
Clustering y segmentación de imágenes de cinco variedades de arroz mediante algoritmos no supervisados

Víctor A. Bettachini
bettachini@gmail.com

Vanesa Flores
vanesaflores0894@gmail.com

Tomás Gianni
tomasgianni11@gmail.com

Malena Pirola
malenapirola@gmail.com

Resumen

La clasificación en variedades y evaluación de calidad de los granos de arroz de forma tradicional insume gran cantidad de tiempo y recursos, los cuales podrían ser optimizados mediante el uso de técnicas de análisis de imagen y aprendizaje automático. En este trabajo se evaluaron soluciones de agrupamiento no supervisado de imágenes color de granos de arroz de cinco variedades distintas mediante los algoritmos K-means, DBSCAN y Partición Alrededor de Medoides (PAM). Las variables sobre la que se realizó el agrupamiento fueron atributos generados en una capa intermedia de la red convolucional entrenada para la clasificación de imágenes (VGG16). El desempeño de los algoritmos de agrupamiento se midió mediante métricas de validación interna y externa. En estos términos, K-means y PAM presentaron un desempeño muy similar y muy superior al de DBSCAN. Asimismo, para ensayar la identificación de granos individuales en imágenes con muchos de estos, se realizó una segmentación con algoritmos de etiquetado de componentes conectados y de agrupamiento espectral. Para el caso de imágenes de granos de arroz bien separados, ambos algoritmos dieron resultados satisfactorios, aunque el agrupamiento espectral arrojó mejores resultados para segmentar imágenes con múltiples granos de arroz superpuestos.

1. Introducción

El arroz, uno de los granos de mayor producción en todo el mundo, tiene muchas variedades genéticas que se diferencian entre sí por atributos como la textura, la forma y el color. Estos atributos permiten clasificar y evaluar la calidad de las semillas, lo cual afecta el precio final del producto. Tradicionalmente, esta tarea requiere de la participación de operadores humanos expertos, con un elevado costos y tiempo de procesamiento. La implementación de algoritmos de aprendizaje automático sobre imágenes de las semillas de arroz puede permitir una agilización del proceso de clasificación y evaluación de la calidad y un uso más eficiente de los recursos.

Se han publicado interesantes resultados en la tareas de clasificación mediante algoritmos supervisados de este tipo de imágenes. Algunos algoritmos clásicos, e.g. Random Forest o KNN, operando con atributos morfológicos y de color alcanzaron tasas de éxito de clasificación superiores al 98 % [1], en tanto que con redes convolucionales profundas se logró una tasa de éxito del 100 % entrenando la red directamente con las imágenes [2].

Utilizando las mismas imágenes que las empleadas en éste último estudio [2], en este trabajo se busca evaluar la capacidad de agrupamiento de imágenes de semillas mediante algoritmos no supervisados a

partir de atributos extraídos de una red convolucional pre-entrenada para la clasificación de imágenes [3]. Asimismo, se busca explorar las posibilidades que ofrecen diferentes algoritmos de segmentación para la identificación de granos de arroz individuales en imágenes con muchos de estos.

Es razonable suponer que si provee capacidad de caracterización de los granos a partir de sus imágenes será por su forma y color siendo insensible a la textura. A pesar de esta limitación, que no hemos cuantificado, se emprendió la tarea de agrupar atributos obtenidos mediante la aplicación de la red convolucional utilizando distintos algoritmos de agrupación: K-Means, DBSCAN y PAM (partición alrededor de medoides).

2. Materiales y métodos

Datos Se obtuvieron de una fuente pública [1] unas 75K imágenes de granos de arroz balanceadas en 5 distintas variedades (15K imágenes por cada tipo): Arborio, Basmati, Ipsala, Jazmín y Karacadag, que corresponden a variedades que suelen cultivarse en Turquía. Las imágenes provistas en formato jpeg presentan tres capas de color RGB (rojo, verde y azul) con un tamaño de 250 x 250 píxeles. Cada píxel tiene un valor de 0 a 255 que codifica la intensidad lumínica del color. En cada imagen se visualiza un único grano de arroz que está centrado en la imagen sobre un fondo negro, como se observa en los ejemplos (Figura 1). El procesamiento y análisis de datos se realizó sobre una muestra aleatoria estratificada sin reemplazo de 5K imágenes, i.e. 1K por cada variedad de arroz.

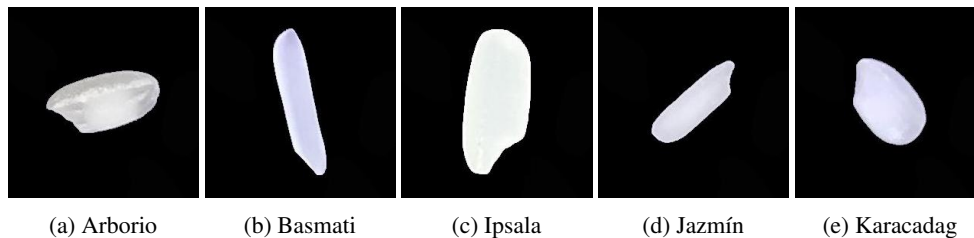


Figura 1: Ejemplo de imagen del conjunto de datos para cada variedad de arroz representada [a) - e]).

Extracción de atributos Para lograr una reducción de la dimensionalidad de los datos de cada imagen se empleó un modelo provisto por la biblioteca Keras bajo el nombre VGG16 [4]. Este es producto de entrenar una red convolucional de 16 capas en forma supervisada para la clasificación de imágenes [3]. Neuronas de sus capas intermedias responden a atributos abstractos útiles para realizar tareas distintas a la clasificación y/o trabajar sobre imágenes disímiles a las usadas en el entrenamiento (*transfer learning*). Así se caracterizan distintos aspectos abstraídos de la imagen que pueden ser utilizados para entrenar algoritmos de clasificación o agrupamiento más específicos a cierto dominio (*fine-tuning*).

