del descriptor de objetos 3D Multi-View, utilizando el mapa de curvatura media [Canul-Ku et al., 2018].

3.5. Resumen

Los trabajos relacionados han sido resumidos usando la Tabla **3-1**, en donde se han seleccionado alguno de los trabajos más relevantes por cada tipo de representación y se describen sus ventajas y desventajas.

Representación	Métodos	Ventajas	Desventajas
<i>Voxels</i>	<ul style="list-style-type: none"> ▪ Voxnet 	<ul style="list-style-type: none"> ▪ Dato euclideo 	<ul style="list-style-type: none"> ▪ Pre-procesamiento ▪ Alto costo computacional
Mallas	<ul style="list-style-type: none"> ▪ MeshCNN ▪ MeshNet 	<ul style="list-style-type: none"> ▪ No es necesario pre-procesar 	<ul style="list-style-type: none"> ▪ Dato no euclideo ▪ Restringido a modelos <i>manifold</i>
Nubes de puntos	<ul style="list-style-type: none"> ▪ Pointnet ▪ DGCNN ▪ BPS 	<ul style="list-style-type: none"> ▪ Representación más sencilla ▪ No es necesario pre-procesar 	<ul style="list-style-type: none"> ▪ Dato no euclideo
Proyecciones 2D	<ul style="list-style-type: none"> ▪ MVCNN ▪ MVCNN-<i>Curvature</i> 	<ul style="list-style-type: none"> ▪ Dato euclideo ▪ Es posible reutilizar arquitecturas CNN 2D 	<ul style="list-style-type: none"> ▪ Pre-procesamiento

Tabla 3-1: Resumen de los trabajos relacionados seleccionando algunos de los más relevantes y describiendo sus ventajas y desventajas.

En este capítulo se describe, de manera detallada los pasos que se siguen en el proceso de clasificación de cerámicas 3D propuesto. En primer lugar, se normaliza la cerámica arqueológica (Sección 4.1), luego, con la cerámica normalizada calculamos mapas de distancias geodésicas desde varias fuentes (Sección 4.2). Una vez concluido este proceso, cada cerámica arqueológica es proyectada usando los mapas de distancias geodésicas previamente calculados, esto con el fin de obtener 12 imágenes tomadas de diferentes ángulos. Finalmente, las 12 imágenes resultantes son pasadas por la arquitectura MVCNN para la posterior clasificación (Sección 4.3). En la Figura 4-1 se muestra el diagrama de flujo del método propuesto el cual está dividido en tres fase: la primera fase (a) normaliza la cerámica utilizando las transformaciones de translación, escalamiento y rotación, la segunda fase (b) abarca el proceso de construcción del descriptor en el cual se ingresa la cerámica normalizada y se retorna un vector característico, finalmente, la tercera fase (c) clasifica el vector característico.

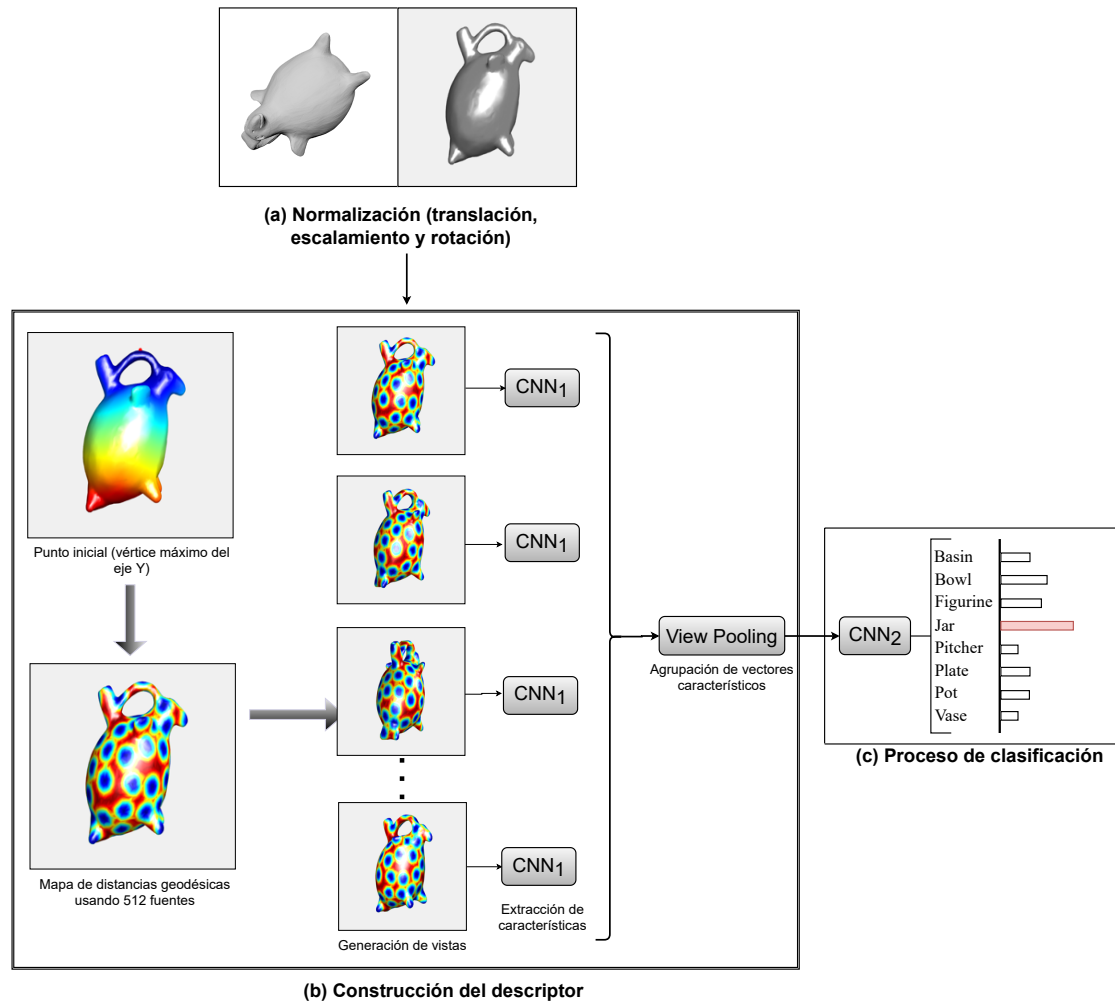


Figura 4-1: Diagrama de Flujo del proceso de clasificación de cerámicas arqueológicas. (a) Normalización de la cerámica arqueológica aplicando traslación, escalamiento y rotación, (b) Construcción del descriptor combinando las distancias geodésicas con la arquitectura *multiview-CNN*, (c) Proceso de clasificación empleando una MLP (*Multi Layer Perceptron*).

4.1. Normalización de modelos 3D

En el proceso de digitalización 3D es común que la posición escala y orientación de los cerámicas 3D sean arbitrarias. Sin embargo, aplicaciones en clasificación, recuperación y reconocimiento requieren que los modelos sean previamente normalizados.

