# Análisis morfométrico y clasificación de granos de cebada mediante armónicos esféricos y deep learning

Javier Esmoris Cerezuela

1 de julio de 2024

**Resumen–** En este trabajo se explora el análisis morfométrico de granos de cebada a través de su representación en armónicos esféricos. Se tratará en detalle el alineamiento de granos, la teoría que fundamenta los armónicos esféricos y sus limitaciones en su capacidad de representación de objetos arbitrarios. Con esta herramienta matemática se desarrolla una metodología que permite realizar operaciones entre objetos punto a punto, y reducir la resolución de los objetos conservando su forma original. Con la representación en armónicos esféricos se entrenarán dos modelos para cuatro experimentos de clasificación, con el objetivo de predecir características de los granos de cebada únicamente con la información de su superficie, tales como razas y lugares de crecimiento. Estos experimentos pretenden demostrar la eficacia de los armónicos esféricos y el Machine Learning en la clasificación morfométrica, así como explorar su potencial para aplicaciones futuras.

**Palabras clave–** Machine Learning, Deep Learning, Armónicos esféricos, Random Forest, MultiLayer Perceptron, Clasificación 3D, Cebada.

**Abstract–** In this work, the morphometric analysis of barley grains is explored through their representation in spherical harmonics. The alignment of grains, the theory underlying spherical harmonics, and their limitations in representing arbitrary objects will be discussed in detail. With this mathematical tool, a methodology is developed in order to allow point-to-point operations between objects and reduce the resolution of objects while preserving their original shape. Two models will be trained for four classification experiments using the representation in spherical harmonics, aiming to predict characteristics of barley grains, such as breeds and growth location, using only their surface information. These experiments aim to demonstrate the effectiveness of spherical harmonics and Machine Learning in morphometric classification and explore their potential for future applications.

**Keywords–** Machine Learning, Deep Learning, Spherical Harmonics, Random Forest, Multi-Layer Perceptron, 3D Classification, Barley.

## 1 Introducción

La morfometría es el análisis cuantitativo de la forma de un objeto y es una técnica que ha ganado popularidad en las últimas dos décadas dentro de la arqueología [1, 2, 3]. El análisis morfométrico está directamente ligado con la estructura usada para representar un objeto, por lo que es de gran importancia elegir cuál se va a utilizar. Hay muchas formas de representar un objeto, como por ejemplo mediante distintas secciones 2D, un vector de características físicas medibles, una nube de puntos, una malla, un conjunto de vóxeles o árboles octales.

En este trabajo se explora el análisis morfométrico aplicado a granos de cebada con la ayuda de una representación en armónicos esféricos y su posible uso en experimentos de clasificación. Es una pequeña parte de un proyecto de investigación llevado a cabo por parte del ICAC[1], el Insti-

• E-mail de contacto: javi.esmoris@gmail.com
• Mención realizada: Computación
• Trabajo tutorizado per: Felipe Lumbreras Ruiz (Computer Science)
• Curso 2023/24

[1] www.icac.cat

tuto Catalán de Arqueología Clásica, con la colaboración del Centro de Visión por Computador de la Universitat Autònoma de Barcelona, proyecto en el que he colaborado desarrollando los resultados que presento en este trabajo.

Este documento está estructurado como sigue. En primer lugar, se presenta el estado del arte, donde se muestran dos estudios que han trabajado la clasificación de objetos 3D. A continuación, se introducirán las herramientas matemáticas e informáticas necesarias para este proyecto. Seguidamente se hablará del dataset de granos de cebada usado en este trabajo. Acto seguido, se expondrá el desarrollo del proyecto, que abarca el alineamiento de granos, su descomposición usando armónicos esféricos y recomposición de estos en un dominio uniforme. Posteriormente, se detallarán los experimentos realizados, se mostrarán los resultados obtenidos y se extraerán unas conclusiones finales.

## 2 ESTADO DEL ARTE

A continuación se presentan dos estudios que han trabajado la clasificación de objetos 3D desde dos perspectivas diferentes. El primero usa representaciones en superficies poligonales (también llamadas mallas), las mismas que conforman nuestro dataset, mientras que el segundo utiliza armónicos esféricos calculados a partir de una nube de puntos.

### 2.1. MeshCNN

MeshCNN [4] es una arquitectura de red neuronal convolucional diseñada específicamente para mallas. A diferencia de las redes convolucionales tradicionales, que están optimizadas para trabajar con imágenes 2D estructuradas en una cuadrícula regular, MeshCNN se adapta para procesar estructuras con geometrías y topologías diferentes, es decir, número de vértices y caras distintos, posibles agujeros o tamaños distintos de objetos.