Cada imagen se redujo a 224 x 224 píxeles que se ordenaron en un vector único de tamaño 150.528 (224 x 224 x 3). Tras alimentar cada una de ellas se registró la salida de la penúltima capa que consta de 4.096 neuronas, obteniéndose una reducción al $\approx 2,7\%$ de su dimensionalidad original.

Clustering Los 4.096 atributos de cada imagen, previamente normalizados entre 0 y 1 a través del método MinMax, se alimentaron a tres distintos algoritmos no supervisados de agrupación (*clustering*): K-means, DBSCAN y PAM (o K-medoides). Se evaluaron los resultados con medidas de validez interna y externa.

El algoritmo de K-means opera en una serie de pasos.

1. Selecciona un número predefinido de clusters (K) y elige K puntos iniciales como centros de los clusters de manera aleatoria.
2. Asigna cada punto de datos al cluster cuyo centro está más cercano, generalmente utilizando la distancia euclidiana.
3. Calcula el nuevo centro de cada cluster como el promedio de los puntos asignados a ese cluster.

4. Repite los pasos 2 y 3 hasta que los centros de los clusters converjan o hasta que se alcance un número máximo de iteraciones predefinido.

El resultado final del algoritmo es un conjunto de K clusters, donde los puntos de datos dentro de cada cluster son similares entre sí y diferentes de los puntos en otros clusters.

El algoritmo DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) es un método de agrupamiento que, a diferencia de K-means, no requiere que se especifique previamente el número de clusters (k). En su lugar, DBSCAN identifica clusters basándose en la densidad de puntos en el espacio de datos. Detecta automáticamente la cantidad de clusters y es especialmente efectivo para encontrar clusters de formas arbitrarias y separados por áreas de baja densidad. DBSCAN también puede manejar ruido de manera más efectiva al clasificar puntos ruidosos como ruido en lugar de asignarlos a un cluster, lo que lo diferencia de K-means en términos de robustez en datos reales y la capacidad de adaptarse a la estructura de los datos sin requerir suposiciones iniciales sobre el número de clusters.

El algoritmo PAM (*Partitioning Around Medoids*), también conocido como K-medoids, es similar a K-means, pero se diferencia de éste en su enfoque en la elección de los puntos representativos del cluster. En lugar de usar la media de los puntos como centro del cluster (como en K-means), PAM selecciona medoides, que son puntos reales del conjunto de datos. Estos medoides son los puntos que minimizan la distancia total a todos los demás puntos dentro del cluster. Esta estrategia hace que PAM sea más robusto frente a valores atípicos o ruidosos en comparación con K-means y permite una interpretación más clara de los puntos representativos de cada cluster.

Las medidas de validez interna empleadas para evaluar los algoritmos fueron el SSE y el coeficiente de Silhouette. El primero es la suma de los cuadrados de las distancias de cada observación al centroide del cluster (más bajo cuanto más compactos y coherentes sean los clusters). El segundo cuantifica entre -1 y 1 cuán similar es cada punto de datos a su propio cluster comparándolo con otros clusters cercanos. Cuánto menor sea el coeficiente de Silhouette, más grande es la diferencia entre el punto y otros miembros del cluster, indicando que el punto está "mal agrupado", mientras que valores en torno a 0 sugieren un gran solapamiento entre los grupos. Estas dos medidas de validez interna también son útiles para determinar la cantidad óptima de clusters. Para esto se aplicaron iterativamente para los casos de 1 a n clusters y luego se compararon los resultados medidos por SSE y Silhouette.

Por otro lado, la validación externa permite comparar diferentes soluciones de agrupamiento entre sí. Se calculó una la matriz de confusión agrupando variedades de arroz según la etiqueta en el conjunto de datos vs. al cluster asignado por los algoritmos. Esto permitió evaluar cuán bien mapearon los clusters las variedades y la utilidad de dichos algoritmos para su separación. Asimismo, se calculó el índice de Rand ajustado, que puede tomar valores entre 0 y 1 donde 1 indica una concordancia perfecta entre las asignaciones de clusters y 0 indica una asignación aleatoria. Además, se calculó el índice de van Dongen, que cuantifica la disimilaridad entre clusters basada también en la intersección máxima de clusters y toma valor 0 cuando los clusters son iguales entre ambas soluciones de agrupamiento.

Segmentación Para realizar la segmentación y detección de objetos individuales se sintetizó una imagen nueva a partir de cuatro imágenes de granos de arroz de diferente variedad. La imagen resultante fue de 500 x 500 píxeles con 3 canales de color, la cual fue pasada a blanco y negro utilizando la función umbral de la biblioteca OpenCV para Python con un valor de 0 (i.e. todos los píxeles con valor superior a 0 fueron llevados a 255). Como evaluación adicional, se aplicaron los algoritmos de segmentación sobre una imagen adicional obtenida de la web con muchos granos de arroz.

Se aplicaron dos algoritmos de segmentación: a) etiquetado de componentes conectados (CCL, por sus siglas en inglés) utilizando la librería OpenCV, y b) clustering espectral mediante la librería Scikit-learn (`sklearn.cluster`) para Python. CCL es un algoritmo iterativo que parte de examinar un píxel, que oficia de "semilla" y luego examina sus píxeles vecinos. Si éstos tienen características similares al píxel semilla, se lo considera parte del mismo componente, de lo contrario se lo considera un componente diferente. De esta manera se recorren todos los píxeles de la imagen hasta que todos se asignan a algún componente. Por otro lado, el clustering por análisis espectral parte de la generación de una matriz de similitud entre los puntos que sirve de base para crear un grafo, en donde cada punto es un nodo y las similitudes son las aristas que los conectan, con menor o mayor "fuerza" según el grado de similitud entre los puntos. Mediante la descomposición en autovectores y autovalores de

las matrices asociadas con el grafo, el algoritmo es capaz de agrupar puntos que están fuertemente conectados y separarlos de otros grupos.