La normalización de modelos 3D consiste en 3 transformaciones básicas: translación, escalamiento y rotación, las cuales son aplicadas a toda la base de datos. Los siguientes pasos describen como normalizamos un objeto 3D la cual es denotado como \mathcal{P} .

4.1.1. Traslación

La transformación de translación consiste en mover un objeto 3D al origen del sistema de coordenadas, los siguientes pasos describen el proceso de translación:

- El primer paso consiste en calcular el centroide del objeto 3D denotado como c :

$$(4-1) \quad c = \frac{1}{n} \sum_{i=1}^n p_i$$

donde $p_i \in \mathcal{P}$ representa un vértice del objeto 3D (x_i, y_i, z_i) , y $\mathcal{P} = \{p_1, p_2, \dots, p_n\}$ es el conjunto de vértices.

- Finalmente, el centroide c es restado a los vértices del objeto 3D.

$$(4-2) \quad p'_i = p_i - c$$

4.1.2. Escalamiento

La transformación de escalamiento consiste en escalar un objeto 3D en un rango entre $[-1, 1]$ del sistema de coordenadas, los siguientes pasos describen el proceso de escalamiento:

- El primer paso consiste en calcular m representado por la distancia más grande entre el centroide y un vértice p_i .

$$(4-3) \quad m = \max_{P_i \in \mathcal{P}} \|p_i - c\|$$

- Finalmente, todos los vértices son divididos por la distancia más grande (m).

$$(4-4) \quad p'' = \frac{p'_i}{m}$$

4.1.3. Rotación

La transformación de rotación consiste en alinear un objeto 3D con respecto a un eje dado (X, Y, Z). Debido a que las cerámicas 3D recolectadas no estaban alineadas con respecto a un solo eje, fue necesario rotar varias de ellas manualmente.

En esta investigación se alineó todas las cerámicas en el eje Y .

4.2. Construcción del descriptor

El descriptor propuesto, es el resultado de combinar los mapas de distancias geodésicas con múltiples vistas (12 imágenes) del objeto 3D. La intuición geométrica detrás del método propuesto es que los descriptores geodésicos de objetos similares también son similares, incluso en vistas 2D. Cada imagen captura una vista parcial de la cerámica 3D la cual es renderizada usando mapas de distancias geodésicas desde varias fuentes, el descriptor final es obtenido agrupando las 12 imágenes utilizando el enfoque de MVCNN.

El método propuesto para la construcción del descriptor, se resume en los siguientes pasos:

4.2.1. Cálculo de mapas de distancias geodésicas desde varias fuentes

Para el cálculo de mapas de distancias desde varias fuentes se utilizó el algoritmo genérico *farthest point sampling* (FPS) introducido por Eldar [Eldar et al., 1997], el algoritmo resuelve el problema de dado un conjunto de n puntos encontrar m de ellos de modo que estén lo más alejados entre sí. FPS recibe como entrada un punto inicial en esta investigación se estableció como punto inicial el vértice cuyo valor en el eje Y sea el máximo, esto debido a que las cerámicas están alineadas

Algoritmo 2 *Farthest Point Sampling* (FPS)

Entrada: Malla triangular (V, F) , vértice origen $s \in V$, número de vértices seleccionados m

Salida: Conjunto de vértices seleccionados $S \subset V$

```

1:  $S \leftarrow \{s\}$ 
2:  $D \leftarrow \text{perform\_fast\_marching\_mesh}(V, F, s)$ 
3: for  $i \leftarrow 2$  to  $m$  do
4:    $s \leftarrow \arg \max_{v \in V} D(v)$ 
5:    $S \leftarrow S \cup \{s\}$ 
6:    $D_1 \leftarrow \text{perform\_fast\_marching\_mesh}(V, F, s)$ 
7:    $D \leftarrow \min(D, D_1)$ 
8: end for
9: return  $S$ 

```

con respecto al eje Y. Un ejemplo de la selección del punto inicial se aprecia en la Figura 4-2.

Para calcular FPS en mallas triangulares es necesario computar la distancia geodésica utilizando el algoritmo *fast marching* en cada iteración. El algoritmo FPS (Algoritmo 2), puede ser descrito de la siguiente manera:

Dada una malla triangular conformada por vértices \mathbf{V} , y *faces* \mathbf{F} , escogemos un vértice origen \mathbf{s} , $\mathbf{s} \in \mathbf{V}$ y es agregado al conjunto solución \mathbf{S} , posteriormente las distancias geodésicas \mathbf{D} correspondientes a \mathbf{s} son calculadas. Inicializamos un bucle *for* desde dos hasta m , dentro del bucle se selecciona el vértice con la mayor distancia en \mathbf{D} el cual es asignado como el nuevo vértice origen y agregado al conjunto solución \mathbf{S} , seguidamente se calcula las nuevas distancias geodésicas \mathbf{D}_1 correspondiente al nuevo vértice origen, finalmente las distancias en \mathbf{D} son actua-

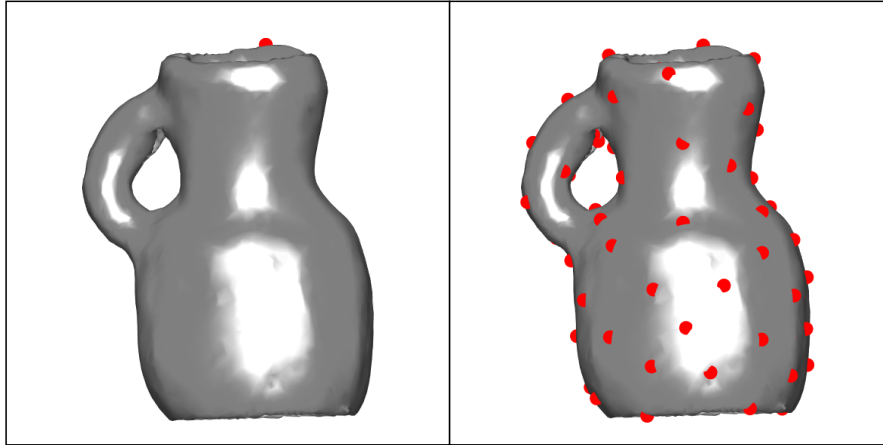


Figura 4-2: A la izquierda se aprecia la cerámica 3D proyectada con el punto inicial (vértice máximo en el eje Y), a la derecha la misma cerámica con 128 puntos obtenidos del cálculo de FPS a partir del punto inicial dado.

lizadas siempre y cuando exista un vértice $\mathbf{v} \in \mathbf{D}_1$ cuya distancia sea menor que la distancia de $\mathbf{v} \in \mathbf{D}$. El bucle termina cuando el tamaño del conjunto solución \mathbf{S} es igual a m .

El algoritmo FPS retorna el conjunto de vértices de modo que estén lo más alejados entre sí, un ejemplo de FPS usando 128 vértices se muestra en la Figura 4-2.