Esta arquitectura realiza convoluciones directamente sobre las aristas de la malla, operando sobre su estructura intrínseca en lugar de evadir el problema con una representación en cuadrícula. MeshCNN implementa operaciones de pooling y unpooling específicas para mallas, así permite reducir y aumentar la complejidad de la malla mientras se conserva la información relevante para la clasificación, como se ve en la figura 1.

Fig. 1: Reducción de complejidad de un mesh *(fuente: R. Hanocka, et al., "Meshcnn: a network with an edge")*

### 2.2. Robust Object Classification Approach using Spherical Harmonics

Este paper [5] presenta un nuevo enfoque para la clasificación de objetos 3D utilizando armónicos esféricos. Este enfoque trata las dificultades que implican las nubes de puntos generadas por escáneres 3D o imágenes multivista, que suelen contener ruido y valores extremos. El objetivo principal es desarrollar un método que mantenga la precisión en la clasificación, incluso en presencia de imperfecciones en los datos.

El proceso comienza con el muestreo de la señal de entrada en un grid de coordenadas esféricas, lo que permite una distribución uniforme de los puntos sobre la bola unitaria perfecta para integrar sobre ella y calcular los coeficientes de cada armónico.

Para la clasificación, se realizan convoluciones esféricas en el dominio de los armónicos. Esto implica la multiplicación de los coeficientes de Fourier de la función con los del kernel, que es la aplicación directa del teorema de la convolución en un dominio esférico. La limitación del nivel de orden de los armónicos esféricos ayuda a suprimir el ruido y los valores atípicos, ya que como veremos más adelante, tener más armónicos no implica una mejor representación del objeto.

El enfoque basado en armónicos esféricos propuesto en este paper demuestra ser una solución efectiva y eficiente para la clasificación robusta de objetos 3D, así como para aplicaciones en las que los datos inevitablemente contendrán ruido.

## 3 HERRAMIENTAS

En esta sección se describen las herramientas tanto matemáticas como informáticas para llevar a cabo este trabajo.

### 3.1. Armónicos esféricos

Los armónicos esféricos [5, 6] son un conjunto de funciones ortonormales definidas sobre la esfera unidad $S^2$ de la siguiente forma:

$$Y_l^m(\theta,\varphi) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(cos\theta) e^{i\,m\,\varphi}, \quad (1)$$

donde $P_l^m(x)$ es el polinomio de Legendre asociado, $l \geq 0$ es el grado de la frecuencia, $m \leq |l|$ es el orden de cada frecuencia, $\theta \in [0,\pi]$ es el ángulo cenital y $\varphi \in [0,2\pi]$ es el ángulo azimutal.

Las funciones de cuadrado integrable definidas sobre $S^2$ generan un espacio vectorial que, junto con el producto escalar inducido (2), definen un espacio de Hilbert del cual los armónicos esféricos son una base.

$$\langle f,g\rangle = \int_0^{2\pi} d\varphi \int_0^{\pi} d\theta \sin(\theta) f(\theta,\varphi) g(\theta,\varphi). \quad (2)$$

Esto permite expresar una función $f$ de cuadrado integrable definida en $S^2$ como una combinación lineal única e

infinita de armónicos esféricos (4). Los coeficientes de la combinación lineal se calculan como el producto escalar de la función con los armónicos (3).

$$f_l^m = \langle f, Y_l^m \rangle, \tag{3}$$

$$f(\theta, \varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} f_l^m Y_l^m(\theta, \varphi). \tag{4}$$

En la figura 2 se muestran algunos armónicos esféricos. Esta representación se obtiene evaluando cada vértice de una esfera y asignándole el valor obtenido como nuevo radio en coordenadas esféricas.

(a) $l = 0, m = 0$ (b) $l = 2, m = 0$ (c) $l = 8, m = 0$

(d) $l = 2, m = 1$ (e) $l = 4, m = -3$ (f) $l = 6, m = 6$

Fig. 2: Armónicos esféricos

## 3.2. Modelos

Los modelos de clasificación usados en este trabajo son Random Forest (RF) y Multilayer Perceptron (MLP).

RF [7] es un modelo de aprendizaje supervisado que genera árboles de decisión de forma aleatoria para mejorar la precisión y evitar el sobreajuste. RF crea múltiples árboles de decisión en el entrenamiento y promedia las predicciones de estos árboles para tomar decisiones, ya sean para regresión o clasificación de muestras. Cada árbol se entrena con un subconjunto aleatorio de datos y características, lo que aumenta la diversidad y reduce la correlación entre árboles. Esta técnica mejora sustancialmente la eficacia de los árboles de decisión, el manejo de datos de alta dimensionalidad y la resistencia al ruido. La figura 3 ilustra una versión simplificada del modelo.