3. Resultados

Clustering por K-means Se analizaron los gráficos de las medidas de validez interna para los clusters en función del número de los mismos.

Para el caso del SSE puede notarse en la figura 2a que disminuye en forma suave, sin presentar un quiebre que indique de forma obvia la cantidad de clusters óptimo. Esto puede deberse a que los atributos no poseen una clara estructura de agrupamiento.

En el caso del coeficiente de Silhouette la figura 2b muestra que este resultó cercano a 0 para todos los números de clusters seleccionados (k). Por tanto calcular este coeficiente no proveyó una guía clara de que número sería el adecuado.

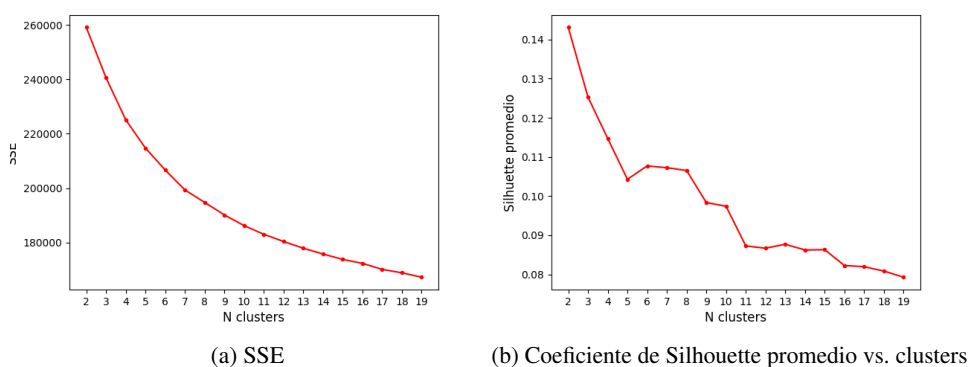


Figura 2: K-means: métricas de validación interna.

Considerando el objetivo del presente trabajo, se decidió proceder con $k = 5$ clusters para evaluar la capacidad del algoritmo de agrupar las imágenes de forma comparable con las variedades de arroz originales, así como $k = 7$ clusters para fines comparativos.

Las medidas de validación externa que figuran en la tabla 1 sugieren que no existen prácticamente diferencias en el desempeño en los algoritmos para cinco o siete clusters. En la misma tabla figuran también los valores numéricos de las validaciones internas. Para ambos números de clusters, el índice Rand ajustado sugiere que si bien los modelos no se ajustan perfectamente a la categorización por variedades conocida, la agrupación modelada es superior a la que se obtendría por azar. Asimismo, el índice van Dongen cercano a 0 indica que para ambos modelos la solución de agrupamiento alcanzada es similar a la que resulta del agrupamiento por las etiquetas conocidas de variedad de arroz.

Clusters (k)	SSE	Silhouette promedio	Rand ajustado	van Dongen
5	214744.0	0.1040	0.4230	0.1880
7	199367.0	0.1070	0.4768	0.1520

Tabla 1: K-means: métricas de validez interna y externa (vs. etiquetas de clase conocidas).

La matriz de confusión del modelo de cinco clusters que presenta la tabla 2 evidencia que al menos el tipo de arroz Karacadag quedó agrupado casi en su totalidad un sólo cluster, el numerado con 0. Por el contrario, los clusters 1 y 2 están compuestas por similares proporciones de Arborio e Ipsala, patrón que se repite en los clusters 3 y 4 con Jazmín y Basmati. En la figura 3 se puede observar que estas agrupaciones responden a patrones de morfología de los granos que hacen difícil su separabilidad.

Como medio de facilitar la visualización de los clusters generados por K-means ($k = 5$), se realizó un análisis de componentes principales para reducir la dimensionalidad del espacio de atributos. La

Cluster Variedad	Arborio	Ipsala	Jazmín	Karacadag	Basmati
0	81	4	1	921	0
1	10	7	525	0	437
2	456	468	40	22	0
3	446	520	42	57	1
4	7	1	392	0	562

Tabla 2: K-means: matriz de confusión para 5 clusters vs. etiquetas de variedad.