4.2.2. Generación de vistas

Para la generación de vistas, por cada cerámica 3D se cálculo 128 vértices usando FPS. Como resultado se tiene los mapas de distancias geodésicas correspondientes a 128 vértices, esas distancias fueron proyectadas sobre la cerámica

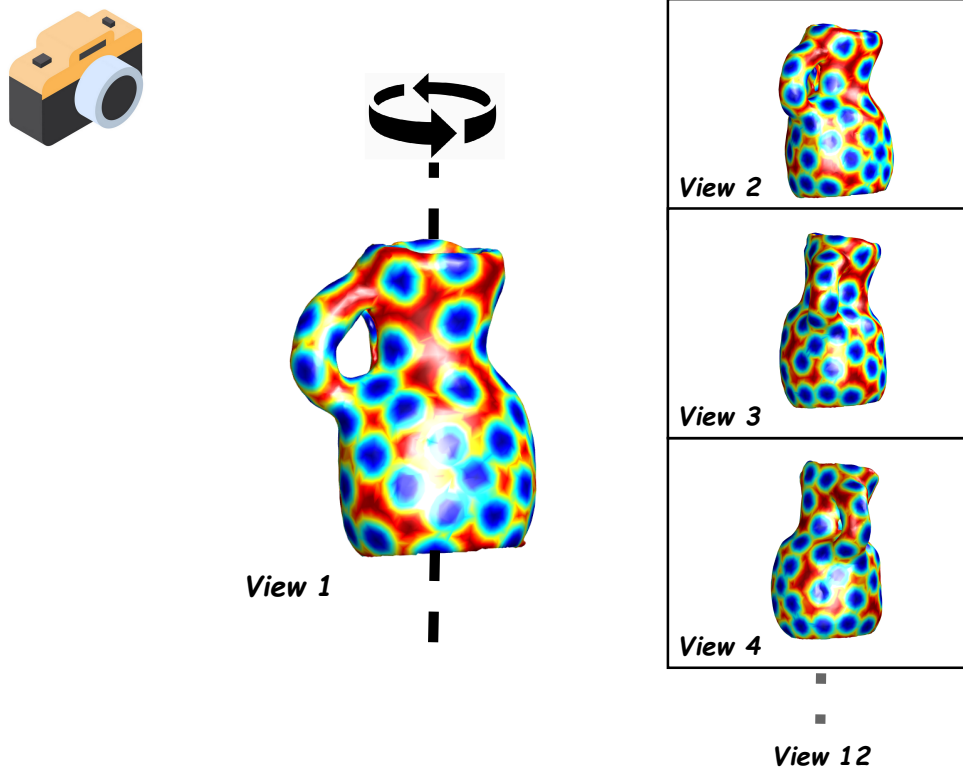


Figura 4-3: Generación de vistas de una cerámica 3D proyectada usando mapas de distancias geodésicas desde 128 fuentes.

3D. Por cada cerámica 3D proyectada con los mapas de distancias se capturó 12 imágenes colocando 12 cámaras alrededor de la cerámica 3D cada 30 grados. Cada imagen captura una vista parcial de la cerámica 3D coloreada con el mapa de distancias geodésicas, la cual es renderizada en una imagen RGB con una dimensión de 224x224 píxeles. Un ejemplo del descriptor propuesto aplicado a una cerámica 3D se aprecia en la (Figura 4-3).

4.3. Proceso de clasificación

Para realizar el proceso de clasificación es necesario agrupar las 12 imágenes previamente generadas en un solo vector característico. En esta tesis para la agrupación de las 12 imágenes se utilizó el enfoque *multiview-CNN*, en donde cada imagen es pasada por una arquitectura CNN. Debido a que no se cuenta con un amplio conjunto de cerámicas 3D, arquitecturas CNNs como Alexnet, ResNet o VGG no pueden garantizar una correcta clasificación. Una alternativa para aplicar arquitecturas CNNs sobre el pequeño conjunto de cerámicas es usando un modelo pre-entrenado con una base de datos más grande y similar, esta técnica de aprendizaje automático es llamada *transfer learning*. Como arquitectura CNN pre-entrenada se escogió VGG-11 cuya entrada es una imagen RGB de 224x224 y como salida se obtiene un vector característico de tamaño 25088 (Figura 4-4).

Posteriormente, los 12 vectores característicos correspondientes a una cerámica 3D son agrupados en un solo vector característico utilizando el concepto de *view pooling*, donde el vector característico con el valor máximo es seleccionado. Finalmente, el vector característico resultante es pasado por un *Multi Layer Perceptron* cuya función es reducir la dimensionalidad y posteriormente clasificar la cerámica 3D (Figura 4-5).

4 Propuesta

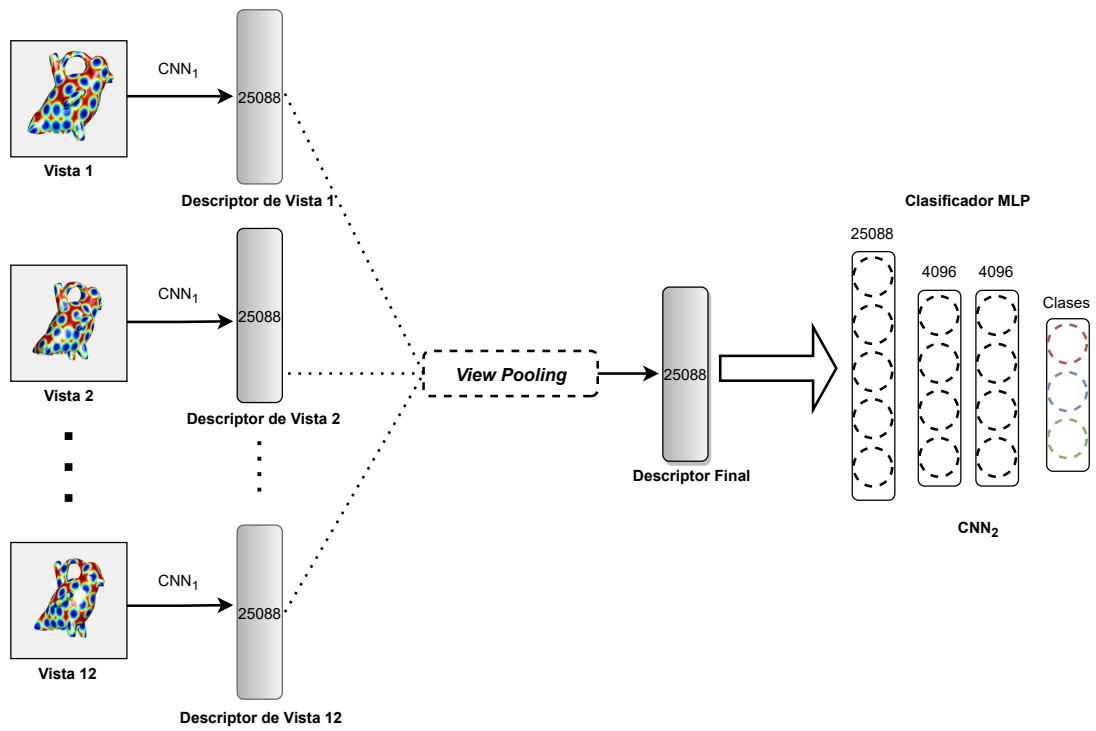


Figura 4-5: Proceso de clasificación a través de la agrupación de cada vector característico.