MLP [8] es un tipo de red neuronal formada por múltiples capas, donde cada capa está completamente conectada a la siguiente. Al igual que RF, es un modelo de aprendizaje supervisado usado en problemas de regresión y clasificación. La estructura típica incluye una capa de entrada, una o más capas ocultas seguidas de activaciones no lineales como ReLU o Sigmoid, y una capa de salida. Durante el entrenamiento, el algoritmo backpropagation ajusta los pesos de las conexiones minimizando el error entre las predicciones y los valores reales. La figura 4 ilustra la estructura de un MLP y las capas que se pueden encontrar.

Fig. 3: Random forest *(fuente: D. Segura, E. Khatib, and R. Barco, "Dynamic packet duplication for industrial urllc")*

Fig. 4: Multilayer perceptron

## 3.3. Software

En este proyecto los datos que se manipulan principalmente son mallas. Una malla es una estructura de datos que representa un objeto geométrico mediante una lista de vértices, que define la nube de puntos del objeto, y una lista de caras, que cubre la superficie del objeto y lo dota de una topología. Permiten representar de manera precisa y manipulable formas tridimensionales complejas.

### MeshLab

MeshLab [9] es un software para visualizar, crear y editar mallas. En especial se usarán dos algoritmos que ofrece este programa. El primero es Poisson Disk Sampling [10], una técnica que permite distribuir puntos de manera uniforme y no aleatoria en un espacio, asegurando que cada punto está a una distancia mínima específica de los demás. El segundo algoritmo es Ball Pivoting [11], que permite generar una superficie tridimensional dada una nube de puntos.

### Trimesh

Trimesh [12] es una librería de Python que permite trabajar con mallas. Se ha usado extensamente a lo largo de este trabajo para cargar, guardar, crear y modificar mallas.

### Scikit-learn

Scikit-learn [13] es una librería de Python ampliamente usada en machine learning. En este trabajo se ha usado para

los modelos random forest y multilayer perceptron, para hacer PCAs y particionar los datos en train, validation y test.

### SciPy

SciPy [14] es una librería de Python que proporciona herramientas y algoritmos avanzados para matemáticas, ciencia e ingeniería. Se ha usado para evaluar los armónicos esféricos.

## 4 Dataset

Hemos partido con el dataset Northern European Barley Dataset (NEBD) [1], que consiste en 565 granos de cebada con las siguientes distinciones:

- 2-ROW/6-ROW: Número de carreras de las espigas. En el dataset hay granos de 2 y de 6 carreras.
- Dundee/Orkney: Lugares de crecimiento de la cebada.
- British/Scotish/Scandinavian/Bere: Razas de cebada cultivadas.

Además de las 565 semillas iniciales de NEBD, se han añadido 142 semillas nuevas, formando un total de 707 (NEBD+). La distribución de características de los granos se puede ver en la figura 5.

Fig. 5: Distribución de características *(fuente: ICAC)*

Los granos vienen representados por mallas generadas por un escáner 3D. Se pueden encontrar mallas de 120k vértices y 40k caras hasta 175k vértices y 87k caras, dependiendo del tamaño y la rugosidad de la semilla. Esto presenta un problema pues cada semilla tiene un dominio distinto. En la figura 6 se puede observar la malla de una semilla del dataset.

Fig. 6: Malla del grano DHBT16T95 4

## 5 Desarrollo

En esta sección se aborda en detalle los métodos llevados a cabo para la alineación de los granos, su descomposición en armónicos esféricos y la recomposición del dataset con un dominio uniforme.

### 5.1. Alineamiento

El dataset producido a partir del escáner 3D se debe preprocesar debido a que las mallas resultantes están orientadas de formas distintas y alejadas del origen. En la figura 7 se pueden ver unos pocos granos sin procesar. Con el objetivo de ver los granos sin que se superpongan, se han distribuido en forma de tabla para la figura.

Fig. 7: Granos sin procesar

Estas diferencias en posición y orientación introducen información irrelevante que puede inducir efectos indeseables a la hora de clasificar los granos, por lo que primero se deben alinear todos de la misma forma. A continuación se describirán los pasos para el alineamiento de un grano cualquiera:

#### Centrado

El primero consiste en trasladar su centro de masas al origen. Esto se consigue fácilmente restando a cada uno de los vértices de la malla el centro de masas. El centro de masas no es necesario calcularlo dado que la librería Trimesh recalcula los atributos de una malla en cada transformación de esta.