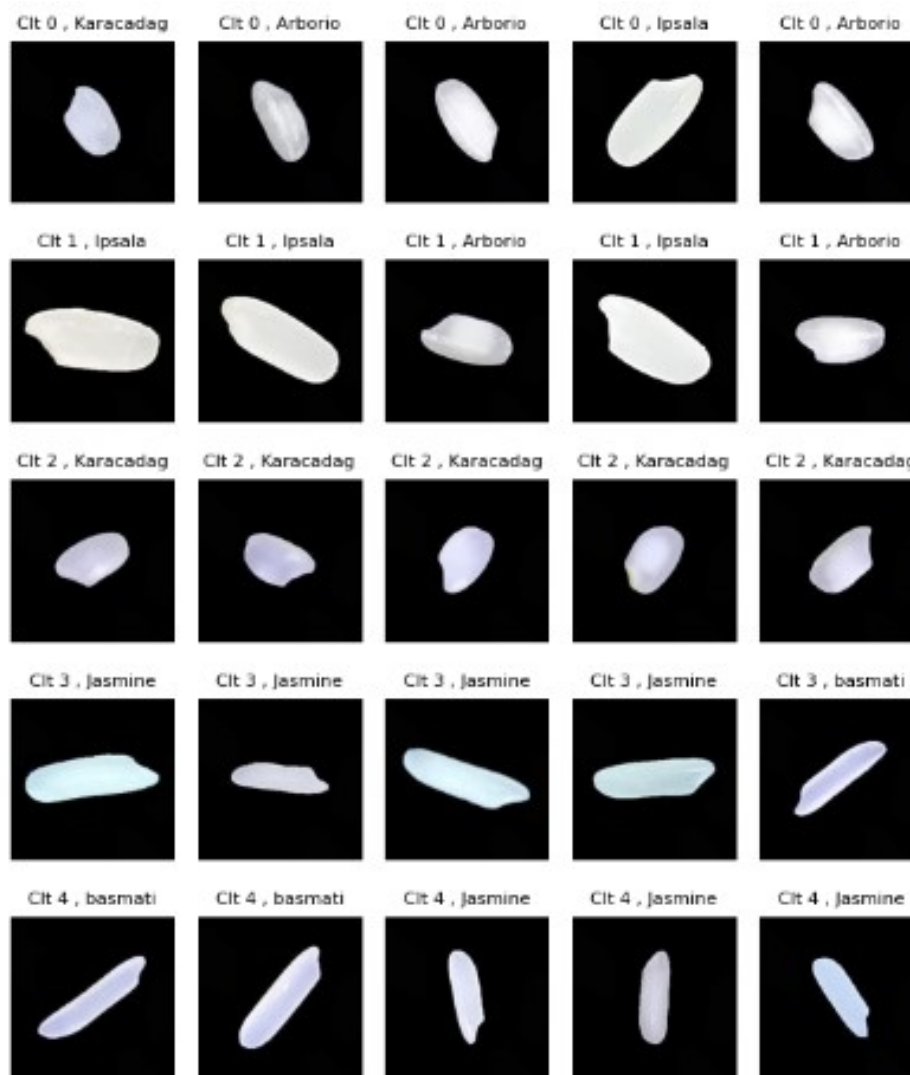


Figura 3: K-means: ejemplos de cada uno de los cinco clusters con su etiqueta original.

figura 4 muestra los cinco clusters graficados sobre los componentes 1 y 2 obtenidos mediante PCA (+40 % de variabilidad explicada), en donde se corrobora cierta superposición entre los grupos y por ende, baja separabilidad.

El caso con siete clusters, que resume la tabla 3, volvió a verificarse la eficiente distinción de la variedad Karacadag donde el cluster 2 se compone casi exclusivamente de esta variedad, pero el resto de los clusters presenta características mixtas.

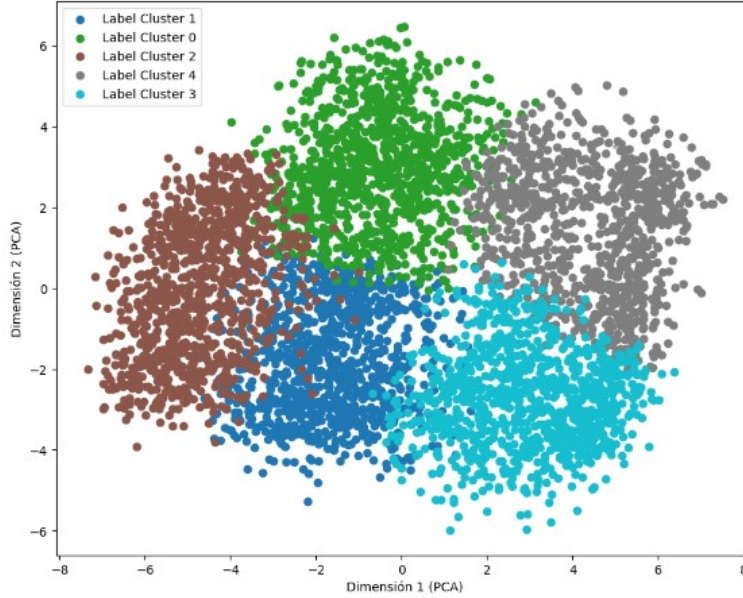


Figura 4: K-means: visualización de los 5 clusters generados. Componentes 1 y 2 obtenidos por PCA sobre atributos originales utilizados para el entrenamiento).

Cluster Variedad	Arborio	Ipsala	Jazmín	Karacadag	Basmati
0	13	2	375	0	230
1	41	503	5	4	0
2	59	3	1	866	0
3	2	0	33	0	451
4	645	44	13	128	1
5	11	8	537	0	318
6	229	440	36	2	0

Tabla 3: K-means: matriz de confusión para 7 clusters vs. etiquetas de variedad.

Clustering por DBSCAN Para este algoritmo se optimizaron dos parámetros, *eps*, el radio del entorno usado para definir los clusters, y *MinPts*, la cantidad mínima de puntos en el entorno.

En una primer instancia, se tomó un criterio sugerido en bibliografía consultada para la optimización de los parámetros: considerar el 4to vecino más cercano para realizar los agrupamientos; con dicha premisa, que equivale a establecer *MinPts* = 4 en el modelo, se corrió el algoritmo de vecinos más cercanos (*KNN*) para $n = 4$, y se buscó un quiebre en la curva representada en la figura 5a, que marca un valor de *eps* óptimo de 7, 25.

Al ajustar nuestro modelo con estos parámetros se identificó un único gran cluster y puntos ruidosos, lo cual no es satisfactorio; por ello, decidimos evaluar un rango de valores para ambos parámetros y evaluar el coeficiente de Silhouette obtenido con cada una de dichas combinaciones. Éstos resultaron muy cercanos a cero para cualquier par de parámetros *eps* y *MinPts* evaluados, como muestra la figura 5b, lo cual es indicativo de un “mal” agrupamiento. Sin embargo, considerando el objetivo del presente trabajo, se decidió continuar con la combinación de parámetros que encontró cinco clusters (*eps* = 6, *MinPts* = 3) para evaluar la capacidad del algoritmo de agrupar las imágenes de forma comparable con las variedades de arroz originales.