5.1. Bases de datos

Para la evaluación del descriptor propuesto se utilizó dos conjuntos de datos arqueológicos diferentes.

5.1.1. *Peruvian Dataset*

La base de datos Peruana está conformada por una colección de 938 cerámicas arqueológicas recolectadas de diferentes museos de la ciudad Lima-Perú. El conjunto de datos incluye modelos 3D de diferentes objetos como platos, vasos, ollas entre otros. Una vez recolectadas las cerámicas 3D se comenzó con el proceso de categorización, para lo cual se optó por la creación de una página web simple, con el fin de optimizar esta tarea.

La página web consta de dos interfaces; la primera muestra el conjunto de

5 Experimentos y resultados

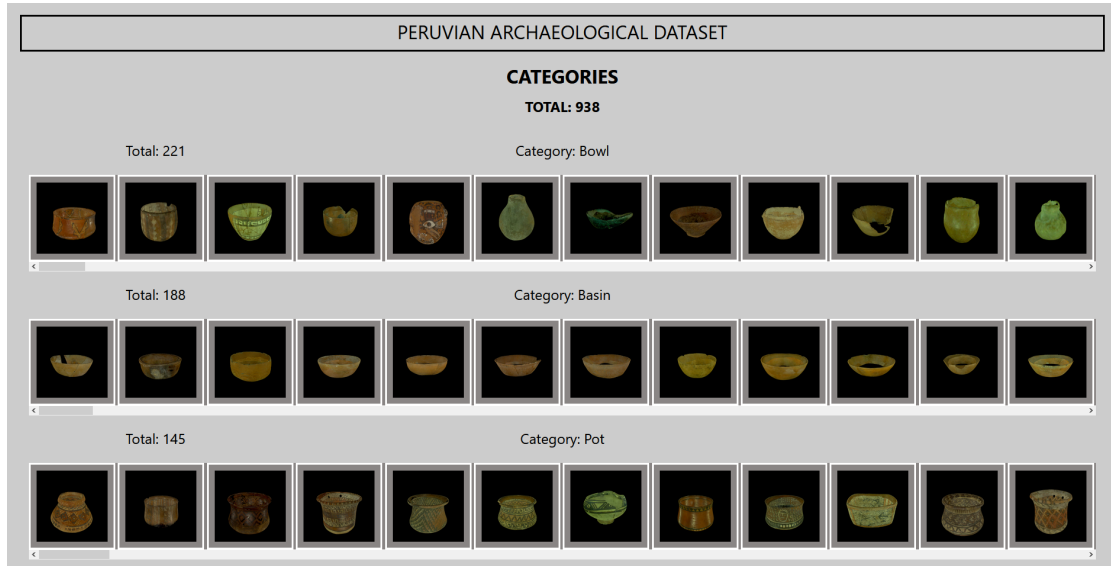


Figura 5-1: Interfaz web inicial para organizar base de datos de cerámicas peruanas por categoría.

cerámicas organizadas por categorías (Figura 5-1) y la segunda muestra la cerámica a categorizar seguida de las diferentes categorías disponibles para seleccionar (Figura 5-2). Finalmente, con la ayuda de arqueólogos y la página web se dividió el conjunto de datos en 8 categorías cuya muestra y distribución se aprecian en las figuras 5-3 y 5-4 respectivamente.

5 Experimentos y resultados

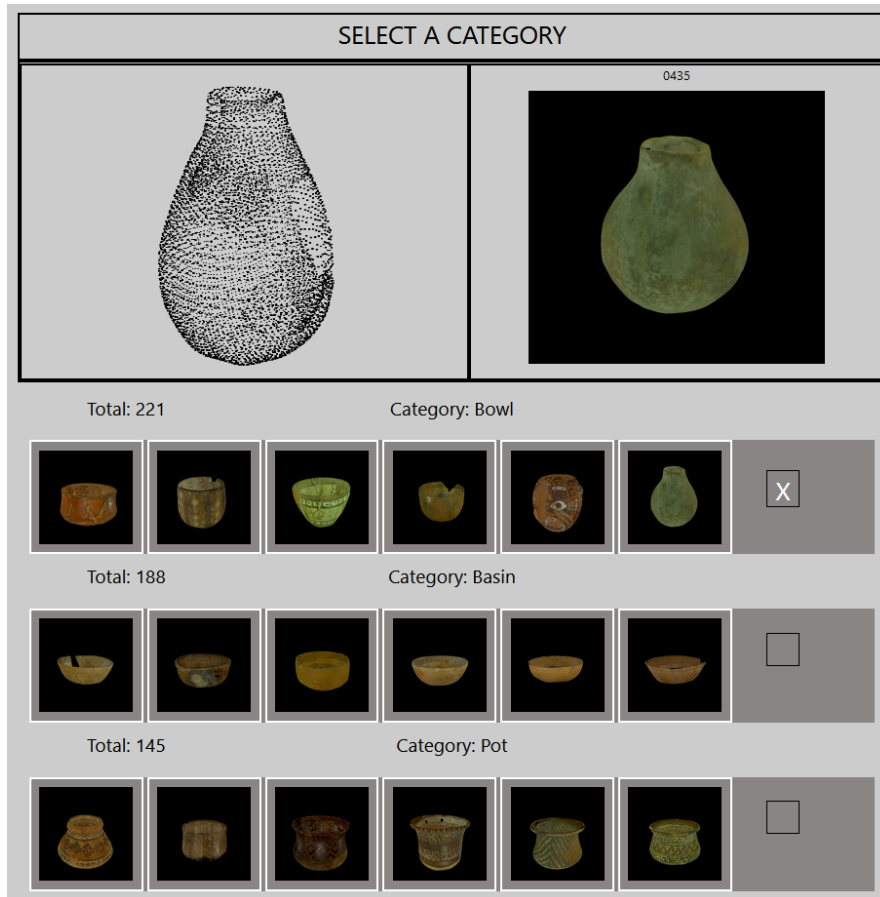


Figura 5-2: Interfaz web para categorizar una cerámica en específico.

5 Experimentos y resultados



Figura 5-3: Muestra aleatoria (una por clase) del conjunto de datos de cerámicas 3D peruanas.