#### Alineamiento sobre los ejes

El segundo paso alinea el grano acorde con los tres ejes. Para ello se hace un Análisis de Componentes Principales (PCA) de los vértices del grano. Esto permite obtener una base ortonormal a partir de los vectores propios de la matriz de covarianza de los vértices. El método PCA ordena los ejes por importancia según las varianzas. La primera componente corresponde al eje alargado del grano, mientras que la segunda y la tercera componente no siempre se ordenan de la misma forma debido a la gran similitud que tienen. Una vez obtenidas estas direcciones, rotamos el grano de forma que la primera componente principal se superponga con el eje $x$, la segunda con el eje $y$ y la tercera con el eje $z$. Esta reordenación de los vectores propios puede introducir una simetría indeseada, pero se puede detectar fácilmente

si el determinante de la matriz de vectores propios es negativo. En este caso basta con cambiar el signo de alguno de los vectores. Dado que las componentes segunda y tercera no se ordenan siempre de la misma forma, queda por determinar si hace falta una rotación de 180º para alinear los ejes $y$ y $z$.

**Alineamiento eje y**

El tercer paso calcula la rotación respecto al eje $x$ necesaria para posicionar la segunda componente principal sobre el eje $y$. Para ello se calculan cinco cortes del grano perpendiculares al eje $x$. Para cada corte se convierten las dos coordenadas no constantes a un sistema de coordenadas polares y se encuentra el punto con el radio menor. De esta forma aislamos la posición relativa del surco y determinamos si la rotación de 180º es necesaria. Se hacen cinco cortes para evitar el posible ruido en la superficie de los granos.

**Alineamiento eje z**

El cuarto paso consiste en determinar si hace falta una rotación de 180º respecto el eje $z$. Siguiendo con la estrategia de cortes del paso anterior, y aprovechando que el surco ya está alineado, el corte que se realiza es el de la barriga que presenta el lado opuesto del surco. A partir del signo de la media de las coordenadas de la sección de la barriga del grano se calcula si hace falta la rotación o no. Para este último paso el resultado ha sido el mismo usando tanto uno como más cortes, y ha funcionado para el 97 % de los granos.

**Revisión manual**

Dado que el alineamiento sobre el eje $z$ no ha funcionado en todos los casos, es necesaria una última revisión manual de los granos para corregir aquellos que no acabaron de orientarse bien. Para agilizar este proceso se han renderizado todos los granos y se han distribuido en el espacio en forma de tabla. Esto permite reconocer a simple vista y de forma muy rápida los granos mal orientados.

La figura 8 muestra parte del dataset orientado y la figura 9 ilustra los pasos de alineación descritos.

Fig. 8: Granos orientados

Fig. 9: Proceso de alineación: a) centrado, b) rotación de ejes mediante PCA, c) rotación según el surco detectado con secciones en coordenadas polares, d) rotación según la asimetría de la barriga detectada con una sección

## 5.2. Descomposición en armónicos esféricos

Dado un grano de cebada, si expresamos su superficie en coordenadas esféricas y consideramos el radio como una función esférica $r = f(\theta, \varphi)$, podemos expresarla como una combinación lineal de armónicos esféricos y obtener una representación analítica independiente del número de vértices y caras de la malla. Esto último será de gran utilidad en la reconstrucción que veremos en la sección 5.3.

Como la base de armónicos es infinita, la combinación lineal para representar la superficie de un grano también. Para solventar esta infinitud, establecemos un valor $l_{max}$ para truncar la descomposición y quedarnos con una cantidad finita de elementos. Esto no supone una gran pérdida de información dado que $lim_{l\to\infty} f_l^m = 0$ (Análogo al Lema de Riemann-Lebesgue). Notemos que el doble sumatorio hace que si cortamos la descomposición en el valor $l = l_{max}$ incluido, la descomposición vendrá dada por $(l_{max} + 1)^2$ coeficientes.

Dado que las superficies de los granos son reales y no complejas, y que facilita la visualización y comprensión de los resultados, se ha elegido trabajar con la base de armónicos esféricos reales.

Hay que tener precaución al usar este método sobre otros datos. Este método proporciona una buena representación porque se considera la superficie del grano como una función esférica, y por lo tanto, se puede descomponer como combinación de armónicos. Esto quiere decir que la superficie a modelar tiene que ser una función esférica, cada par $(\phi, \theta)$ solo debe tener una imagen $r$ en coordenadas esféricas. Por lo tanto, no puede ser una función multivaluada.

La elección del parámetro $l_{max}$ es crucial, pues un valor pequeño dará lugar a una representación pobre del grano, mientras que un valor muy grande introducirá una rugosidad que no estaba en el grano original. Esto se ve claramente en la figura 10.

Esta rugosidad puede deberse a la representación finita de los reales en punto flotante, a la propagación de errores durante el cálculo de los coeficientes o a la existencia de

(a) Grano original

(b) $l_{max} = 5$

(c) $l_{max} = 15$

(d) $l_{max} = 30$

(e) $l_{max} = 50$

(f) $l_{max} = 70$

(g) $l_{max} = 85$

Fig. 10: Descomposiciones SH variando $l_{max}$

algunos puntos donde la función esférica sí es multivaluada.