Las medidas de validación externa resumidas en la tabla 4 sugieren, del mismo modo, un agrupamiento poco efectivo: tanto el índice Rand ajustado como el índice van Dongen indican una mala coincidencia entre lo obtenido por el modelo y las etiquetas reales de los datos.

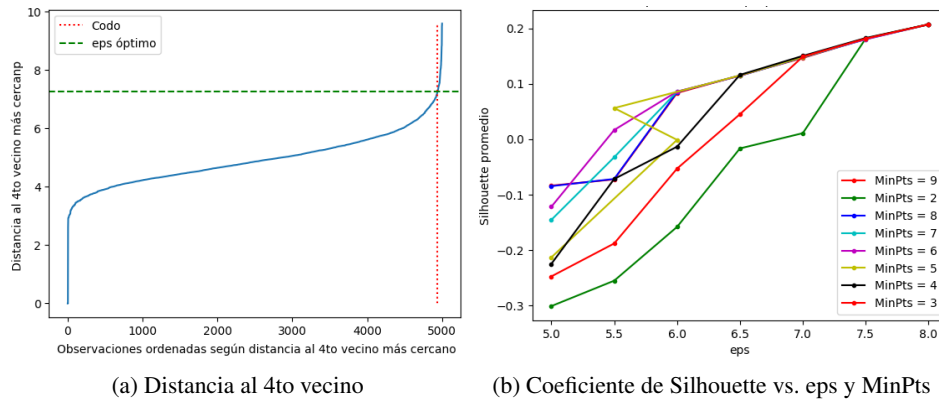


Figura 5: DBSCAN: optimización de parámetros y validación interna.

Parámetros	Clusters	Silhouette promedio	Rand ajustado	van Dongen
eps = 8, MinPts = 9	1	0.201	0	0.8
eps = 86, MinPts = 3	5	-0.05	0.006	0.67

Tabla 4: DBSCAN: métricas de validez interna y externa (vs. etiquetas de clase conocidas).

La matriz de confusión del modelo para cinco clusters 5 refuerza estas conclusiones: se aprecia que el cluster 0 abarca la gran mayoría de los puntos, y que evidentemente los 4 clusters restantes han sido completamente forzados; el cluster -1 corresponde a los puntos que el modelo catalogó como ruido", es decir, sin agrupamiento claro. En la figura 6 se puede observar las agrupaciones que realizó el modelo: solo para el cluster 0 detectó al menos 10 elementos, lo cual es suficiente prueba de su baja eficiencia debido a la poca separabilidad de los datos.

La figura 7 muestra los cinco clusters graficados sobre los componentes 1 y 2 obtenidos mediante PCA (+40 % de variabilidad explicada), en donde se corrobora una total superposición entre los grupos y la nula separabilidad que advirtieron los resultados.

Clustering por PAM Finalmente, probaremos el rendimiento del método de clustering *PAM* (*Partition Around Medoids*), o también conocido como *K-medoids* para evaluar sus diferencias de desempeño respecto de los otros algoritmos empleados.

Al igual que lo realizado para K-means, se analizaron los gráficos de las medidas de validez interna para los clusters en función del número de los mismos:

Similar a lo que ocurre con K-means, de la figura 8a podemos notar que el valor de SSE disminuye mientras aumenta la cantidad de clusters, aunque de forma menos suave que en el primero modelo. El quiebre de la curva alrededor de los 5 o 6 clusters permite suponer que en ese entorno se encontrará el valor óptimo.

Cluster Variedad	Arborio	Ipsala	Jazmín	Karacadag	Basmati
-1	120	238	106	50	52
0	880	746	894	950	948
1	0	6	0	0	0
2	0	3	0	0	0
3	0	4	0	0	0
4	0	3	0	0	0

Tabla 5: DBSCAN: matriz de confusión para 5 clusters vs. etiquetas de variedad.