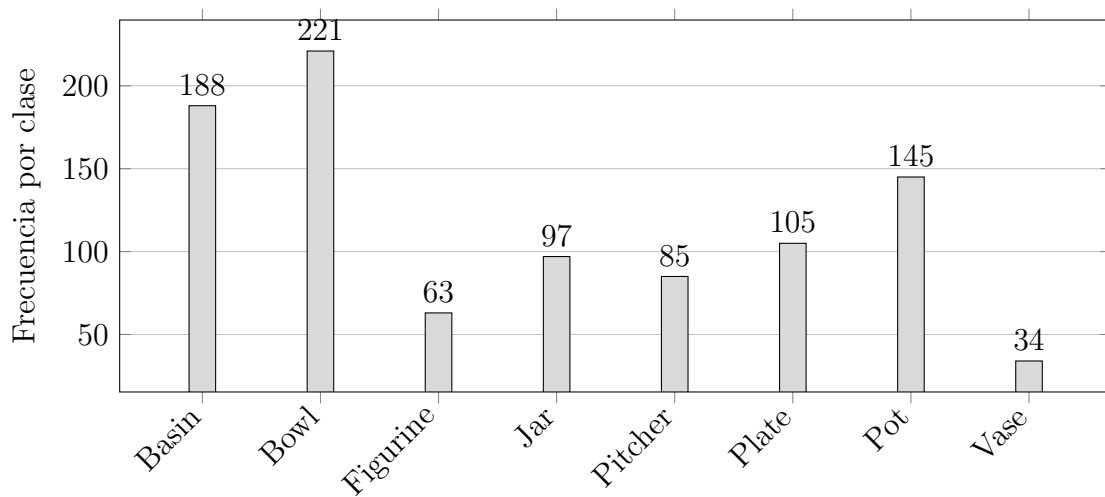


Figura 5-4: Diagrama de barras de la distribución de clases del conjunto de datos Peruano. En la parte superior de cada barra se aprecia el número de cerámicas correspondientes a cada categoría.

5.1.2. 3D Pottery Dataset

Este conjunto de datos es propiedad del museo arqueológico de Hampson en Arkansas [Koutsoudis et al., 2010] el cual contiene 993 objetos, divididos en 23 categorías. Uno de los problemas de esta base de datos es que el número de objetos es muy escaso. El trabajo de [Canul-Ku et al., 2018] resolvió este problema seleccionando 411 objetos divididos en 11 clases. En esta tesis se adoptó dicha partición, cuya muestra y distribución se aprecian en las figuras 5-5 y 5-6 respectivamente.

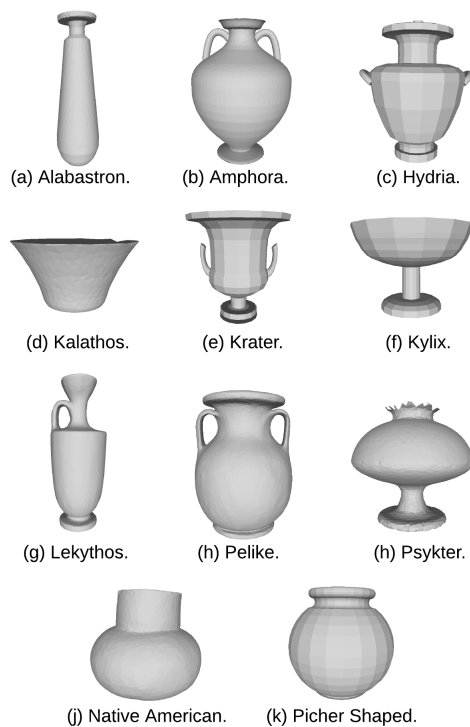


Figura 5-5: Muestra aleatoria de formas elegidas (una por clase) del conjunto de datos *3D Pottery*.

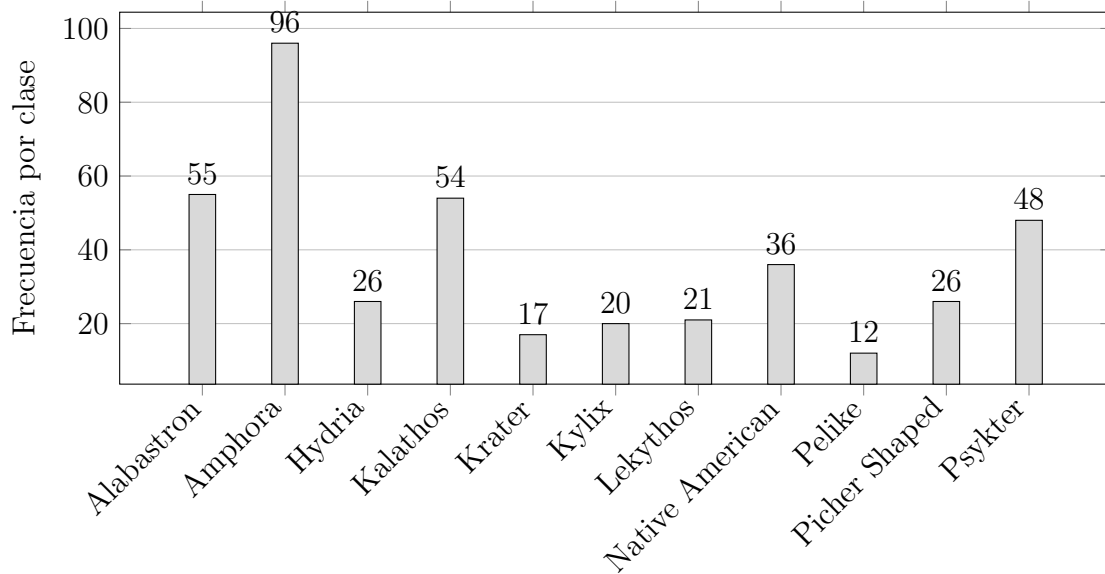


Figura 5-6: Diagrama de barras de la distribución de clases del conjunto de datos *3D Pottery*. En la parte superior de cada barra se aprecia el número de cerámicas correspondientes a cada categoría.

5.2. Criterio de evaluación

El criterio de evaluación está basado en determinar el desempeño del descriptor propuesto, para lo cual se comparó el descriptor con seis métodos de clasificación Voxnet [Maturana and Scherer, 2015], Pointnet [Qi et al., 2017a], DGCNN [Wang et al., 2019], BPS-MLP [Prokudin et al., 2019], MVCNN [Su et al., 2015] y MVCNN-*Curvature* [Canul-Ku et al., 2018] sobre las dos bases de datos de cerámicas 3D las cuales fueron divididas en 70 % para el entrenamiento y el 30 % para pruebas. En cada método de clasificación se utilizó la misma configuración con ligeras modificaciones, las cuales son mencionadas a continuación.

En primer lugar, para Voxnet se convirtió cada malla en una cuadrícula binaria de *voxels* 3D (*binvox*) de tamaño 32x32x32. Seguidamente, para Pointnet, DGCNN y BPS-MLP se convirtió cada malla en una nube de puntos la cual fue reducida uniformemente a 2048 puntos para posteriormente ser normalizada. Para fines de pruebas, se comparó los tres métodos usando la nube de puntos normalizada y sin normalizar (a nivel de escala y posición). En Pointnet y DGCNN se utilizó una tasa de aprendizaje de $1e - 3$ con 250 épocas y en BPS-MLP se redujo la tasa de aprendizaje de $1e - 4$ a $1e - 5$ con 1200 épocas. Respectivamente, para los métodos basados en vistas como MVCNN, MVCNN-Curvature y el método propuesto MVCNN-*Geodesic* se utilizó la arquitectura CNN VGG-11 con los pesos pre-entrenados de *Imagenet* (*transfer learning*), se redujo la tasa de aprendizaje de $5e - 5$ a $5e - 6$ y se aumentó el número de épocas de 30 a 50. Finalmente, con el objetivo de tener una correcta reproducibilidad de los experimentos se configuró una semilla inicial con el valor de cero para la generación de números aleatorios tanto en CPU como en GPU.