En la sección 6 abordaremos el proceso seguido para la elección del parámetro $l_{max}$.

## 5.3. Reconstrucción de granos

La representación de los granos como una serie de armónicos esféricos permite expresar su superficie de forma abstracta sin la necesidad de puntos ni mallas. Pero al mismo tiempo, esto permite reconstruir un grano cualquiera modificando los vértices de una malla de referencia (o teselación) según los armónicos evaluados junto con sus coeficientes de la descomposición SH.

La primera teselación usada es el icosaedro, un sólido platónico con 12 vértices y 20 caras triangulares, que se ha usado como base para crear otras teselaciones. Para ello, se ha dividido de forma recursiva cada una de sus caras en cuatro triángulos y se han proyectado los nuevos vértices en la esfera que circunscribe al icosaedro.

Para diferenciar las distintas teselaciones generadas a partir del icosaedro, se han denominado según el nombre de veces que se ha dividido las caras de la malla original de forma recursiva, por lo que el icosaedro es la teselación T0, la siguiente T1 y así sucesivamente. A medida que aumenta el número de veces que dividimos las caras, tanto la resolución como el tamaño que ocupa en memoria aumentan. En la figura 11 se pueden ver las distintas teselaciones así como en la tabla 1 su resolución y su tamaño en memoria.

(a) T0 (b) T1 (c) T2

(d) T3 (e) T4 (f) T5

Fig. 11: Teselaciones derivadas del icosaedro

| Teselacion | n.º vertices | n.º caras | Espacio en memoria |
|---|---|---|---|
| T0 | 12 | 20 | 2 KB |
| T1 | 42 | 80 | 4 KB |
| T2 | 162 | 320 | 16 KB |
| T3 | 642 | 1280 | 63 KB |
| T4 | 2562 | 5120 | 251 KB |
| T5 | 10242 | 2048 | 1 MB |
| T6 | 40962 | 81920 | 4 MB |
| T7 | 163842 | 327680 | 16 MB |

Tabla 1: TAMAÑO DE LAS TESELACIONES

La ventaja que ofrece la reconstrucción de granos usando su descomposición SH es la posibilidad de operar entre ellos punto a punto. Si se reconstruye todo el dataset usando la misma teselación tendremos que todos los granos comparten la distribución de vértices en la esfera unidad. Esto quiere decir que si expresamos los vértices en coordenadas esféricas, todos los granos tendrán sus componentes $(\theta, \varphi)$ iguales, cambiando solo el radio.

Esto supone una herramienta muy potente puesto que permite sumar granos vértice a vértice, hacer el grano promedio y restar granos generando granos error. Una aplicación más práctica podría ser analizar las diferencias entre las razas de granos a partir de sus granos promedio.

El problema que tienen estas teselaciones es que tienen forma esférica, y al evaluar la serie de armónicos sobre los vértices la esfera se estira de dos puntas, provocando que la mayoría de vértices se acumulen en el centro del grano y las puntas se queden con menos información. Esto se ve claramente en la figura 12.

Fig. 12: Grano reconstruido sobre T5

Para resolverlo se ha optado por reconstruir todo el dataset con la teselación T5, promediar el dataset reconstruido para obtener una malla común, generar una nube de puntos uniformemente distribuidos sobre la malla promedio con Poisson Disk Sampling, y finalmente reconstruir la superficie usando Ball Pivoting. Poisson Disk Sampling deja escoger aproximadamente el número de vértices a generar, permitiendo reducir o aumentar la resolución de forma arbitraria. Los granos promedio se pueden ver en la figura 13.

(a) Grano promedio sobre T5

(b) Grano promedio sobre T5 resampleado por Poisson Disk Sampling y reconstruido mediante Ball Pivoting

Fig. 13: Granos promedio

En la figura 14 se ilustra el proceso a seguir para reconstruir un dataset con una teselación en forma de grano que tenga los vértices distribuidos uniformemente.

Fig. 14: Reconstrucción del dataset

## 6 Experimentos

El objetivo inicial de este trabajo es clasificar los granos según sus características, por lo que se han definido cuatro posibles clasificaciones: Rows, Growth, Landrace y Bere. Las tres primeras clasificaciones ya se introdujeron en la sección 4, mientras que la última consiste en la distinción de la raza de grano Bere respecto al resto. Observemos que excepto Landrace, el resto son clasificaciones binarias. Con el desarrollo de este proyecto, se ha definido un nuevo experimento con el objetivo de encontrar el valor $l_{max}$ óptimo para generar una nueva representación de los datos.