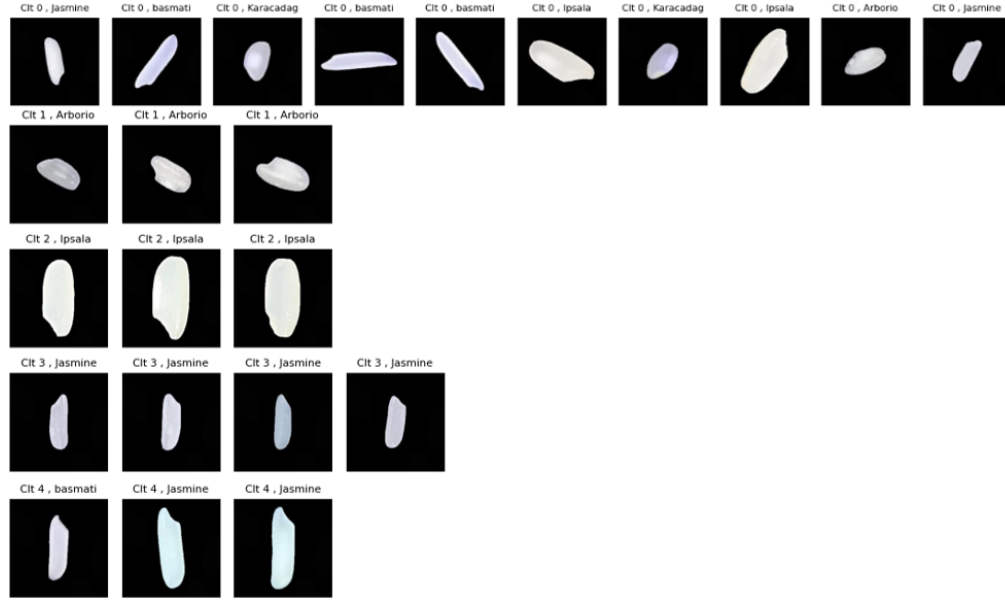


Figura 6: DBSCANs: muestra de los clusters generados

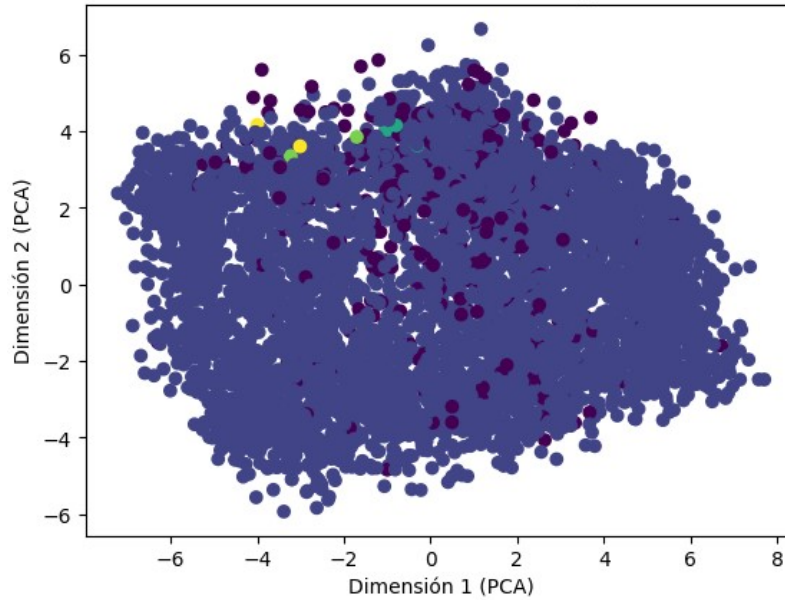


Figura 7: DBSCANs: visualización de los 5 clusters generados. Componentes 1 y 2 obtenidos por PCA sobre atributos originales utilizados para el entrenamiento). Hay una nula separación entre clusters.

En el caso del coeficiente promedio de Silhouette, la figura 8b también muestra un comportamiento similar a K-means: el coeficiente resultó cercano a 0 para todos los números de clusters seleccionados (k). Por tanto, calcular este coeficiente no proveyó una guía clara de qué número sería el adecuado.

A fin de realizar una comparación fructífera entre distintos algoritmos de agrupamiento, se decidió proceder con los mismos criterios que los empleados al realizar k-means: evaluar el modelo con $k = 5$, emulando las variedades de arroz originales, así como con $k = 7$ clusters para fines comparativos.

Las medidas de validación externa que figuran en la tabla 6 sugieren que no existen diferencias significativas en el desempeño en los algoritmos para cinco o siete clusters, similar a lo ocurrido con

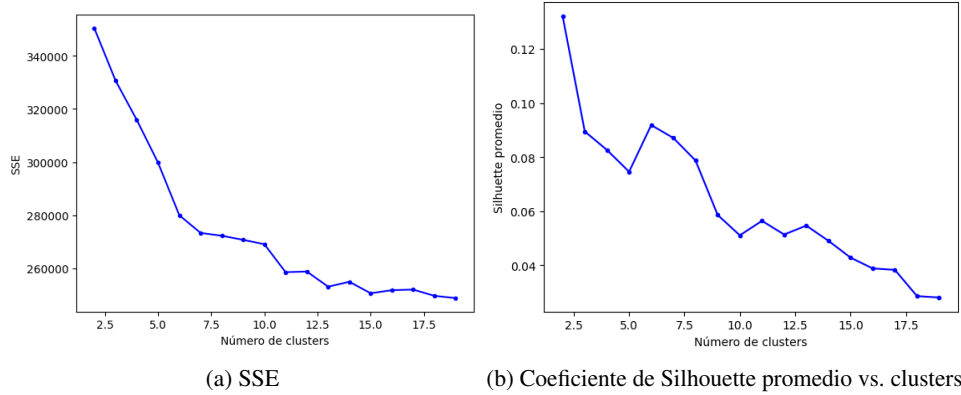


Figura 8: PAM: métricas de validación interna.

K-means. En dicha tabla figuran también los valores numéricos de las validaciones internas. Para ambos números de clusters, el índice Rand ajustado sugiere que si bien los modelos no se ajustan perfectamente a la categorización por variedades conocida, la agrupación modelada es superior a la que se obtendría por azar. Cabe destacar que para cinco clusters éste índice fue ligeramente superior (beneficioso) que el obtenido al realizar K-means, mientras que se dio la situación inversa para 7 clusters. Asimismo, el índice van Dongen cercano a 0 indica que para ambos modelos la solución de agrupamiento alcanzada es similar a la que resulta del agrupamiento por las etiquetas conocidas de variedad de arroz; al comparar con lo obtenido con K-means, se da una situación completamente análoga a lo descrito para el índice de Rand ajustado.

Clusters (k)	SSE	Silhouette promedio	Rand ajustado	van Dongen
5	299729.0	0.0746	0.4357	0.1831
7	273286.0	0.0871	0.4244	0.1674

Tabla 6: PAM: métricas de validez interna y externa (vs. etiquetas de clase conocidas).