5.3. Experimentos

El mejor número de puntos FPS para obtener la más alta precisión de clasificación es una pregunta abierta, por tal motivo en esta tesis se comparó la precisión de clasificación usando diferentes números de puntos FPS (1,2,32,64,128,256,512), además en este experimento se utilizó dos criterios para seleccionar el punto inicial (máximo y mínimo vértice en el eje Y) necesario para el cálculo de FPS. Los

5 Experimentos y resultados

resultados de este experimento se muestran en la Tabla **5-1**, en donde se puede visualizar que para la base de datos *3D Pottery* la más alta precisión (97,56 %) fue obtenida cuando se utiliza como punto inicial el vértice máximo del eje *Y* y 1 punto FPS, por otro lado, en la base de datos *Peruvian* la más alta precisión (82,67 %) fue obtenida cuando se utiliza el vértice máximo y 128 o 512 puntos FPS.

Por otro lado, en la Tabla **5-2** se comparó el método MVCNN-*Geodesic* con otros métodos de clasificación, en donde, se puede apreciar que el método propuesto obtiene la mejor puntuación tanto en la base de datos *Peruvian* con 82,67 % de precisión como en la base de datos *Pottery* con 97,56 %. Además, los experimentos realizados muestran una clara mejora en todos los métodos basados en nubes de puntos cuando esta se encuentra normalizada.

Precisión de Clasificación			
Punto Inicial	Nro. de Puntos FPS	<i>3D Pottery</i>	<i>Peruvian</i>
max	1	97.56 %	80.51 %
max	2	96.75 %	81.59 %
max	32	95.12 %	80.51 %
max	64	94.31 %	81.59 %
max	128	95.93 %	82.67 %
max	256	95.93 %	81.95 %
max	512	95.93 %	82.67 %
min	1	96.75 %	79.42 %
min	2	96.75 %	81.23 %
min	32	95.12 %	81.59 %
min	64	95.12 %	81.23 %
min	128	96.75 %	81.59 %
min	256	95.93 %	80.14 %
min	512	95.12 %	81.95 %

Tabla 5-1: Comparación de la precisión de clasificación obtenida del método propuesto (MVCNN-*Geodesic*) con diferente número de puntos FPS utilizando como punto inicial el vértice máximo y mínimo del eje Y.

Método	<i>3D Pottery</i>	<i>Peruvian</i>
Voxnet [Prokudin et al., 2019]	94.30 %	74.01 %
Pointnet [Qi et al., 2017a]	91.06 %	76.17 %
Pointnet non_normalized [Qi et al., 2017a]	87.8 %	71.84 %
DGCNN [Wang et al., 2019]	93.5 %	77.62 %
DGCNN non_normalized [Wang et al., 2019]	95.12 %	77.62 %
BPS-MLP [Prokudin et al., 2019]	95.12 %	81.95 %
BPS-MLP non_normalized [Prokudin et al., 2019]	92.68 %	79.06 %
MVCNN [Su et al., 2015]	96.75 %	80.87 %
MVCNN- <i>Curvature</i> [Canul-Ku et al., 2018]	97.56 %	81.23 %
MVCNN-<i>Geodesic</i>	97.56 %	82.67 %

Tabla 5-2: Comparación de la precisión de clasificación obtenida del método propuesto (MVCNN-*Geodesic*) con otros métodos de clasificación sobre los dos conjuntos de datos de cerámicas arqueológicas (*Peruvian* y *3D Pottery*). Para el método propuesto MVCNN-*Geodesic* se presentó el porcentaje de precisión más alto obtenido de la Tabla 5-1.

5.4. Discusión

Con el objetivo de entender mejor el descriptor propuesto se comparó las matrices de confusión de MVCNN-*Geodesic* con el método base MVCNN, además se hizo un análisis de las categorías en las que el método propuesto falla.

En la Figura 5-7 se muestra las matrices de confusión correspondientes a la base de datos *3D Pottery* en la que se aprecia que el método propuesto mejora la predicción de la categoría *Krater*.

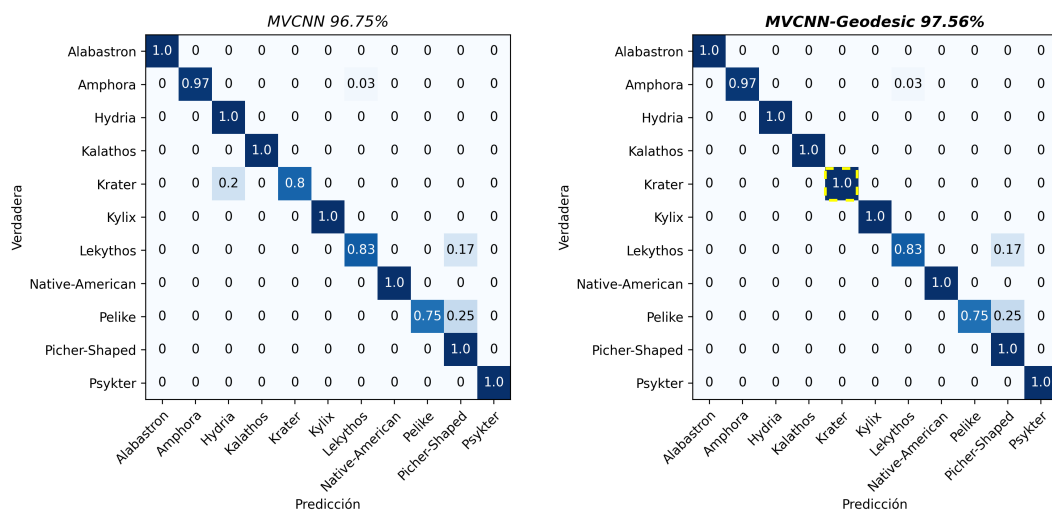


Figura 5-7: Comparación de las matrices de confusión correspondientes al método MVCNN y el método propuesto (MVCNN-*Geodesic*) usando el conjunto de pruebas de *Pottery Dataset*. Para generar la matriz de confusión de MVCNN-*Geodesic* se utilizó el experimento max-1 de la tabla 5-1.

5 Experimentos y resultados

Por otro lado, en la Figura 5-8 se aprecia la comparación correspondiente a la base de datos *Peruvian* en donde se aprecia que MVCNN-*Geodesic* obtuvo una mejora de 1,8% con respecto a MVCNN, además las categorías que aumentaron su porcentaje de predicción se señalan en color amarillo (*Jar*, *Pot* y *Vase*), y las categorías que se redujeron en color verde (*Bowl*, *Plate*).