### 6.1. Determinación del valor de truncamiento $l_{max}$ óptimo

Para encontrar el mejor valor $l_{max}$ para un grano se ha procedido de la siguiente forma. En primer lugar, se extrae la nube de puntos del grano y se divide aleatoriamente en dos conjuntos de train y test con una proporción 70:30. Se calcula la descomposición en armónicos esféricos del conjunto train hasta el valor $l = 85$, el máximo permitido por Scipy, y para cada valor de $l$ se considera la expansión hasta ese valor. Esto produce 86 descomposiciones distintas que podemos evaluar sobre el conjunto de test, y ver qué error se comete al aproximar estos puntos. Finalmente, se calcula el MSE entre el conjunto test original y los reconstruidos. En las figuras 16 y 17 se observa la evolución de esta medida al añadir más coeficientes a la descomposición.

### 6.2. Clasificación de granos de cebada

Para cada experimento de clasificación se ha hecho una partición del dataset en cinco folds, tres para entrenar, uno para validar y el último para testear. Todos los folds conservan la proporción de clases, por lo que se han descartado algunos granos para preservar esta proporción uniforme. La figura 15 muestra el tamaño de las particiones.

Las clasificaciones que abordo con este trabajo se han centrado en RF y MLP aplicados a los coeficientes SH. El motivo por el cual solo he considerado RF y no otros modelos clásicos en Machine Learning, fue que otros compañeros del proyecto de investigación obtuvieron sus mejores resultados con RF usando un vector de descriptores físicos del grano. De esta forma, la comparación entre los experimentos también mide la mejora del cambio de representación de los datos. Un compañero también usó

Fig. 15: Partición del dataset *(fuente: ICAC)*

el modelo MeshCNN, pero el tamaño de los granos en memoria era tan grande que se tuvo que reconstruir el dataset a resoluciones más pequeñas siguiendo la metodología descrita en la sección 5.3. No describo explícitamente el trabajo de estos compañeros puesto que en breves se publicará en una revista el artículo conjunto.

Para la clasificación base contra la que comparar el resto de métodos, se ha usado un vector de descriptores físicos de 354 medidas. Entre ellas se encuentran: las dimensiones de la bounding box del grano alineado, volumen de la convex hull, area de la convex hull, aspect ratios entre longitud, ancho y alto, compacticidad, excentricidad, direcciones de los vectores principales obtenidos mediante PCA y descriptores de Fourier para las intersecciones del grano con planos.

Tanto para RF como para MLP, se han entrenado distintos modelos variando sus hiperparámetros y guardando el modelo que obtuvo el mejor accuracy sobre los datos de validación. Finalmente, una vez fijados los hiperparámetros del modelo, se evalúa el modelo con los datos de test. A continuación se muestran los distintos hiperparámetros considerados así como los valores probados:

- RF
  - n.º de árboles: [100, 200, 300, 400, 500]
  - n.º máximo de características: [sqrt, log2, 0.5]
  - Profundidad máxima: [Ninguna, 10, 20, 30]
  - n.º mínimo para dividir un nodo: [2, 5, 10]
  - n.º mínimo para formar hoja: [1, 2, 3, 4, 5]
- MLP
  - Arquitectura de capas: [(50,), (100,), (100, 50), (100, 100), (100, 50, 25), (200, 100, 50)]
  - Regularización L2 ($\alpha$): [0.0001, 0.001, 0.01, 0.1]
  - Tasa de aprendizaje inicial: [0.001, 0.01, 0.1]

## 7 Resultados y discusión

En esta sección se presentan los resultados de la determinación del mejor parámetro $l_{max}$ y de los experimentos de clasificación de granos.

En la figura 16 se aprecia la evolución del MSE para distintos granos en función de $l_{max}$.

Fig. 16: MSE según $l_{max}$

Vemos que para todos los granos hay un comportamiento similar, un error alto que decrece en una larga planicie y que vuelve a aumentar con descomposiciones muy altas. En la figura 17 se aprecia la evolución del MSE promediado de los granos de la figura 16 en función de $l_{max}$.

Fig. 17: MSE promedio de varios granos según $l_{max}$

El valor óptimo se encuentra entre 50 y 60. Se ha escogido 50 por ser un número redondo y porque cuanto más pequeño $l_{max}$, menos computo hará falta. Si escogiéramos el valor $l_{max} = 55$, los arrays tendrían 3136 elementos, que son 535 más que si escogiéramos $l_{max} = 50$.

A continuación, se muestran los resultados obtenidos en los diferentes experimentos. Con el propósito de generar una comparativa de resultados más rica, incluiré los resultados de mis compañeros con los modelos RF+descriptores y MeshCNN.