La matriz de confusión del modelo de cinco clusters que presenta la tabla 7 evidencia que el tipo de arroz Karacadag fue clasificado casi en su totalidad en el cluster 2; el tipo Basmati también fue predominantemente agrupado en el cluster 0, pero en él también se incluyó una cantidad notable de imágenes del tipo Jasmine. Por el contrario, los clusters 1, 3 y 4 no tienen una tan marcada superioridad de algún tipo de arroz: podemos notar, no obstante, que componen mayormente de los tipos Jasmine, Ipsala y Arborio, respectivamente. Al igual que lo ocurrido al aplicar K-means, estas agrupaciones responden a patrones morfológicos de los granos que hacen difícil su separabilidad.

Cluster Variedad	Arborio	Ipsala	Jazmín	Karacadag	Basmati
0	5	17	387	0	919
1	96	128	441	7	43
2	91	14	0	925	0
3	403	530	36	17	3
4	405	311	136	51	35

Tabla 7: PAM: matriz de confusión para 5 clusters vs. etiquetas de variedad.

De la misma manera a lo realizado en los algoritmos anteriores, se realizó un análisis de componentes principales para reducir la dimensionalidad del espacio de atributos ($k=5$) y poder graficar los clusters hallados por el algoritmo PAM. La figura 9 muestra los cinco clusters graficados sobre los componentes 1 y 2 obtenidos mediante PCA (+40 % de variabilidad explicada): se aprecia una situación intermedia entre los casos graficados para los dos algoritmos anteriores, con clusters relativamente identificables, pero también muy superpuestos, resultando en baja separabilidad.

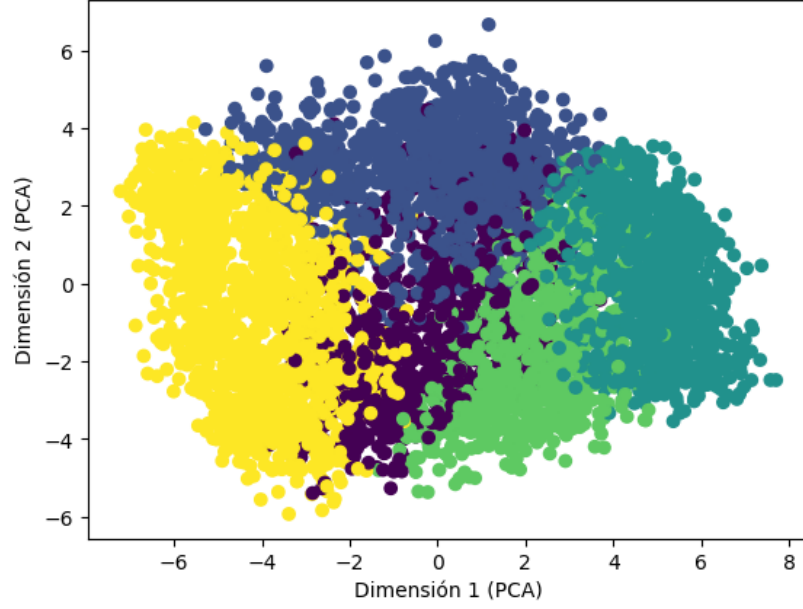


Figura 9: PAM: visualización de los 5 clusters generados. Componentes 1 y 2 obtenidos por PCA sobre atributos originales utilizados para el entrenamiento.

Cluster Variedad	Arborio	Ipsala	Jazmín	Karacadag	Basmati
0	0	0	36	0	584
1	12	33	379	0	45
2	68	13	0	891	0
3	264	213	26	7	1
4	378	318	16	50	3
5	3	26	533	0	367
6	275	397	10	52	0

Tabla 8: PAM: matriz de confusión para 7 clusters vs. etiquetas de variedad.

El caso con siete clusters, que resume la tabla (8), se comprueba nuevamente la eficiente distinción de las variedades Karacadag (cluster 2, compuesto casi exclusivamente de esta variedad) y Basmati (cluster 0, también casi exclusivamente); el resto de los clusters presenta características mixtas.

Segmentación por etiquetado de componentes conectados La imagen original alimentada al algoritmo contenía cuatro granos de arroz, los cuales fueron correctamente identificados como objetos separados entre sí y en relación al fondo como puede apreciarse en las figuras 10 (a - d). El algoritmo identificó los recuadros delimitadores de cada grano (*bounding box*) y su centroide en forma correcta. Al aplicarse a la imagen adicional de múltiples granos de arroz superpuestos obtenida de la web (figura 10e, el algoritmo resultó en la detección de más de 90 componentes. Esto sólo uno representa la mayor parte de los granos presentes en la foto, sin distinción entre ellos como muestra la figura 10f.

Segmentación por análisis espectral Para métodos de segmentación por análisis espectral se ensayaron diferentes estrategias de descomposición en autovalores. So logró el mejor resultado con el método de gradiente conjugado preconditionado de bloque localmente óptimo local o *lobpcg* [5]. Con este se obtuvo una correcta identificación de los cuatro granos de arroz en la imagen y su diferenciación con el fondo, tal como puede apreciarse en la figura 11a. En la imagen obtenida de la web (figura 11b), el mejor resultado se obtuvo con “arpack”, una estrategia de descomposición basada en el algoritmo de Arnoldi. En este caso, se indicaron 200 clusters como objetivo de segmentación (figura 11c), alcanzándose una separación entre granos significativamente más alta que con CCL.