A pesar de que en la base de datos *Peruvian* el descriptor propuesto tuvo un buen impacto, esta base de datos aún presenta varios desafíos, en la Figura 5-9 se puede apreciar cinco categorías las cuales se confunden con mayor frecuencia como es el caso de *Vase*, *Pot*, *Plate*, *Jar* y *Pitcher*, se puede inferir que esto es debido a la similitud que tiene cada cerámica de una categoría con respecto a otra como por ejemplo *Jar* con *Pitcher*.

5 Experimentos y resultados

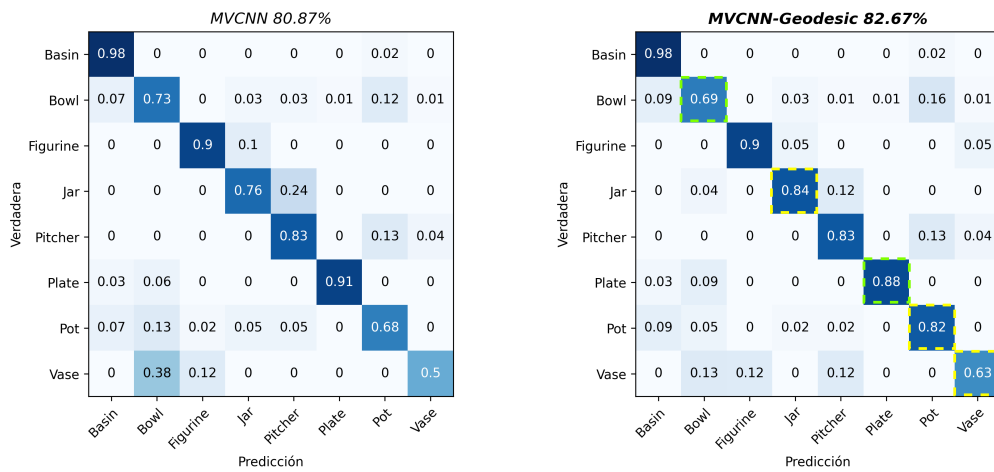


Figura 5-8: Comparación de las matrices de confusión correspondientes al método MVCNN y el método propuesto (MVCNN-Geodesic) usando el conjunto de pruebas de *Peruvian Dataset*. Para generar la matriz de confusión de MVCNN-Geodesic se utilizó el experimento max-128 de la tabla 5-1.

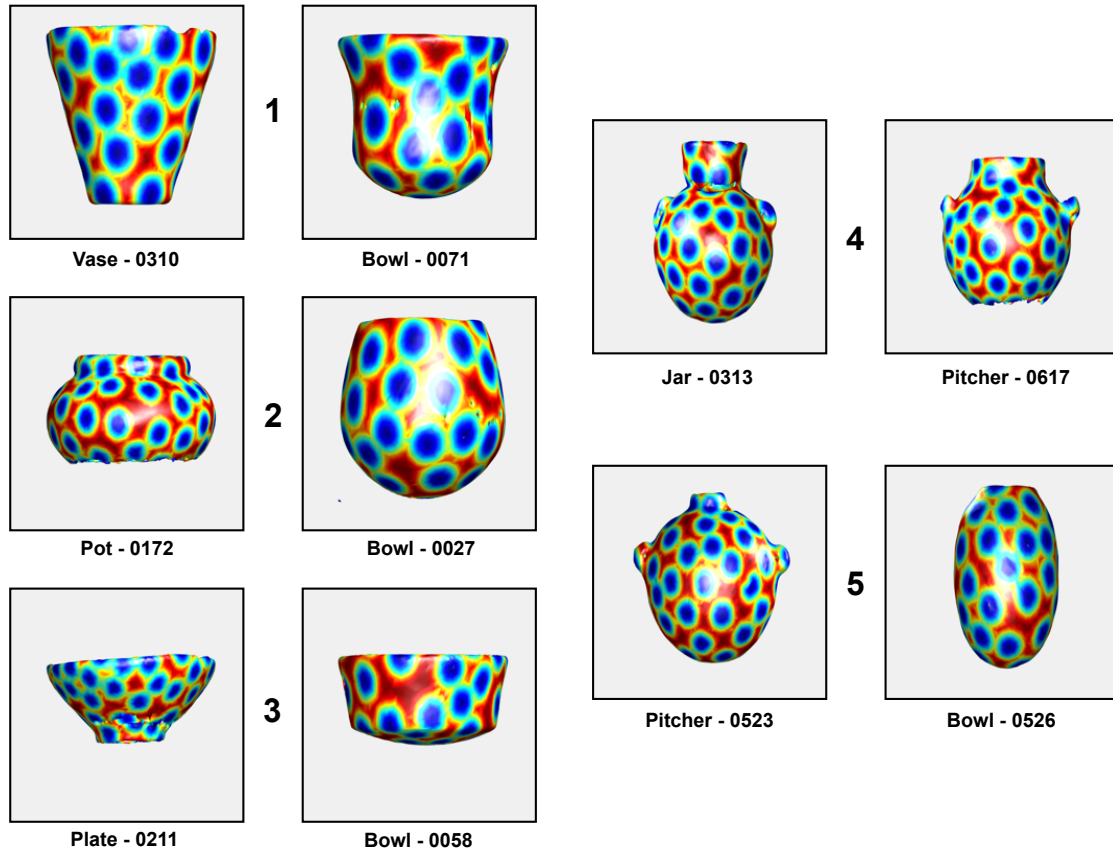


Figura 5-9: Análisis de las categorías que se confunden con mayor frecuencia pertenecientes a la base de datos *Peruvian*, en donde cada categoría está representada por una cerámica proyectada usando el descriptor propuesto.

5.5. Código fuente

El código fuente de este trabajo de investigación se encuentra disponible en <https://github.com/patrick100/classification-pottery>, libre para su descarga y futuras investigaciones.

6.1. Conclusiones

En la presente investigación se ha propuesto un descriptor basado en la combinación de distancias geodésicas con la arquitectura MVCNN para la clasificación de cerámicas arqueológicas.

Se recolectó dos bases de datos de cerámicas arqueológicas *Peruvian* y *3D Pottery*, las cuales fueron normalizadas a nivel de escala, posición y orientación.

El método propuesto fue comparado con seis métodos de clasificación pertenecientes al estado del arte usando dos bases de datos de cerámicas arqueológicas 3D, de donde se puede concluir que, el método propuesto, MVCNN-*Geodesic*, obtuvo la mejor precisión para las dos bases de datos, la cual fue de 82,67% para *Peruvian* y de 97,56% para *3D Pottery*.

Con el objetivo de probar la robustez de los métodos de clasificación basados en nube de puntos (Pointnet, DGCNN y BPS), se comparó cada uno de ellos usando la nube sin normalizar a nivel de escala y posición, de donde se puede concluir que tanto Pointnet como BPS necesitan que la nube de puntos sea previamente normalizada, sin embargo para DGCNN no es necesario.