Los resultados obtenidos se muestran en tablas, una para cada modelo considerado. Cada fila corresponde a un experimento, mientras que las columnas indican los accuracies obtenidos. Acc. 1 y Acc. 2 hacen referencia a los accuracies obtenidos en las clasificaciones binarias, mientras que Acc. 1, ..., Acc. 4 son los accuracies obtenidos en la clasificación multiclase de razas. Acc. General es el accuracy general del experimento.

En el experimento Rows la clase 1 corresponde a 2-row y la clase 2 a 6-row. En el experimento Growth la clase 1 corresponde a Dundee y la clase 2 a Orkney. En el experimento Bere, la clase 1 corresponde a Bere y la clase 2 a No-Bere. En el experimento Landrace, la clase 1 corresponde a Bere, la clase 2 a British, la clase 3 a Scandinavian y la clase 4 a Scottish.

En la tabla 2 se muestran los accuracies obtenidos con los modelos RF usando la representación SH y en la tabla 3 los hiperparámetros de cada modelo.

| Experimento | Acc. General | Acc. 1 | Acc. 2 | Acc. 3 | Acc. 4 |
|---|---|---|---|---|---|
| Rows | 87,50 % | 86,76 % | 88,23 % | | |
| Growth | 87,31 % | 85,07 % | 89,56 % | | |
| Bere | 87,14 % | 91,43 % | 82,86 % | | |
| Landrace | 77,00 % | 80,00 % | 80,00 % | 80,00 % | 60,00 % |

Tabla 2: ACCURACIES OBTENIDOS CON RF+SH

| Experimento | n.º árboles | max. carac. | profundidad | min. división | min. hoja |
|---|---|---|---|---|---|
| Rows | 200 | sqrt | 10 | 10 | 1 |
| Growth | 100 | 0.5 | None | 2 | 1 |
| Bere | 500 | sqrt | None | 10 | 1 |
| Landrace | 200 | sqrt | None | 10 | 3 |

Tabla 3: HIPERPARÁMETROS DE RF+SH

En la tabla 4 se muestran los accuracies obtenidos con los modelos MLP usando la representación SH y en la tabla 5 los hiperparámetros de cada modelo.

| Experimento | Acc. General | Acc. 1 | Acc. 2 | Acc. 3 | Acc. 4 |
|---|---|---|---|---|---|
| Rows | 97,06 % | 95,59 % | 98,53 % | | |
| Growth | 90,30 % | 92,54 % | 88,06 % | | |
| Bere | 95,71 % | 97,14 % | 94,28 % | | |
| Landrace | 80,00 % | 92,00 % | 56,00 % | 96,00 % | 76,00 % |

Tabla 4: ACCURACIES OBTENIDOS CON MLP+SH

| Experimento | Arquitectura | Regularización L2 | Tasa de aprendizaje |
|---|---|---|---|
| Rows | (200,100,50) | 0.001 | 0.01 |
| Growth | (100,100) | 0.001 | 0.001 |
| Bere | (200,100,50) | 0.01 | 0.001 |
| Landrace | (100,50) | 0.01 | 0.01 |

Tabla 5: HIPERPARÁMETROS DE MLP+SH

Observamos que los accuracies obtenidos con los modelos MLP son superiores respecto RF para todos los experimentos. Aunque la clasificación multiclase con MLP presente un accuracy mejor, las clases British y Scottish se han confundido bastante entre ellas, algo que no ocurre con el modelo RF.

En la tabla 6 se muestran los accuracies obtenidos del modelo RF con un vector de descriptores como input. Solo se evaluó para el experimento Landrace y en su entrenamiento se utilizó la técnica Leave One Out Cross Validation. El número de muestras no es uniforme para cada clase. Además, incorpora también una quinta clase que posteriormente hemos descartado en los experimentos por las pocas muestras de las que se dispone.

| Experimento | Acc. General | Acc. 1 | Acc. 2 | Acc. 3 | Acc. 4 | Acc. 5 |
|---|---|---|---|---|---|---|
| Landrace | 64,95 % | 80,00 % | 73,57 % | 73,57 % | 15,71 % | 0,00 % |

Tabla 6: ACCURACIES OBTENIDOS CON RF Y VECTOR DE DESCRIPTORES

Los resultados del modelo RF usando la representación SH superan con creces la representación de vector de descriptores. Para hacer una comparativa más sólida, se deberían replicar las condiciones de entrenamiento y validación, además de probar otras clasificaciones.