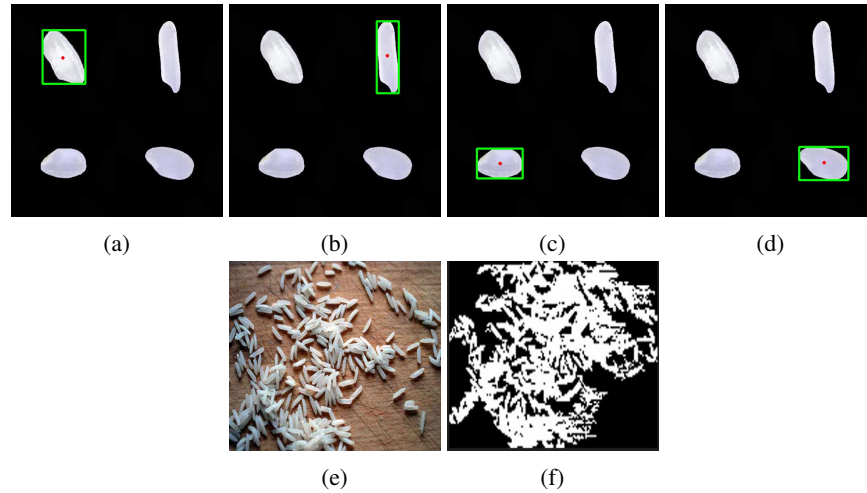


Figura 10: Detección de objetos en una imagen mediante etiquetado de componentes conectados. En rojo, centroide de cada objeto y en verde, recuadro delimitador (a-d). e) Imagen de muchos granos de arroz obtenida de la web y f) recorte de 1 de los 94 componentes etiquetados para la imagen.

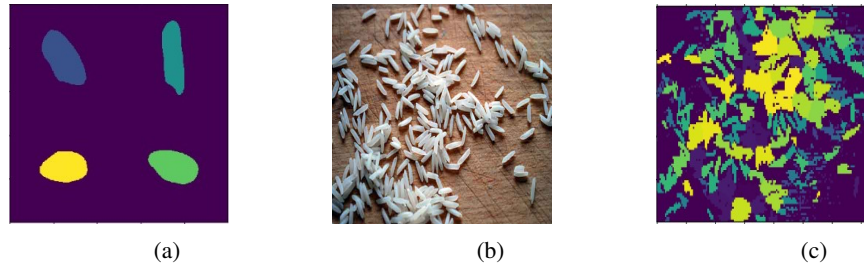


Figura 11: Segmentaciones de imágenes mediante análisis espectral. a) Imagen sintética a partir del conjunto de datos de evaluación. b) Imagen obtenida de la web con muchos granos de arroz. c) Segmentación de la imagen de la web.

4. Discusión y conclusiones

En estudios anteriores sobre este mismo conjunto de datos se evaluaron algoritmos para la clasificación supervisada de imágenes de granos de arroz, ya sea directamente sobre las imágenes o previo procesamiento para extracción de atributos de tamaño, morfología y color. En contraste, en este trabajo se buscó evaluar el desempeño de algoritmos no supervisados de clustering en la tarea de separar las imágenes en grupos mapeables con las variedades genéticas originales, partiendo de las imágenes. Para el caso de K-means y DBSCAN, las medidas de validación interna y externa no dieron un indicio claro de que el número de clusters óptimos correspondiera con el número de variedades de arroz presentes en el conjunto de datos original. DBSCAN tuvo un desempeño muy pobre, posiblemente porque los puntos estaban distribuidos de forma homogénea en el espacio de atributos y DBSCAN se basa en contrastes de espacios de alta y baja densidad para realizar la agrupación y separación de los puntos. K-means y PAM tuvieron un mejor desempeño en términos de comparabilidad de la agrupación resultante con la clasificación original en variedades de arroz, siendo PAM ligeramente superior. No obstante, para los tres algoritmos la variedad Karacadag resultó ser la más distintiva, dada su forma redondeada. Tanto K-means como PAM agrupó casi la totalidad de estos granos en un cluster cuasi-exclusivo, separado de las otras variedades, mientras que ambos algoritmos tuvieron dificultad en separar entre sí las variedades alargadas, tanto las finas (Jazmín y Basmati) como las más robustas (Arborio e Ipsala).

En el caso de los algoritmos de segmentación, tanto CCL como clustering espectral presentaron resultados satisfactorios cuando los granos de arroz individuales se presentaban bien separados en la imagen. En cambio, para una imagen con gran cantidad de granos superpuestos, CCL no fue capaz de

separar los granos ni siquiera en grupos más pequeños, mientras que el clustering espectral segmentó la imagen en conjuntos de granos más reconocibles. No se exploraron mayor cantidad de clusters objetivo para clustering espectral por limitaciones de tiempo, por lo que no se puede descartar que haya una versión del algoritmo que tenga un desempeño aún mejor que el evaluado aquí.

Por último, cabe destacar que este trabajo se realizó sobre una muestra pequeña de imágenes (5000) que resulta muy parecida al número de atributos alimentados a los algoritmos de clasificación, debido a que se trató de un trabajo preliminar. Esta línea de trabajo podría expandirse en el futuro a conjuntos de datos más nutridos que podrían resultar en agrupamientos no supervisados de mejor desempeño.

Referencias

- [1] Murat Kuklu. *Rice Image Dataset*. 2022. URL: <https://www.kaggle.com/datasets/muratkokludataset/rice-image-dataset>.
- [2] Murat Koklu, Ilkay Cinar y Yavuz Selim Taspinar. «Classification of rice varieties with deep learning methods». En: *Computers and Electronics in Agriculture* 187 (ago. de 2021), pág. 106285. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0168169921003021>.
- [3] Karen Simonyan y Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 10 de abr. de 2015. arXiv: 1409.1556[cs]. URL: <http://arxiv.org/abs/1409.1556>.
- [4] Keras Team. *Keras documentation: VGG16 and VGG19*. URL: <https://keras.io/api/applications/vgg/>.
- [5] Andrew V. Knyazev. «Toward the Optimal Preconditioned Eigensolver: Locally Optimal Block Preconditioned Conjugate Gradient Method». En: *SIAM Journal on Scientific Computing* 23.2 (ene. de 2001), págs. 517-541. URL: <http://epubs.siam.org/doi/10.1137/S1064827500366124>.