6.2. Trabajos Futuros

- Aumentar las vistas generadas ubicando las cámaras alrededor de una esfera, esto con el fin de obtener un vector característico más robusto.
- Seleccionar, del conjunto de datos de *Imagenet* [Deng et al., 2009], las categorías relacionadas con cerámicas. Esto con el fin de mejorar la inicialización de los pesos en la fase de entrenamiento.

- [Ahmed et al., 2018] Ahmed, E., Saint, A., Shabayek, A. E. R., Cherenkova, K., Das, R., Gusev, G., Aouada, D., and Ottersten, B. (2018). Deep learning advances on different 3d data representations: A survey. *arXiv preprint arXiv:1808.01462*.
- [Barroso-Laguna et al., 2019] Barroso-Laguna, A., Riba, E., Ponsa, D., and Mikolajczyk, K. (2019). Key. net: Keypoint detection by handcrafted and learned cnn filters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5836–5844.
- [Ben-Shabat et al., 2017] Ben-Shabat, Y., Lindenbaum, M., and Fischer, A. (2017). 3d point cloud classification and segmentation using 3d modified fisher vector representation for convolutional neural networks. *arXiv preprint arXiv:1711.08241*.
- [Benhabiles and Tabia, 2016] Benhabiles, H. and Tabia, H. (2016). Convolu-

Bibliografía

- tional neural network for pottery retrieval. *Journal of Electronic Imaging*, 26(1):011005.
- [Bronstein et al., 2008] Bronstein, A. M., Bronstein, M. M., and Kimmel, R. (2008). *Numerical geometry of non-rigid shapes*. Springer Science & Business Media.
- [Bruna et al., 2013] Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.
- [Canul-Ku et al., 2018] Canul-Ku, M., Hasimoto-Beltran, R., Jiménez-Badillo, D., Ruiz-Correa, S., and Román-Rangel, E. (2018). Classification of 3d archaeological objects using multi-view curvature structure signatures. *IEEE Access*, 7:3298–3313.
- [Cintas et al., 2020] Cintas, C., Lucena, M., Fuertes, J. M., Delrieux, C., Navarro, P., Gonzalez-Jose, R., and Molinos, M. (2020). Automatic feature extraction and classification of iberian ceramics based on deep convolutional networks. *Journal of Cultural Heritage*, 41:106–112.
- [Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- [Eldar et al., 1997] Eldar, Y., Lindenbaum, M., Porat, M., and Zeevi, Y. Y. (1997).

Bibliografía

- The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315.
- [Feng et al., 2019] Feng, Y., Feng, Y., You, H., Zhao, X., and Gao, Y. (2019). Meshnet: mesh neural network for 3d shape representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8279–8286.
- [Hanocka et al., 2019] Hanocka, R., Hertz, A., Fish, N., Giryas, R., Fleishman, S., and Cohen-Or, D. (2019). Meshcnn: a network with an edge. *ACM Transactions on Graphics (TOG)*, 38(4):1–12.
- [Hermes, 1991] Hermes, B. (1991). Propuesta para la clasificación de artefactos cerámicos en contexto arqueológico. *Mayab*, (7):5–9.
- [Hermoza and Sipiran, 2018] Hermoza, R. and Sipiran, I. (2018). 3d reconstruction of incomplete archaeological objects using a generative adversarial network. In *Proceedings of Computer Graphics International 2018*, pages 5–11. ACM.
- [Hou et al., 2020] Hou, Y., Cui, X., Canul-Ku, M., Jin, S., Hasimoto-Beltran, R., Guo, Q., and Zhu, M. (2020). Admorph: a 3d digital microfossil morphology dataset for deep learning. *IEEE Access*, 8:148744–148756.
- [Huang et al., 2018] Huang, J., Su, H., and Guibas, L. (2018). Robust watertight manifold surface generation method for shapenet models. *arXiv preprint arXiv:1802.01698*.
- [Hussain et al., 2018] Hussain, M., Bird, J. J., and Faria, D. R. (2018). A study on

- cnn transfer learning for image classification. In *UK Workshop on computational Intelligence*, pages 191–202. Springer.
- [Kazmi et al., 2013] Kazmi, I. K., You, L., and Zhang, J. J. (2013). A survey of 2d and 3d shape descriptors. In *2013 10th International Conference Computer Graphics, Imaging and Visualization*, pages 1–10. IEEE.
- [Kimmel and Sethian, 1998] Kimmel, R. and Sethian, J. A. (1998). Computing geodesic paths on manifolds. *Proceedings of the national academy of Sciences*, 95(15):8431–8435.
- [Koutsoudis et al., 2010] Koutsoudis, A., Pavlidis, G., Liami, V., Tsiafakis, D., and Chamzas, C. (2010). 3d pottery content-based retrieval based on pose normalisation and segmentation. *Journal of Cultural Heritage*, 11(3):329–338.
- [Li et al., 2018] Li, Y., Bu, R., Sun, M., Wu, W., Di, X., and Chen, B. (2018). Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, pages 820–830.
- [Maturana and Scherer, 2015] Maturana, D. and Scherer, S. (2015). Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE.
- [Mitchell et al., 1987] Mitchell, J. S., Mount, D. M., and Papadimitriou, C. H. (1987). The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668.

Bibliografia

- [Prokudin et al., 2019] Prokudin, S., Lassner, C., and Romero, J. (2019). Efficient learning on point clouds with basis point sets. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0.
- [Qi et al., 2017a] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660.
- [Qi et al., 2016] Qi, C. R., Su, H., Nießner, M., Dai, A., Yan, M., and Guibas, L. J. (2016). Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656.
- [Qi et al., 2017b] Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108.
- [Samet, 1984] Samet, H. (1984). The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2):187–260.
- [Scopigno et al., 2011] Scopigno, R., Callieri, M., Cignoni, P., Corsini, M., Dellepiane, M., Ponchio, F., and Ranzuglia, G. (2011). 3d models for cultural heritage: Beyond plain visualization. *Computer*, 44(7):48–55.
- [Su et al., 2015] Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. (2015).

Bibliografia

- Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953.
- [Tangelder and Veltkamp, 2008] Tangelder, J. W. and Veltkamp, R. C. (2008). A survey of content based 3d shape retrieval methods. *Multimedia tools and applications*, 39(3):441–471.
- [Tatarchenko et al., 2017] Tatarchenko, M., Dosovitskiy, A., and Brox, T. (2017). Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2088–2096.
- [Wang et al., 2019] Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2019). Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):1–12.
- [Wu et al., 2015] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920.
- [Ying, 2019] Ying, X. (2019). An overview of overfitting and its solutions. In *Journal of Physics: Conference Series*, volume 1168, page 022022. IOP Publishing.
- [Zhang et al., 2017] Zhang, J., Xia, Y., Xie, Y., Fulham, M., and Feng, D. D. (2017). Classification of medical images in the biomedical literature by jointly

Bibliografia

using deep and handcrafted visual features. *IEEE journal of biomedical and health informatics*, 22(5):1521–1530.