En la tabla 7 se muestran los accuracies obtenidos con el modelo MeshCNN con la resolución de los granos reconstruidos al 10 %.

| Experimento | Acc. General | Acc. 1 | Acc. 2 | Acc. 3 | Acc. 4 |
|---|---|---|---|---|---|
| Rows | 83,82 % | 85,29 % | 82,35 % | | |
| Growth | 79,85 % | 76,12 % | 83,58 % | | |
| Bere | 78,57 % | 77,14 % | 80,00 % | | |
| Landrace | 64,00 % | 68,00 % | 52,00 % | 84,00 % | 52,00 % |

Tabla 7: ACCURACIES OBTENIDOS CON MESHCNN

Observamos que con MeshCNN también se producen dificultades para distinguir las clases British y Scottish.

## 8 CONCLUSIONES

En primer lugar, comparando los resultados de los modelos RF, podemos concluir que la representación en armónicos esféricos es mucho más rica que un vector de descriptores de características físicas medibles.

El modelo MLP+SH resulta ser sustancialmente mejor que RF+SH y MeshCNN. Cabe destacar que MeshCNN ha trabajado sobre granos que han sido reconstruidos con armónicos esféricos sobre una teselación de resolución reducida, un proceso que ha comportado pérdida de información e introducción de error.

Podemos deducir que dado un grano, es posible distinguir el lugar de crecimiento, la raza y el número de carreras con alta precisión analizando sólo la superficie. También concluimos que las razas British y Scottish se parecen bastante ya que la mayoría de modelos presentan dificultades para distinguir estas dos clases.

Para futuros trabajos quedan pendientes algunos temas que procedo a comentar. Para poder hacer una comparación sólida contra MeshCNN se debería encontrar la forma de entrenar con el dataset original, pero por ahora la limitación de memoria no lo permite. Probar diferentes formas de reducir el error introducido al realizar las descomposiciones SH. La arquitectura del modelo MLP es bastante sencilla y ha demostrado muy buenos resultados, quizás probar arquitecturas más avanzadas producirá mejores resultados.

## 9 Agradecimientos

En primer lugar, me gustaría agradecer a mis amigos y compañeros informáticos, en especial a Itziar y a Arnau, por acompañarme durante el desarrollo de todo el proyecto, el soporte emocional y hacerlo todo más llevadero.

Agradecer a Pablo, mi novio, que siempre ha estado a mi lado cuando lo he necesitado.

Agradecer a Felipe, mi profesor, jefe y tutor, por guiarme, animarme, resolver mis dudas y no dejar que me rinda. Sin él, nada de esto habría sido posible.

Finalmente, agradecer al ICAC que me haya permitido ser partícipe de un proyecto que he disfrutado enormemente y del que he aprendido muchísimo.

## Referencias

[1] M. Wallace, V. Bonhomme, J. Russell, E. Stillman, T. George, L. Ramsay, J. Wishart, S. Timpany, H. Bull, A. Booth, and P. Martin, "Searching for the origins of bere barley: a geometric morphometric approach to cereal landrace recognition in archaeology," *Journal of Archaeological Method and Theory*, vol. 26, 09 2019.

[2] R. M. Godinho, C. Umbelino, and C. Gonçalves, "Mesolithic and chalcolithic mandibular morphology: Using geometric morphometrics to reconstruct incomplete specimens and analyse morphology," *Open Archaeology*, vol. 8, no. 1, pp. 536–549, 2022.

[3] A. G. Drake, M. Coquerelle, and G. Colombeau, "3d morphometric analysis of fossil canid skulls contradicts the suggested domestication of dogs during the late paleolithic," *Scientific reports*, vol. 5, no. 1, p. 8299, 2015.

[4] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or, "Meshcnn: a network with an edge," *ACM Transactions on Graphics*, vol. 38, p. 1–12, July 2019.

[5] A. Mukhaimar, R. Tennakoon, C. Y. Lai, R. Hoseinnezhad, and A. Bab-Hadiashar, "Robust object classification approach using spherical harmonics," 2020.

[6] D. Eschweiler, M. Rethwisch, S. Koppers, and J. Stegmaier, "Spherical harmonics for shape-constrained 3d cell segmentation," 2020.

[7] G. Biau and E. Scornet, "A random forest guided tour," *TEST*, vol. 25, 11 2015.

[8] M.-C. Popescu, V. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Transactions on Circuits and Systems*, vol. 8, 07 2009.

[9] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "Meshlab: an open-source mesh processing tool.," vol. 1, pp. 129–136, 01 2008.

[10] M. Corsini, P. Cignoni, and R. Scopigno, "Efficient and flexible sampling with blue noise properties of triangular meshes," *IEEE transactions on visualization and computer graphics*, vol. 18, pp. 914–24, 01 2012.

[11] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 5, pp. 349 – 359, 11 1999.

[12] D.-H. et al., "trimesh." `https://github.com/mikedh/trimesh`, 2019.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[14] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.